intel®

Peripheral Components

Chip Sets

Floppy Disk Controllers

Memory Controllers

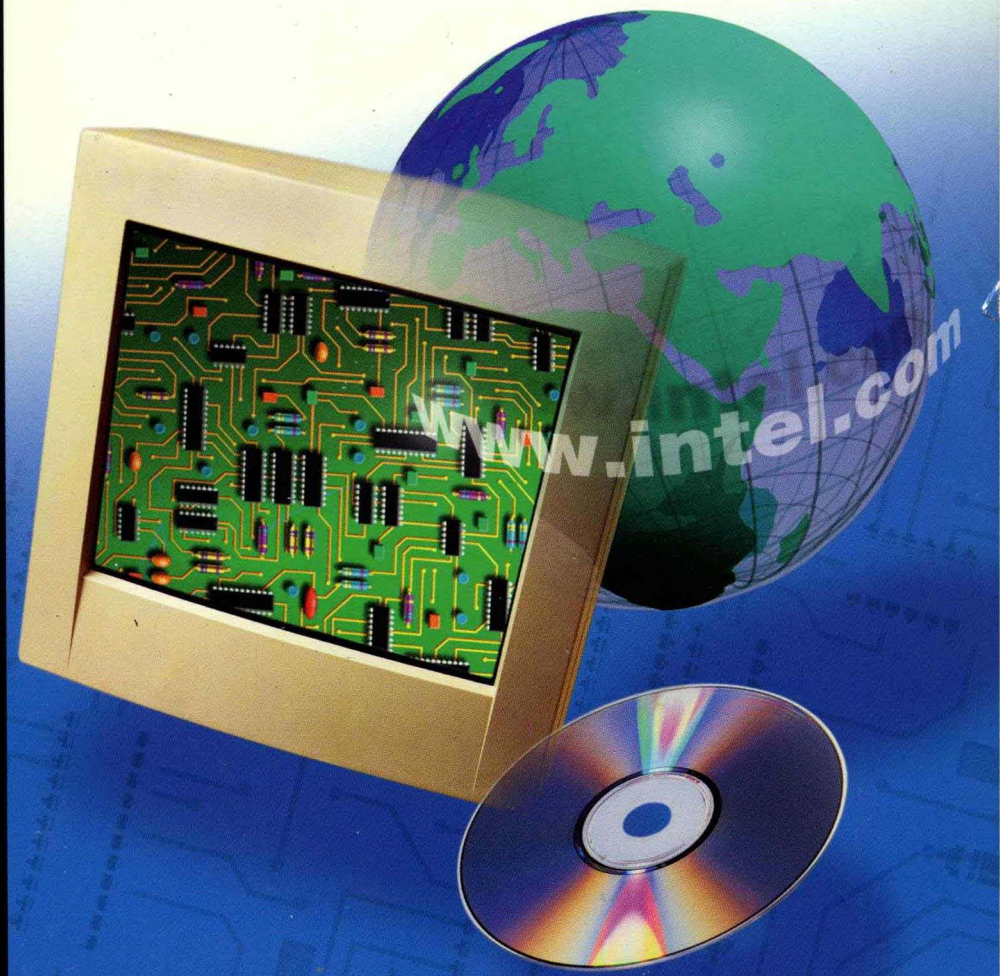
Keyboard Controllers

Support Peripherals

1996

intel®

Peripheral Components



LITERATURE

For additional information on Intel products in the U.S. or Canada, call Intel's Literature Center at (800) 548-4725 or write to:

INTEL LITERATURE SALES
P. O. Box 7641
Mt. Prospect, IL 60056-7641

To order literature outside of the U.S. and Canada contact your local sales office.

Additional information about Intel products is available on Intel's web site: http://www.intel.com.

CURRENT DATABOOKS

Product line databooks contain datasheets, application notes, article reprints, and other design information. All databooks can be ordered individually, and most are available in a pre-packaged set in the U.S. and Canada. Databooks can be ordered in the U.S. and Canada by calling TAB/McGraw-Hill at 1-800-822-8158; outside of the U.S. and Canada contact your local sales office.

Title	Intel Order Number	ISBN
SET OF NINE DATABOOKS (Available in U.S. and Canada)	**231003**	**N/A**
CONTENTS LISTED BELOW FOR INDIVIDUAL ORDERING:		
EMBEDDED MICROCONTROLLERS	270646	1-55512-248-5
EMBEDDED MICROPROCESSORS	272396	1-55512-249-3
FLASH MEMORY (2 volume set)	210830	1-55512-250-7
i960® PROCESSORS AND RELATED PRODUCTS	272084	1-55512-252-3
NETWORKING	297360	1-55512-256-6
OEM BOARDS, SYSTEMS AND SOFTWARE	280407	1-55512-253-1
PACKAGING	240800	1-55512-254-X
PENTIUM® AND PENTIUM PRO PROCESSORS AND RELATED PRODUCTS	241732	1-55512-251-5
PERIPHERAL COMPONENTS	296467	1-55512-255-8
ADDITIONAL LITERATURE: (Not included in handbook set)		
AUTOMOTIVE PRODUCTS	231792	1-55512-257-4
COMPONENTS QUALITY/RELIABILITY	210997	1-55512-258-2
EMBEDDED APPLICATIONS (1995/96)	270648	1-55512-179-9
MILITARY	210461	N/A
SYSTEMS QUALITY/RELIABILITY	231762	1-55512-046-6

A complete set of this information is available on CD-ROM through Intel's Data on Demand program, order number 240897. For information about Intel's Data on Demand ask for item number 240952.

December 1995
Order Number: 000900-001

Intel Application Support Services

World Wide Web [URL: http://www.intel.com/]

Intel's Web site now contains technical and product information that is available 24-hours a day! Also visit Intel's site for financials, history, current news and events, job opportunities, educational news and much, much more!

FaxBack

Technical and product information are available 24-hours a day! Order documents containing:

- Product Announcements
- Product Literature
- Intel Device Characteristics
- Design/Application Recommendations
- Stepping/Change Notifications
- Quality and Reliability Information

Information on the following subjects are available:

- Microcontroller and FLASH
- OEM Branded Systems
- Multibus and iRMX Software/BBS listing
- Multimedia
- Development Tools
- Quality and Reliability/Change Notification
- Microprocessor/PCI/Peripheral
- Intel Architecture Labs

To use FaxBack (for Intel components and systems), dial **(800) 628-2283** or 916-356-3105 (U.S./Canada/APAC/Japan) *or +44{0} 1793-496646 (Europe)* and follow the automated voice-prompt. Document orders will be faxed to the fax number you specify. For information on how the Intel Application Support team can help you, order our Customer Service Agreement, document #1201. Catalogs are updated as needed, so call for the latest information!

Bulletin Board System (BBS)

To use the Intel Application BBS (components and systems), dial **(503) 264-7999** or **(916) 356-3600** (U.S./Canada/APAC/Japan) *or +44{0} 1793-432955 (Europe).* The BBS will support 1200-19200 baud rate modem. *Typical modem configuration: 14.4K baud rate, No Parity, 8 Data Bits, 1 Stop Bit.*

CompuServe *Just type* 'Go Intel'

Intel maintains several forums where people come together to meet their peers, gather information, share discoveries and debate issues. For more information about service fees and access, call CompuServe at 1-800-848-8199 or 614-529-1340 (outside the U.S.). The INTELC forum is setup to support designers using various Intel components.

General Information Help Desk

Dial 1-800-628-8686 or 916-356-7599 (U.S. and Canada) between 5 a.m. to 5 p.m. PST for help with Intel products. For customers not in the U.S. or Canada, please contact your local distributor.

Intel Literature Centers

U.S.	+1-800-548-4725	France	+44{0} 1793 421777
U.S. (from overseas)	+1-708-296-9333	Germany	+44{0} 1793 421333
England	+44{0} 1793 431 155	Japan (fax only)	+81{0} 120 47 88 32

Intel Distributors

Check the back of an Intel data book or request one of the following distributor listing FaxBack documents: #4083 (U.S. Eastern Time Zone), #4084 (U.S. Central Time Zone), #4085 (Mountain Time Zone), #4086 (U.S. Alaska/Pacific Time Zone), #4209 (Europe) or #4403 (Canada).

November 1995
Order Number: 000901-001

intel®

Peripheral Components

Chip Sets, Floppy Disk Controllers, Memory Controllers, Keyboard Controllers, Support Peripherals

1996

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

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

Intel Corporation
P.O. Box 7641
Mt. Prospect, IL 60056-7641

or call 1-800-879-4683

DATASHEET DESIGNATIONS

Intel uses various datasheet markings to designate each phase of the document as it relates to the product. The markings appear in the lower inside corner of each datasheet page. Following are the definitions of each marking:

Datasheet Marking	Description
Product Preview	Contains information on products in the design phase of development. Do not finalize a design with this information. Revised information will be published when the product becomes available.
Advanced Information	Contains information on products being sampled or in the initial production phase of development.*
Preliminary	Contains preliminary information on new products in production.*
No Marking	Contains information on products in full production.*

* Specifications within these datasheets are subject to change without notice. Verify with your local Intel sales office that you have the latest datasheet before finalizing a design.

November 1995
Order Number: 000902-001

Table of Contents

Table of Contents (Continued)

Alphanumeric Index

1

Chip Sets

82437MX SYSTEM CONTROLLER (MTSC) AND 82438MX DATA PATH UNIT (MTDP)

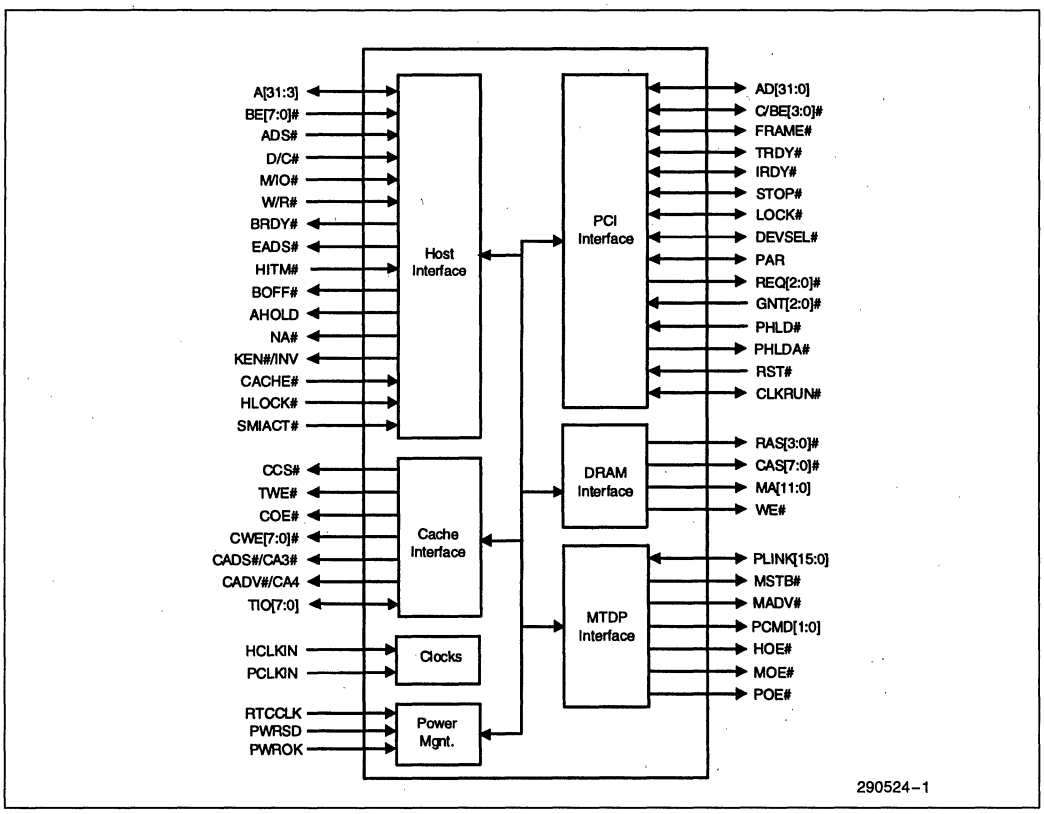
- **Supports the Pentium Processor at iCOMP™ Index 1000/120 MHz, iCOMP Index 735/90 MHz and the 610/75 MHz Pentium Processor**
- **Integrated Second Level Cache Controller**
  - **— Direct Mapped Organization**
  - **— Write-Back Cache Policy**
  - **— Cacheless, 256 Kbyte, and 512 Kbyte**
  - **— Standard, Burst and Pipelined Burst SRAMs**
  - **— Cache Hit Read/Write Cycle Timings at 3-1-1-1 with Burst or Pipelined Burst SRAMs**
  - **— Back-to-Back Read Cycles at 3-1-1-1-1-1-1-1 with Burst or Pipelined Burst SRAMs**
  - **— Integrated Tag/Valid Status Bits for Cost Savings and Performance**
  - **— Supports 3.3V SRAMs for Tag Address**
- **Integrated DRAM Controller**
  - **— 64-Bit Data Path to Memory**
  - **— 4 Mbytes to 128 Mbytes Main Memory**
  - **— EDO/Hyper Page Mode DRAM (x-2-2-2 Reads) Provides Superior Cacheless Designs**
  - **— Standard Page Mode DRAMs**
  - **— 4 RAS Lines**
  - **— 4 Qword Deep Buffer for 3-1-1-1 Posted Write Cycles**
  - **— Symmetrical and Asymmetrical DRAMs**
  - **— 3V or 5V DRAMs**
- **Power Management**
  - **— DRAM Refresh During Suspend**
  - **— Self Refresh and Extended Refresh**
- **Fully Synchronous 25/30/33 MHz PCI Bus Interface**
  - **— 100 MB/s Instant Access Enables native signal processing on Pentium Processors**
  - **— Synchronized CPU-to-PCI Interface for High Performance Graphics**
  - **— PCI Bus Arbiter: MPIIX and Three PCI Bus Masters Supported**
  - **— CPU-to-PCI Memory Write Posting with 4 Dword deep buffers**
  - **— Converts Back-to-Back Sequential CPU to PCI Memory Writes to PCI Burst Writes**
  - **— PCI-to-DRAM Posting of 12 Dwords**
  - **— PCI-to-DRAM up to 120 Mbytes/Sec Bandwidth Utilizing Snoop Ahead Feature**
- **NAND Tree for Board-Level ATE Testing**
- **208 Pin QFP for the 82437MX System Controller (MTSC); 100 Pin TQFP for Each 82438MX Data Path (MTDP)**

1

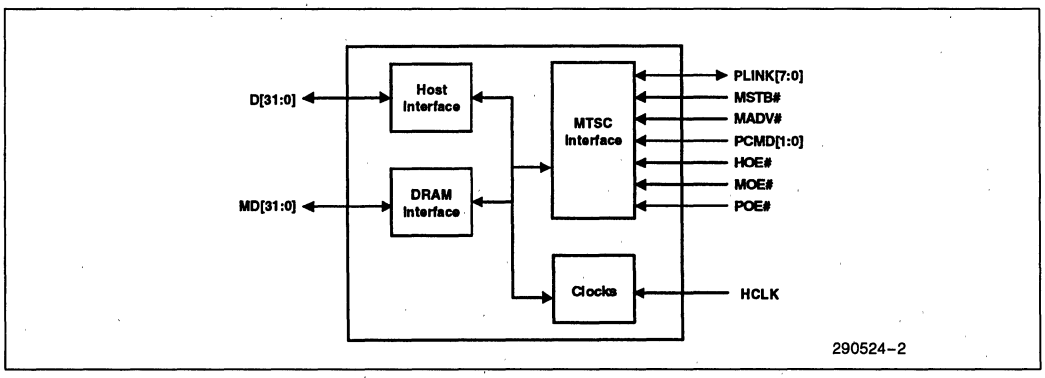
The 82430MX PCIset consists of the 82437MX System Controller (MTSC), two 82438MX Data Paths (MTDP), and the 82371MX PCI IO IDE Xcelerator (MPIIX). The MTSC/MTDP form a Host-to-PCI bridge and provide the second level cache control and a full function 64-bit data path to main memory. The MTSC integrates the cache and main memory DRAM control functions and provides bus control for transfers between the CPU, cache, main memory, and the PCI Bus. The second level (L2) cache controller supports a write-back cache policy for cache sizes of 256 KBytes and 512 KBytes. Cacheless designs are also supported. The cache memory can be implemented with either standard, burst, or pipelined burst SRAMs. An external Tag RAM is used for the address tag and an internal Tag RAM for the cache line status bits. For the MTSC's DRAM controller, four rows are supported for up to 128 MBytes of main memory. The MTSC's optimized PCI interface allows the CPU to sustain the highest possible bandwidth to the graphics frame buffer at all frequencies. Using the snoop ahead feature, the MTSC allows PCI masters to achieve full PCI bandwidth. The MTDPs provide the data paths between the CPU/cache, main memory, and PCI. For increased system performance, the MTDPs contain read prefetch and posted write buffers.

The complete document for this product is available from Intel's Literature Center at 1-800-548-4725.

December 1995
Order Number: 290524-001



82437FX TSC Simplified Block Diagram



82438FX TDP Simplified Block Diagram

82371MX PCI I/O IDE XCELERATOR (MPIIX)

- **Provides a Bridge between the PCI Bus and Extended I/O Bus**
  - **— PCI Bus; 25 MHz–33 MHz**
  - **— Extended I/O Bus; 7.5 MHz–8.33 MHz**
- **System Power Management (Intel SMM Support)**
  - **— Programmable System Management Interrupt (SMI)— Hardware/Software Events, EXTSMI#**
  - **— Programmable CPU Clock Control (STPCLK#) With Auto Clock Throttle**
  - **— Peripheral Device Power Management (Local Standby)**
  - **— Suspend State Support (Suspend-to-DRAM and Suspend-to-Disk)**
- **Enhanced DMA Functions**
  - **— Two 8237 DMA Controllers**
  - **— Compatible DMA Transfers**
  - **— PC/PCI DMA Expansion for Docking Support**
- **Fast IDE Interface**
  - **— PIO Mode 4 Transfers**
  - **— 2x16 Bit Posted Write Buffer and 1x32 bit Read Prefetch Buffer**
- **Plug-n-Play Port for Motherboard Devices**
  - **— 3 Steerable DMA Channels**
  - **— 1 Steerable Interrupt Line (Plus 2 Steerable PCI Interrupts)**
  - **— 1 Programmable Chip Select**
- **Functionality of One 82C54 Timer**
  - **— System Timer**
  - **— Refresh Request**
  - **— Speaker Tone Output**
- **Functionality of Two 82C59 Interrupt Controllers**
  - **— 14 Interrupts Supported**
  - **— Independently Programmable for Edge/Level Sensitivity**
- **X-Bus Peripheral Support**
  - **— Chip Select Decode**
  - **— Controls Lower X-Bus Data Byte Transceiver**
- **Non-Maskable Interrupts (NMI)**
  - **— PCI System Error Reporting**
- **NAND Tree for Board-Level ATE Testing**
- **176-Pin TQFP**

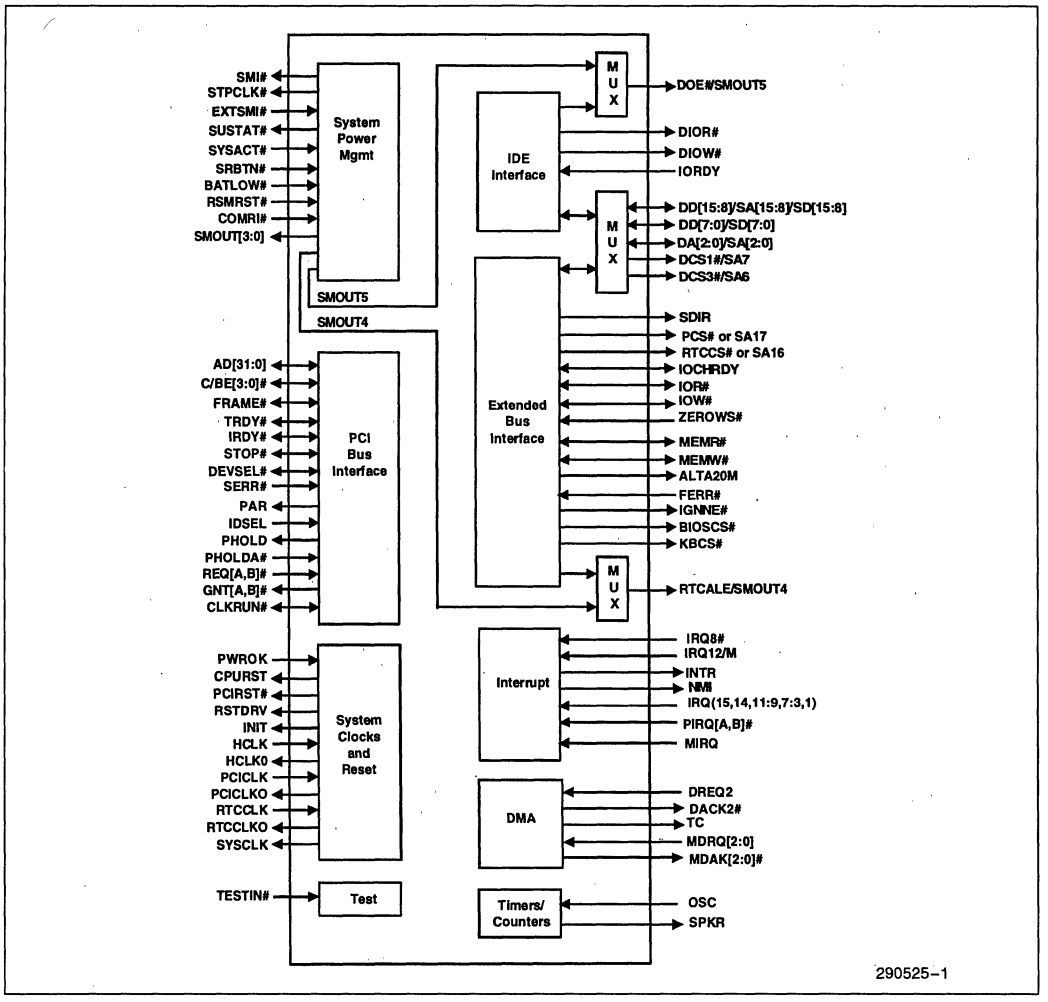
The 82371MX PCI I/O IDE Xcelerator (MPIIX) provides the bridge between the PCI bus and the ISA-like Extended I/O expansion bus. In addition, the 82371MX has an IDE interface that supports two IDE devices providing an interface for IDE hard disks and CD ROMs. The MPIIX integrates many common I/O functions found in ISA based PC systems—a seven channel DMA controller, two 82C59 interrupt controllers, an 8254 timer/counter, Intel SMM power management support, and control logic for NMI generation. Chip select decoding is provided for BIOS, real time clock, and keyboard controller. Edge/Level interrupts and interrupt steering are supported for PCI plug and play compatibility.

The MPIIX also provides the Extended I/O Bus for a direct connection to Super I/O devices providing a complete PC compatible I/O solution. MPIIX also provides support for the "Mobile PC/PCI" DMA Expansion protocol that enables the implementation of Docking Stations with full ISA and PCI capability without running the full ISA bus across the docking connector. For motherboard plug and play compatibility, the 82371MX also provides three steerable DMA channels, up to three steerable interrupt lines, and a programmable chip select. The interrupt lines can be routed to any of the available ISA interrupts.

The MPIIX's power management function supports SMI# interrupt sources, extensive clock control (including Auto Clock Throttling), peripheral power idle detection with access traps, system Suspend-to-DRAM and Suspend-to-Disk.

The complete document for this product is available from Intel's Literature Center at 1-800-548-4725.

December 1995
Order Number: 290525-001



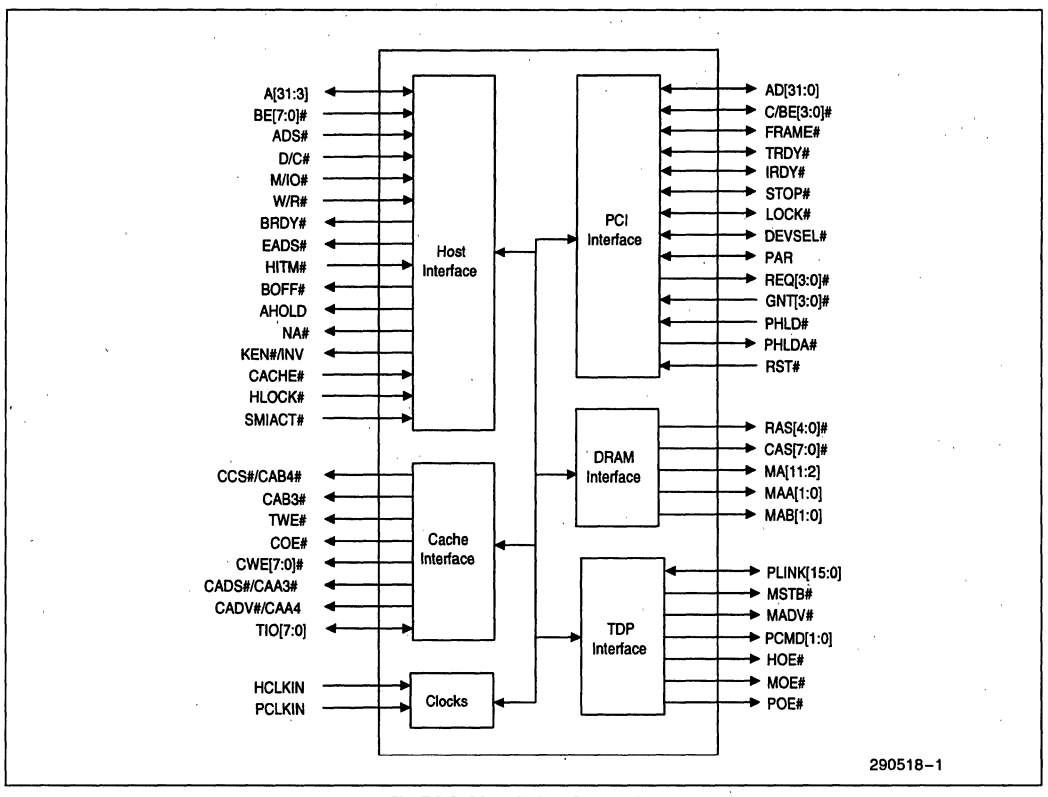
82371MX MPIIX Simplified Block Diagram

82430FX PCIset DATASHEET 82437FX SYSTEM CONTROLLER (TSC) AND 82438FX DATA PATH UNIT (TDP)

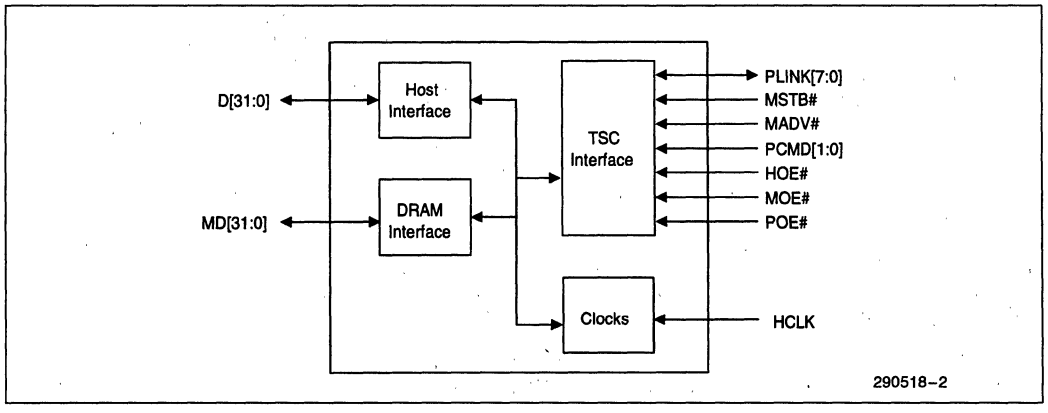
- Supports the Pentium® Processor at iCOMP® Index 1110\133 MHz, iCOMP Index 1000\120 MHz, iCOMP Index 815\100 MHz, iCOMP Index 735\90 MHz and the iCOMP Index 610\75 MHz
- Integrated Second Level Cache Controller
  - Direct Mapped Organization
  - Write-Back Cache Policy
  - Cacheless, 256-Kbyte, and 512-Kbyte
  - Standard Burst and Pipelined Burst SRAMs
  - Cache Hit Read/Write Cycle Timings at 3-1-1-1 with Burst or Pipelined Burst SRAMs
  - Back-to-Back Read Cycles at 3-1-1-1-1-1-1-1 with Burst or Pipelined Burst SRAMs
  - Integrated Tag/Valid Status Bits for Cost Savings and Performance
  - Supports 5V SRAMs for Tag Address
- Integrated DRAM Controller
  - 64-Bit Data Path to Memory
  - 4 Mbytes to 128 Mbytes Main Memory
  - EDO/Hyper Page Mode DRAM (x-2-2-2 Reads) or Standard Page Mode DRAMs
  - 5 RAS Lines
  - 4 Qword Deep Buffer for 3-1-1-1 Posted Write Cycles
  - Symmetrical and Asymmetrical DRAMs
  - 3V or 5V DRAMs
- EDO DRAM Support
  - Highest Performance with Burst or Pipelined Burst SRAMs
  - Superior Cacheless Designs
- Fully Synchronous 25/30/33 MHz PCI Bus Interface
  - 100 MB/s Instant Access Enables Native Signal Processing (NSP) on Pentium Processors
  - Synchronized CPU-to-PCI Interface for High Performance Graphics
  - PCI Bus Arbiter: PIIX and Four PCI Bus Masters Supported
  - CPU-to-PCI Memory Write Posting with 4 Dword Deep Buffers
  - Converts Back-to-Back Sequential CPU to PCI Memory Writes to PCI Burst Writes
  - PCI-to-DRAM Posting of 12 Dwords
  - PCI-to-DRAM up to 120 Mbytes/Sec Bandwidth Utilizing Snoop Ahead Feature
- NAND Tree for Board-Level ATE Testing
- 208 Pin QFP for the 82437FX System Controller (TSC); 100 Pin QFP for Each 82438FX Data Path (TDP)

The 82430FX PCIset consists of the 82437FX System Controller (TSC), two 82438FX Data Paths (TDP), and the 82371FB PCI ISA IDE Xcelerator (PIIX). The PCIset forms a Host-to-PCI bridge and provides the second level cache control and a full function 64-bit data path to main memory. The TSC integrates the cache and main memory DRAM control functions and provides bus control for transfers between the CPU, cache, main memory, and the PCI Bus. The second level (L2) cache controller supports a write-back cache policy for cache sizes of 256 Kbytes and 512 Kbytes. Cacheless designs are also supported. The cache memory can be implemented with either standard, burst, or pipelined burst SRAMs. An external Tag RAM is used for the address tag and an internal Tag RAM for the cache line status bits. For the TSC's DRAM controller, five rows are supported for up to 128 Mbytes of main memory. The TSC's optimized PCI interface allows the CPU to sustain the highest possible bandwidth to the graphics frame buffer at all frequencies. Using the snoop ahead feature, the TSC allows PCI masters to achieve full PCI bandwidth. The TDPs provide the data paths between the CPU/cache, main memory, and PCI. For increased system performance, the TDPs contain read prefetch and posted write buffers.

November 1995
Order Number: 290518-001



82437FX TSC Simplified Block Diagram



82438FX TDP Simplified Block Diagram

82430FX PCIset DATASHEET 82437FX SYSTEM CONTROLLER (TSC) AND 82438FX DATA PATH UNIT (TDP)

CONTENTS

PAGE

CONTENTS

PAGE

1.0 ARCHITECTURE OVERVIEW OF TSC/TDP

The 82430FX PCIset (Figure 1) consists of the 82437FX System Controller (TSC), two 82438FX Data Path (TDP) units, and the 82371FB PCI IDE ISA Xcelerator (PIIX). The TSC and two TDPs form a Host-to-PCI bridge. The PIIX is a multi-function PCI device providing a PCI-to-ISA bridge and a fast IDE interface. The PIIX also provides power management and has a plug and play port.

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

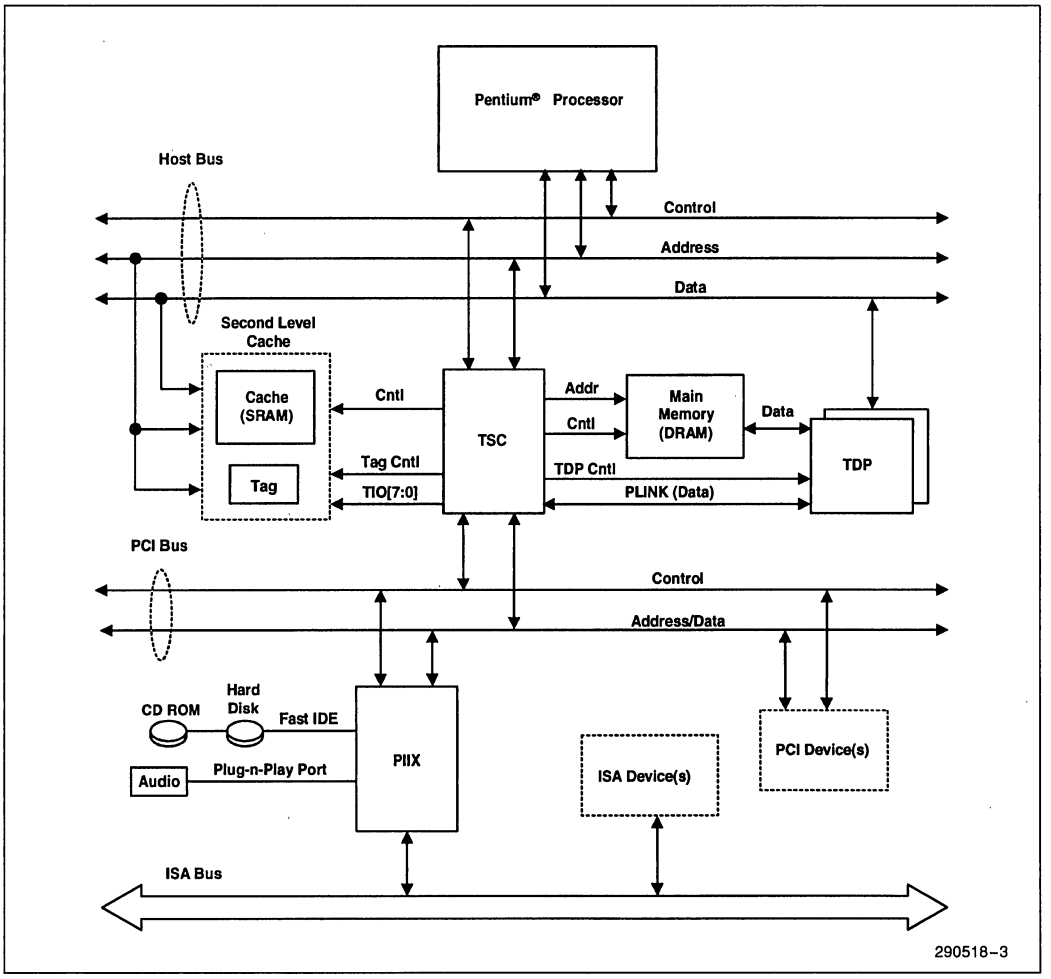


Figure 1. 82430FX PCIset System

DRAM Interface

The DRAM interface is a 64-bit data path that supports both standard page mode and Extended Data Out (EDO) (also known as Hyper Page Mode) memory. The DRAM interface supports 4 Mbytes to 128 Mbytes with five RAS lines available and also supports symmetrical and asymmetrical addressing for 512K, 1M, 2M, and 4M deep DRAMs.

Second Level Cache

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

TDP

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

PCI Interface

The PCI interface is 2.0 compliant and supports up to 4 PCI bus masters in addition to the PIIX bus master requests. While the TSC and TDP's together provide the interface between PCI and main memory, only the TSC connects to the PCI Bus.

Buffers

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

System Clocking

The processor, second level cache, main memory subsystem, and PLINK bus all run synchronous to the host clock. The PCI clock runs synchronously at half the host clock frequency. The TSC and TDP's have a host clock input and the TSC has a PCI clock input. These clocks are derived from an external source and have a maximum clock skew requirement with respect to each other.

2.0 SIGNAL DESCRIPTION

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

The "#" symbol at the end of a signal name indicates that the active, or asserted state occurs when the signal is at a low voltage level. When "#" is not present after the signal name, the signal is asserted when at the high voltage level.

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

The following notations are used to describe the signal type.

I Input is a standard input-only signal.

O Totem pole output is a standard active driver.

o/d Open drain.

t/s Tri-State is a bi-directional, tri-state input/output pin.

s/t/s Sustained tri-state is an active low tri-state signal owned and driven by one and only one agent at a time. The agent that drives a s/t/s pin low must drive it high for at least one clock before letting it float. A new agent can not start driving a s/t/s signal any sooner than one clock after the previous owner tri-states it. An external pull-up is required to sustain the inactive state until another agent drives it and must be provided by the central resource.

2.1 TSC Signals

2.1.1 HOST INTERFACE (TSC)

Signal Name	Type	Description
A[31:3]	I/O 3V	**ADDRESS BUS:** A[31:3] connect to the address bus of the CPU. During CPU cycles A[31:3] are inputs. These signals are driven by the TSC during cache snoop operations. Note that A[31:28] provide poweron/reset strapping options for the second level cache.
BE[7:0]#	I 3V	**BYTE ENABLES:** The CPU byte enables indicate which byte lane the current CPU cycle is accessing. All eight byte lanes are provided to the CPU if the cycle is a cacheable read regardless of the state of BE[7:0]#.
ADS#	I 3V	**ADDRESS STATUS:** The CPU asserts ADS# to indicate that a new bus cycle is being driven.
BRDY#	O 3V	**BUS READY:** The TSC asserts BRDY# to indicate to the CPU that data is available on reads or has been received on writes.
NA#	O 3V	**NEXT ADDRESS:** When burst SRAMs are used in the second level cache or the second level cache is disabled, the TSC asserts NA# in T2 during CPU write cycles and with the first assertion of BRDY# during CPU linefills. NA# is never asserted if the second level cache is enabled with asynchronous SRAMs. NA# on the TSC must be connected to the CPU NA# pin for all configurations.
AHOLD	O 3V	**ADDRESS HOLD:** The TSC asserts AHOLD when a PCI master is accessing main memory. AHOLD is held for the duration of the PCI burst transfer. The TSC negates AHOLD when the PCI to main memory read/write cycles complete and during PCI peer transfers.
EADS#	O 3V	**EXTERNAL ADDRESS STROBE:** Asserted by the TSC to inquire the first level cache when servicing PCI master accesses to main memory.
BOFF#	O 3V	**BACK OFF:** Asserted by the TSC when required to terminate a CPU cycle that was in progress.
HITM#	I 3V	**HIT MODIFIED:** Asserted by the CPU to indicate that the address presented with the last assertion of EADS# is modified in the first level cache and needs to be written back.
M/IO#, D/C#, W/R#	I 3V	**MEMORY/IO; DATA/CONTROL; WRITE/READ:** Asserted by the CPU with ADS# to indicate the type of cycle on the host bus.
HLOCK#	I 3V	**HOST LOCK:** All CPU cycles sampled with the assertion of HLOCK# and ADS#, until the negation of HLOCK# must be atomic (i.e., no PCI activity to main memory is allowed).
CACHE#	I 3V	**CACHEABLE:** Asserted by the CPU during a read cycle to indicate the CPU can perform a burst line fill. Asserted by the CPU during a write cycle to indicate that the CPU will perform a burst write-back cycle. If CACHE# is asserted to indicate cacheability, the TSC asserts KEN# either with the first BRDY#, or with NA#, if NA# is asserted before the first BRDY#.

Signal Name	Type	Description
KEN#/INV	O 3V	**CACHE ENABLE/INVALIDATE:** KEN#/INV functions as both the KEN# signal during CPU read cycles and the INV signal during first level cache snoop cycles. During CPU cycles, KEN#/INV is normally low. The TSC drives KEN# high during the first BRDY# or NA# assertion of a non-cacheable (in first level cache) CPU read cycle. The TSC drives INV high during the EADS# assertion of a PCI master DRAM write snoop cycle and low during the EADS# assertion of a PCI master DRAM read snoop cycle.
SMIACT#	I 3V	**SYSTEM MANAGEMENT INTERRUPT ACTIVE:** The CPU asserts SMIACT# when it is in system management mode as a result of an SMI. After SMM space (located at A0000h) is loaded and locked by BIOS, this signal must be sampled active with ADS# for the processor to access the SMM space of DRAM.

2.1.2 DRAM INTERFACE (TSC)

Signal Name	Type	Description
RAS[4:0]#	O 3V	**ROW ADDRESS STROBE:** These pins select the DRAM row.
CAS[7:0]#	O 3V	**COLUMN ADDRESS STROBE:** These pins always select which bytes are affected by a DRAM cycle.
MA[11:2]	O 3V	**MEMORY ADDRESS:** This is the row and column address for DRAM.
MAA[1:0]	O 3V	**MEMORY ADDRESS COPY A**: One copy of the MAs that change during a burst read or write of DRAM.
MAB[1:0]	O 3V	**MEMORY ADDRESS COPY B:** A second copy of the MAs that change during a burst read or write of DRAM.

2.1.3 SECONDARY CACHE INTERFACE (TSC)

Signal Name	Type	Description
CADV#/ CAA4	O 3V	**CACHE ADVANCE/CACHE ADDRESS 4 (COPY A):** This pin has two modes of operation depending on the type of SRAMs selected via hardware strapping options or programming the CC Register. The CAA4 mode is used when the L2 cache consists of asynchronous SRAMs. CAA4 is used to sequence through the Qwords in a cache line during a burst operation. CADV# mode is used when the L2 cache consists of burst SRAMs. In this mode, assertion causes the burst SRAM in the L2 cache to advance to the next Qword in the cache line.
CADS#/CAA3	O 3V	**CACHE ADDRESS STROBE/CACHE ADDRESS 3 (COPY A):** This pin has two modes of operation depending on the type of SRAMs selected via hardware strapping options or programming the CC Register. The CAA3 mode is used when the L2 cache consists of asynchronous SRAMs. CAA3 is used to sequence through the Qwords in a cache line during a burst operation. CADS# mode is used when the L2 cache consists of burst SRAMs. In this mode assertion causes the burst SRAM in the L2 cache to load the BSRAM address register from the BSRAM address pins.
CAB3	O 3V	**CACHE ADDRESS 3 (COPY B):** CAB3 is used when the L2 cache consists of asynchronous SRAMs. CAB3 is used to sequence through the Qwords in a cache line during a burst operation
CCS#/CAB4	O 3V	**CACHE CHIP SELECT/CACHE ADDRESS (COPY B):** This pin has two modes of operation depending on the type of SRAMs selected via hardware strapping options or programming the CC Register. The CAB4 mode is used when the L2 cache consists of asynchronous SRAMs. CAB4 is used to sequence through the Qwords in a cache line during a burst operation. A L2 cache consisting of burst SRAMs will power up, if necessary, and perform an access if CCS# is asserted when CADS# is asserted. A L2 cache consisting of burst SRAMs will power down if CCS# is negated when CADS# is asserted. When CCS# is negated, a L2 cache consisting of burst SRAMs ignores ADS#. If CCS# is asserted when ADS# is asserted, a L2 cache consisting of burst SRAMs will power up, if necessary, and perform an access.
COE#	O 3V	**CACHE OUTPUT ENABLE**: The secondary cache data RAMs drive the CPU's data bus when COE# is asserted.
CWE[7:0]#	O 3V	**CACHE WRITE ENABLE:** Each CWE# corresponds to one byte lane. Assertion causes the byte lane to be written into the secondary cache data RAMs if they are powered up.
TIO[7:0]	I/O 5V	**TAG ADDRESS:** These are inputs during CPU accesses and outputs during L2 cache line fills and L2 cache line invalidates due to inquire cycles. TIO[7:0] contain the L2 tag address for 256-Kbyte L2 caches. TIO[6:0] contains the L2 tag address and TIO7 contains the L2 cache valid bit for 512-Kbyte caches.
TWE#	O 5V	**TAG WRITE ENABLE:** When asserted, new state and tag addresses are written into the external tag.

2.1.4 PCI INTERFACE (TSC)

Signal Name	Type	Description
AD[31:0]	I/O 5V	**ADDRESS DATA BUS:** The standard PCI address and data lines. The address is driven with FRAME# assertion and data is driven or received in following clocks.
C/BE[3:0]#	I/O 5V	**COMMAND, BYTE ENABLE:** The command is driven with FRAME# assertion. Byte enables corresponding to supplied or requested data are driven on following clocks.
FRAME#	I/O 5V	**FRAME:** Assertion indicates the address phase of a PCI transfer. Negation indicates that one more data transfer is desired by the cycle initiator.
DEVSEL#	I/O 5V	**DEVICE SELECT:** The TSC drvies DEVSEL# when a PCI initiator attempts to access main memory. DEVSEL# is asserted at medium decode time.
IRDY#	I/O 5V	**INITIATOR READY:** Asserted when the initiator is ready for a data transfer.
TRDY#	I/O 5V	**TARGET READY:** Asserted when the target is ready for a data transfer.
STOP#	I/O 5V	**STOP:** Asserted by the target to request the master to stop the current transaction.
LOCK#	I/O 5V	**LOCK:** Used to establish, maintain, and release resource locks on PCI.
REQ[3:0]#	I 5V	**REQUEST:** PCI master requests for PCI.
GNT[3:0]#	O 5V	**GRANT:** Permission is given to the master to use PCI.
PHLD#	I 5V	**PCI HOLD**: This signal comes from the PIIX. PHLD# is the PIIX request for the PCI Bus. The TSC flushes the DRAM Write Buffers and acquires the host bus before granting PIIX via PHLDA#. This ensures that the guaranteed access time is met for ISA masters.
PHLDA#	O 5V	**PCI HOLD ACKNOWLEDGE:** This signal is driven by the TSC to grant PCI to the PIIX.
PAR	I/O 5V	**PARITY:** A single parity bit is provided over AD[31:0] and C/BE[3:0].
RST#	I 5V	**RESET:** When asserted, RST# resets the TSC and sets all register bits to the default value.

2.1.5 TDP INTERFACE (TSC)

Signal Name	Type	Description
PLINK[15:0]	I/O 3V	**PCI LINK:** These signals are connected to the PLINK data bus on the TDP. This is the data path between the TSC and TDP. Each TDP connects to one byte of the 16-bit bus.
MSTB#	O 3V	**MEMORY STROBE:** Assertion causes data to be posted in the DRAM Write Buffer.
MADV#	O 3V	**MEMORY ADVANCE:** For memory write cycles, assertion causes a Qword to be drained from the DRAM Write Buffer and the next data to be made available to the MD pins of the TDPs. For memory read cycles, assertion causes a Qword to be latched in the DRAM Input Register.
PCMD[1:0]	O 3V	**PLINK COMMAND:** This field controls how data is loaded into the PLINK input and output registers.
HOE#	O 3V	**HOST OUTPUT ENABLE:** This signal is used as the output enable for the Host Data Bus.
MOE#	O 3V	**MEMORY OUTPUT ENABLE:** This signal is used as the output enable for the memory data bus. A buffered copy of MOE# also serves as a WE# select for the DRAM array.
POE#	O 3V	**PLINK OUTPUT ENABLE:** This signal is used as the output enable for the PLINK Data Bus.

2.1.6 CLOCKS (TSC)

Signal Name	Type	Description
HCLKIN	I 5V	**HOST CLOCK IN:** This pin receives a buffered host clock. This clock is used by all of the TSC logic that is in the Host clock domain. This should be the same clock net that is delivered to the CPU. The net should tee and have equal lengths from the tee to the CPU and the TSC.
PCLKIN	I 5V	**PCI CLOCK IN:** This pin receives a buffered divide-by-2 host clock. This clock is used by all of the TSC logic that is in the PCI clock domain.

2.2 TDP Signals

2.2.1 DATA INTERFACE SIGNALS (TDP)

Signal Name	Type	Description
HD[31:0]	I/O 3V	**HOST DATA**: These signals are connected to the CPU data bus. The CPU data bus is interleaved between the two TDPs for every byte, effectively creating an even and an odd TDP.
MD[31:0]	I/O 3V/5V	**MEMORY DATA:** These signals are connected to the DRAM data bus. The DRAM data bus is interleaved between the two TDPs for every byte, effectively creating an even and an odd TDP.
PLINK[7:0]	I/O 3V	**PCI LINK:** These signals are connected to the PLINK data bus on the TSC. This is the data path between the TSC and TDP. Each TDP connects to one byte of the 16-bit bus.

2.2.2 TSC INTERFACE SIGNALS (TDP)

Signal Name	Type	Description
MSTB#	I 3V	**MEMORY STROBE:** Assertion causes data to be posted in the DRAM Write Buffer.
MADV#	I 3V	**MEMORY ADVANCE:** For memory write cycles, assertion causes a Qword to be flushed from the DRAM Write Buffer and the next data to be made available to the MD pins of the TDPs. For memory read cycles, assertion causes a Qword to be latched in the DRAM Input register.
PCMD[1:0]	I 3V	**PLINK COMMAND:** This field controls how data is loaded into the PLINK input and output registers.
HOE#	I 3V	**HOST OUTPUT ENABLE:** This signal is used as the output enable for the Host Data Bus.
MOE#	I 3V	**MEMORY OUTPUT ENABLE:** This signal is used as the output enable for the Memory Data Bus.
POE#	I 3V	**PLINK OUTPUT ENABLE:** This signal is used as the output enable for the PLINK Data Bus.

2.2.3 CLOCK SIGNAL (TDP)

Signal Name	Type	Description
HCLK	I 5V	**HOST CLOCK**: Primary clock input used to drive the part.

2.3 Signal State During Reset

Table 1 shows the state of all TSC and TDP output and bi-directional signals during a hard reset (RST# asserted). The TSC samples the strapping options on the A[31:28] signal lines on the rising edge of RST#. When RST# is asserted, the TSC enables the TDP outputs via the HOE#, MOE#, and POE# TSC/TDP interface signals. When RST# is negated, the TSC resets the TDP logic by driving HOE#, MOE#, and POE# inactive for two HCLKs.

Table 1. Output and I/O Signal States During Hard Reset

Signal	State
TSC	
A[31:28]	Input
A[27:3]	Low
BRDY#	High
NA#	High
AHOLD	High
EADS#	High
BOFF#	High
KEN#/INV	Low
RAS[4:0]#	Low
CAS[7:0]#	Low
MA[11:2]	Low
MAA[1:0]	Low
MAB[1:0]	Low
CADV#/CAA4	High
CADS#/CAA3	High

Signal	State
CCS#/CAB4	High
CAB3	High
COE#	High
CWE[7:0]#	High
TIO[7:0]#	Tri-state
TWE#	High
AD[31:0]	Low
C/BE[3:0]#	Low
FRAME#	Tri-state
DEVSEL#	Tri-state
IRDY#	Tri-state
TRDY#	Tri-state
STOP#	Tri-state
LOCK#	Tri-state
GNT[3:0]#	Tri-state
PHLDA#	Tri-state

Signal	State
PAR	Low
PLINK[15:0]	Low
MSTB#	High
MADV#	High
PCMD[1:0]	High
HOE#	High
MOE#	Low
POE#	Low
TDP	
HD[31:0]	Tri-state Low
MD[31:1]	Tri-state Low
MD0	NAND Tree Output
PLINK[7:0]	Tri-state Low

3.0 REGISTER DESCRIPTION

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

The TSC internal registers (both I/O Mapped and Configuration registers) are only accessible by the Host CPU and cannot be accessed by PCI masters. The registers can be accessed as Byte, Word (16-bit), or Dword (32-bit) quantities, with the exception of CONFADD which can only be accessed as a Dword. All multi-byte numeric fields use "little-endian" ordering (i.e., lower addresses contain the least significant parts of the field). The following nomenclature is used for access attributes.

RO **Read Only**. If a register is read only, writes to this register have no effect.

R/W **Read/Write**. A register with this attribute can be read and written.

R/WC **Read/Write Clear**. A register bit with this attribute can be read and written. However, a write of 1 clears (sets to 0) the corresponding bit and a write of 0 has no effect.

Some of the TSC registers described in this section contain reserved bits. Software must deal correctly with fields that are reserved. On reads, software must use appropriate masks to extract the defined bits and not rely on reserved bits being any particular value. On writes, software must ensure that the values of reserved bit positions are preserved. That is, the values of reserved bit positions must first be read, merged with the new values for other bit positions and then written back.

In addition to reserved bits within a register, the TSC contains address locations in the PCI configuration space that are marked "Reserved" (Table 2). The TSC responds to accesses to these address locations by completing the Host cycle. Software should not write to reserved TSC configuration locations in the device-specific region (above address offset 3Fh).

During a hard reset (RST# asserted), the TSC sets its internal configuration registers to predetermined **default** states. The default state represents the minimum functionality feature set required to successfully bring up the system. Hence, it does not represent the optimal system configuration. It is the responsibility of the system initialization software (usually BIOS) to properly determine the DRAM configurations, cache configuration, operating parameters and optional system features that are applicable, and to program the TSC registers accordingly.

3.1 Control Registers

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

3.1.1 CONFADD—CONFIGURATION ADDRESS REGISTER

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

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

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

3.1.2 CONFDATA—CONFIGURATION DATA REGISTER

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

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

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

3.2 PCI Configuration Registers

The PCI Bus defines a slot based "configuration space" that allows each device to contain up to 256 8-bit configuration registers. The PCI specification defines two bus cycles to access the PCI configuration space—**Configuration Read** and **Configuration Write**. While memory and I/O spaces are supported by the Pentium microprocessor, configuration space is not supported. The PCI specification defines two mechanisms to access configuration space, Mechanism #1 and Mechanism #2. The TSC only supports Mechanism #1. Table 2 shows the TSC configuration space.

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

Type 0 Access

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

Type 1 Access

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

Table 2. TSC Configuration Space

Address Offset	Symbol	Register Name	Access
00–01h	VID	Vendor Identification	RO
02–03h	DID	Device Identification	RO
04–05h	PCICMD	Command Register	R/W
06–07h	PCISTS	Status Register	RO, R/WC
08h	RID	Revision Identification	RO
09h		Reserved	
0Ah	SUBC	Sub-Class Code	RO
0Bh	BCC	Base Class Code	RO
0Ch		Reserved	
0Dh	MLT	Master Latency Timer	R/W
0Eh		Reserved	
0Fh	BIST	BIST Register	R/W
10–49h		Reserved	
50h	PCON	PCI Control Register	R/W
51h		Reserved	
52h	CC	Cache Control	R/W
53–56h		Reserved	
57h	DRAMC	DRAM Control	R/W
58h	DRAMT	DRAM Timing	R/W
59–5Fh	PAM[6:0]	Programmable Attribute Map (7 registers)	R/W
60–64h	DRB[4:0]	DRAM Row Boundary (5 registers)	R/W
65–67h		Reserved	
68h	DRT	DRAM Row Type	R/W
69–71h		Reserved	
72h	SMRAM	System Management RAM Control	R/W
73–FFh		Reserved	

3.2.1 VID—VENDOR IDENTIFICATION REGISTER

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

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

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

3.2.2 DID—DEVICE IDENTIFICATION REGISTER

Address Offset: 02–03h
Default Value: 122h
Attribute: Read Only

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

Bit	Description
15:0	**Device Identification Number.** This is a 16-bit value assigned to the TSC.

3.2.3 PCICMD—PCI COMMAND REGISTER

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

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

Bit	Descriptions
15:10	**Reserved.**
9	**Fast Back-to-Back: (Not Implemented)** This bit is hardwired to 0.
8	**SERR# Enable (SERRE): (Not Implemented)** This bit is hardwired to 0.
7	**Address/Data Stepping: (Not Implemented)** This bit is hardwired to 0.
6	**Parity Error Enable (PERRE): (Not Implemented)** This bit is hardwired to 0.
5:3	**Reserved:** These bits are hardwired to 0.
2	**Bus Master Enable (BME): (Not Implemented)** The TSC does not support disabling of its bus master capability on the PCI Bus. This bit is hardwired to 1.
1	**Memory Access Enable (MAE):** 1 = Enable PCI master access to main memory, if the PCI address selects enabled DRAM space; 0 = Disable (TSC does not respond to main memory accesses).
0	**I/O Access Enable (IOAE): (Not Implemented)** This bit is hardwired to 0. The TSC does not respond to PCI I/O cycles.

3.2.4 PCISTS—PCI STATUS REGISTER

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

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

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

3.2.5 RID—REVISION IDENTIFICATION REGISTER

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

This register contains the revision number of the TSC.

Bit	Description
7:0	**Revision Identification Number:** This is an 8-bit value that indicates the revision identification number for the TSC.

3.2.6 SUBC—SUB-CLASS CODE REGISTER

Address Offset: 0Ah
Default Value: 00h
Access: Read Only

This register indicates the function sub-class in relation to the Base Class Code.

Bit	Description
7:0	**Sub-Class Code (SUBC):** 00h = Host bridge.

3.2.7 BCC—BASE CLASS CODE REGISTER

Address Offset: 0Bh
Default Value: 06h
Access: Read Only

This register contains the Base Class Code of the TSC.

Bit	Description
7:0	**Base Class Code (BASEC):** 06h = Bridge device.

3.2.8 MLT—MASTER LATENCY TIMER REGISTER

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

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

Bit	Description
7:3	**Master Latency Timer Count Value:** The number of clocks programmed in the MLT represents the minimum guaranteed time slice (measured in PCI clocks) allotted to the TSC, after which it must surrender the bus as soon as other PCI masters are granted the bus. The default value of MLT is 00h or 0 PCI clocks. The MLT should always be programmed to a non-zero value.
2:0	**Reserved:** Hardwired to 0.

3.2.9 BIST—BIST REGISTER

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

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

Bit	Descriptions
7	**BIST Supported—RO:** 00h = Disable BIST function.
6	**Start BIST:** This function is not supported and writes have no affect.
5:4	**Reserved.**
3:0	**Completion Code—RO:** This field always returns 0 when read and writes have no affect.

3.2.10 PCON—PCI CONTROL REGISTER

Address Offset: 50h
Default: 00h
Access: Read/Write

The PCON Register enables and disables features related to the PCI unit operation not already covered in the PCI required configuration space.

Bit	Descriptions
7:5	**CPU Inactivity Timer Bits:** This field selects the value used in the CPU Inactivity Timer. This timer counts CPU inactivity in PCI clocks. The inactivity window is defined as the last BRDY# to the next ADS#. When active, the CPU is default owner of the PCI Bus. If the CPU is inactive, PHOLD and the REQx# lines are given priority. **Bits[7:5] PCI Clocks** 000 1 001 2 010 3* 011 4 100 5* 101 6 110 7 111 8 * Recommended settings
4	**Reserved.**
3	**Peer Concurrency Enable Bit (PCE):** 1 = Peer Concurrency enabled. 0 = Peer Concurrency disabled. This bit is normally programmed to 1.
2	**CPU-to-PCI Write Bursting Disable Bit:** 0 = CPU-to-PCI Write bursting enabled. 1 = CPU-to-PCI Write bursting disabled.
1	**PCI Streaming Bit:** 0 = PCI streaming enabled. 1 = PCI streaming disabled.
0	**Bus Concurrency Disable Bit:** 0 = Bus concurrency enabled. 1 = Bus concurrency disabled.

3.2.11 CC—CACHE CONTROL REGISTER

Address Offset: 52h
Default: SSSS0010 (S = Strapping option)
Access: Read/Write

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

Bit	Description
7:6	**Secondary Cache Size (SCS):** This field reflects the inverted signal level on the A[31:30] pins at the rising edge of the RESET signal (default). The default values can be overwritten with subsequent writes to the CC Register. The options for this field are: **Bits[7:6] — Secondary Cache Size** 0 0 — Cache not populated 0 1 — 256 Kbytes 1 0 — 512 Kbytes 1 1 — Reserved **NOTE:** 1. When SCS = 00, the L2 cache is disabled and the cache tag state is frozen. 2. To enable the L2 cache, SCS must be non-zero and the FLCE bit must be 1.
5:4	**SRAM Type (SRAMT):** This field reflects the inverted signal level on the A[29:28] pins at the rising edge of the RESET signal (default). The default values can be overwritten with subsequent writes to the CC Register. The options for this field are: **Bits[5:4] — SRAM Type** 0 0 — Pipelined Burst 0 1 — Burst 1 0 — Asynchronous 1 1 — Pipelined Burst for 512K/Dual-bank inplementations. Selects 3-1-1-1-2-1-1-1 instead of 3-1-1-1-1-1-1-1 back-to-back burst timings with NA# enabled. An extra clock is inserted for bank turnaround. SCS must be set to 10.
3	**NA# Disable Bit:** 0 = NA# will be asserted as appropriate by the TSC (default). 1 = TCS's NA# signal is never asserted. This bit should be configured as desired before either the L1 or L2 caches are enabled.
2	**Reserved.**
1	**Secondary Cache Force Miss or Invalidate (SCFMI):** When SCFMI = 1, the L2 hit/miss detection is disabled, and all tag lookups result in a miss. If the L2 is enabled, the cycle is processed as a miss. If the L2 is populated but disabled (FLCE = 0) and SCFMI = 1, any CPU read cycle invalidates the selected tag entry. When SCFMI = 0, normal L2 cache hit/miss detection and cycle processing occurs. Software can flush the cache (cause all modified lines to be written back to main memory) by setting SCFMI to 1 with the L2 cache enabled (SCS≠00 and FLCE = 1), and reading all L2 cache tag address locations.

Bit	Description
0	**First Level Cache Enable (FLCE):** FLCE enables/disables the first level cache. When FLCE = 1, the TSC responds to CPU cycles with KEN# asserted for cacheable memory cycles. When FLCE = 0, KEN# is always negated and line fills to either the first level or L2 cache are prevented. Note that, when FLCE = 1 and SCFMI = 1, writes to the cache are also forced as misses. Thus, it is possible to create incoherent data between main memory and the L2 cache. A summary of FLCE/SCFMI bit interactions is as follows: **FLCE SCFMI L2 Cache Result** 0 0 Disabled 0 1 Disabled; tag invalidate on reads 1 0 Normal L2 cache operation (dependent on SCS) 1 1 Enabled; miss forced on reads/writes

3.2.12 DRAMC—DRAM CONTROL REGISTER

Address Offset: 57h
Default Value: 01h
Access: Read/Write

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

Bit	Description
7:6	**Hole Enable (HEN):** This field enables a memory hole in main memory space. CPU cycles matching an enabled hole are passed on to PCI. PCI cycles matching an enabled hole are ignored by the TSC (no DEVSEL#). Note that a selected hole is not remapped. Note that this field should not be changed while the L2 cache is enabled. **Bits[7:6] Hole Enabled** 00 None 01 512 Kbytes –640 Kbytes 10 15 Mbytes –16 Mbytes 11 Reserved
5:4	**Reserved.**
3	**EDO Detect Mode Enable (EDME):** This bit, if set to 1, enables a special timing mode for BIOS to detect EDO DRAM type on a bank-by-bank basis. Once all DRAM row banks have been tested for EDO, the EDME bit should be set to 0. Otherwise, performance will be seriously impacted. An algorithm for using the EDME bit 3 follows the table.
2:0	**DRAM Refresh Rate (DRR):** The DRAM refresh rate is adjusted according to the frequency selected by this field. Note that refresh is also disabled via this field, and that disabling refresh results in the eventual loss of DRAM data. **Bits[2:0] Host Bus Frequency** 000 Refresh Disabled 001 50 MHz 010 60 MHz 011 66 MHz 1XX Reserved

DRAM Type Detection

The EDO Detect Mode Enable field (bit 3) provides a special timing mode that allows BIOS to determine the DRAM type in each of the banks of main memory DRAM. To exploit the performance improvements from EDO DRAMs, the BIOS should provide for dynamic detection of any EDO DRAMs in the DRAM rows.

3.2.13 DRAMT—DRAM TIMING REGISTER

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

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

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>7</td><td>Reserved.</td></tr>
<tr><td>6:5</td><td>DRAM Read Burst Timing (DRBT): The DRAM read burst timings are controlled by the DRBT field. The timing used depends on the type of DRAM on a per-bank basis, as indicated by the DRT register.

<table>
<tr><th>DRBT</th><th>EDO Burst Rate</th><th>Standard Page Mode Rate</th></tr>
<tr><td>00</td><td>x444</td><td>x444</td></tr>
<tr><td>01</td><td>x333</td><td>x444</td></tr>
<tr><td>10</td><td>x222</td><td>x333</td></tr>
<tr><td>11</td><td>Reserved</td><td>Reserved</td></tr>
</table>
This field is typically set to "01" or "10" depending on the system configuration.</td></tr>
<tr><td>4:3</td><td>DRAM Write Burst Timing (DWBT): The DRAM write burst timings are controlled by the DWBT field. Slower rates may be required in certain system designs to support layouts with longer trace lengths or slower DRAMs. Most system designs will be able to use one of the faster burst mode timings.

<table>
<tr><th>DWBT</th><th>Standard Page Mode Rate</th></tr>
<tr><td>00</td><td>x444</td></tr>
<tr><td>01</td><td>x333</td></tr>
<tr><td>10</td><td>x222 (see notes 1 and 2)</td></tr>
<tr><td>11</td><td>Reserved</td></tr>
</table>
NOTES:
1. Minimum 3-Clock CAS# Cycle Time for Single Writes. The DWBT field controls the minimum CAS# cycle time for single and burst write cycles, except for the x222 programming case in which the minimum cycle time for single writes is limited to 3-clocks.
2. Two clock writes should not be programmed at 66 MHz.</td></tr>
<tr><td>2</td><td>RAS to CAS Delay (RCD): RCD controls the DRAM page miss and row miss leadoff timings. When RCD = 1, the RAS active to CAS active delay is 2 clocks. When RCD = 0, the timing is 3 clocks. Note that RCD timing adjustments are independent to DLT timing adjustments.

<table>
<tr><th>RCD</th><th>RAS to CAS Delay</th></tr>
<tr><td>0</td><td>3</td></tr>
<tr><td>1</td><td>2</td></tr>
</table></td></tr>
</table>

Bit	Description
1:0	**DRAM Leadoff Timing (DLT):** The DRAM leadoff timings for page/row miss cycles are controlled by the DLT bits. DLT controls the MA setup to the first CAS# assertion. The DLT bits do not effect page hit cycles. **DLT** **Read Leadoff** **Write Leadoff** **RAS# Precharge** 00 8 6 3 01 7 5 3 10 8 6 4 11 7 5 4 Note that the DLT field and RCD bit have cumulative affects (i.e., setting DLT0 = 0 and RCD = 0 results in two additional clocks betwwen RAS# assertion and CAS# assertion).

3.2.14 PAM—PROGRAMMABLE ATTRIBUTE MAP REGISTERS (PAM[6:0])

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

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

RE **Read Enable**. When RE = 1, the CPU read accesses to the corresponding memory segment are directed to main memory. Conversely, when RE = 0, the CPU read accesses are directed to PCI.

WE **Write Enable**. When WE = 1, the CPU write accesses to the corresponding memory segment are directed to main memory. Conversely, when WE = 0, the CPU write accesses are directed to PCI.

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

The RE and WE attributes permit a memory segment to be Read Only, Write Only, Read/Write, or disabled (Table 3.). For example, if a memory segment has RE = 1 and WE = 0, the segment is Read Only.

Table 3. Attribute Definition

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

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

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

Table 4. Attribute Bit Assignment

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

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

Table 5. PAM Registers and Associated Memory Segments

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

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

DOS Application Area (00000–9FFFh)

Read, write, and cacheability attributes are always enabled and are not programmable for the 0–640-Kbyte DOS application region.

Video Buffer Area (A0000–BFFFFh)

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

Expansion Area (C0000–DFFFFh)

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

Extended System BIOS Area (E0000–EFFFFh)

This 64-Kbyte area is divided into four 16-Kbyte segments. Each segment can be assigned independent cacheability, read, and write attributes. Memory segments that are disabled are not remapped elsewhere.

System BIOS Area (F0000–FFFFFh)

This area is a single 64-Kbyte segment. This segment can be assigned cacheability, read, and write attributes. When disabled, this segment is not remapped.

Extended Memory Area (100000–FFFFFFFFh)

The extended memory area can be split into several parts;

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

On power-up or reset the CPU vectors to the Flash BIOS area, mapped in the range of 4 Gbyte to 4 Gbyte - 512 Kbyte. This area is physically mapped on the expansion bus. Since these addresses are in the upper 4 Gbyte range, the request is directed to PCI.

The main memory space can occupy extended memory from a minimum of 1 Mbyte up to 128 Mbytes. This memory is cacheable.

PCI memory space from the top of main memory to 4Gbytes is always non-cacheable.

1

3.2.15 DRB—DRAM ROW BOUNDARY REGISTERS

Address Offset: 60h(DRB0)—64h(DRB4)

Default Value: 02h

Access: Read/Write

The TSC supports 5 rows of DRAM. Each row is 64 bits wide. The DRAM Row Boundary Registers define upper and lower addresses for each DRAM row. Contents of these 8-bit registers represent the boundary addresses in 4-Mbyte granularity. Note that bit 0 of each DRB must always be programmed to 0 for proper operation.

DRB0 = Total amount of memory in row 0 (in 4 Mbytes)
DRB1 = Total amount of memory in row 0 + row 1 (in 4 Mbytes)
DRB2 = Total amount of memory in row 0 + row 1 + row 2 (in 4 Mbytes)
DRB3 = Total amount of memory in row 0 + row 1 + row 2 + row 3 (in 4 Mbytes)
DRB4 = Total amount of memory in row 0 + row 1 + row 2 + row 3 + row4 (in 4 Mbytes)

The DRAM array can be configured with 512Kx32, 1Mx32, 2Mx32, and 4Mx32 SIMMs. Each register defines an address range that causes a particular RAS# line to be asserted (e.g., if the first DRAM row is 8 Mbytes in size then accesses within the 0 to 8-Mbyte range causes RAS0# to be asserted).

Bit	Description
7:6	**Reserved.**
5:0	**Row Boundary Address:** This 6-bit field is compared against address lines A[27:22] to determine the upper address limit of a particular row (i.e., DRB minus previous DRB = row size).

Row Boundary Address

These 6-bit values represent the upper address limits of the 5 rows (i.e., this row minus previous row = row size). Unpopulated rows have a value equal to the previous row (row size = 0). DRB4 reflects the maximum amount of DRAM in the system. The top of memory is determined by the value written into DRB4. If DRB4 is greater than 128 Mbytes, then 128 Mbytes of DRAM are available.

As an example of a general purpose configuration where 4 physical rows are configured for either single-sided or double-sided SIMMs, the memory array would be configured like the one shown Figure 2. In this configuration, the TSC drives two RAS# signals directly to the SIMM rows. If single-sided SIMMs are populated, the even RAS# signal is used and the odd RAS# is not connected. If double-sided SIMMs are used, both RAS# signals are used.

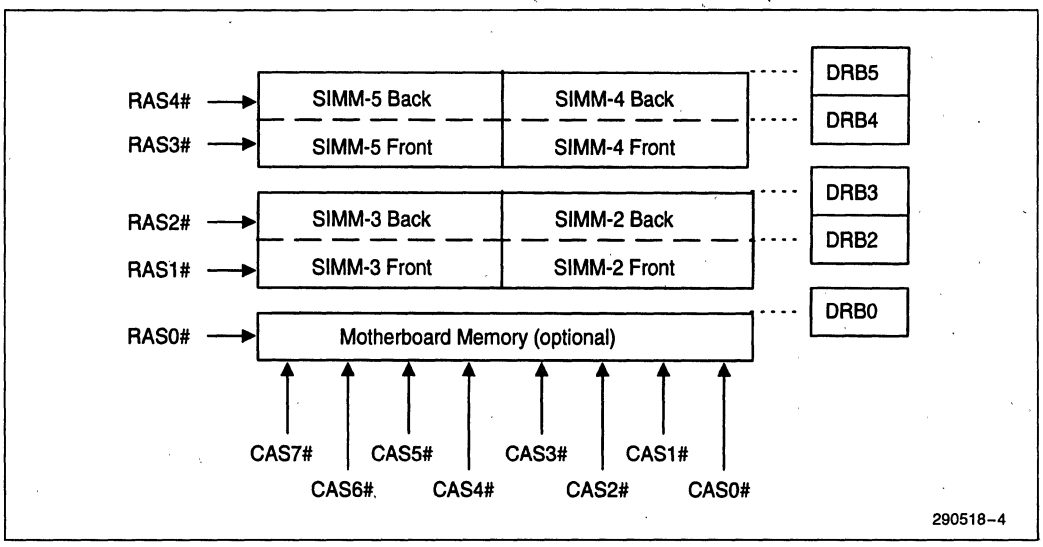


Figure 2. SIMMs and Corresponding DRB Registers

The following 2 examples describe how the DRB Registers are programmed for cases of single-sided and double-sided SIMMs on a motherboard having a total of four 8-byte or eight 4-byte SIMM sockets.

Example #1

The memory array is populated with four single-sided 1MB x 32 SIMMs, a total of 16 Mbytes of DRAM. Two SIMMs are required for each populated row making each populated row 8 Mbytes in size.

DRB0 = 02h	populated (2 SIMMs, 8 Mbyte this row)
DRB1 = 04h	populated (2 SIMMs, 8 Mbyte this row)
DRB2 = 04h	empty row
DRB3 = 04h	empty row
DRB4 = 04h	empty row

Example #2

The memory array is populated with two 2-Mbyte x 32 double-sided SIMMs (one row), and four 4-Mbyte x 32 single-sided SIMMs (two rows), yielding a total of 96 Mbytes of DRAM. The DRB Registers are programmed as follows:

DRB0 = 04h	populated with 16 Mbytes, ½ of double-sided SIMMs
DRB1 = 08h	the other 16 Mbytes of the double-sided SIMMs
DRB2 = 10h	populated with 32 Mbytes, one of the sided SIMMs
DRB3 = 18h	the other 32 Mbytes of single-sided SIMMs
DRB4 = 18h	empty row

3.2.16 DRT—DRAM ROW TYPE REGISTER

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

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

Bit	Description
7:5	**Reserved.**
4:0	**DRAM Row Type (DRT[4:0]):** Each bit in this field corresponds to the DRAM row identified by the corresponding DRB Register. Thus, DRT0 corresponds to row 0, DRT1 to row 1, etc. When DRTx = 0, page mode DRAM timings are used for that bank. When DRTx = 1, EDO DRAM timings are used for that bank.

3.2.17 SMRAM—SYSTEM MANAGEMENT RAM CONTROL REGISTER

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

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

Bit	Description
7	**Reserved.**
6	**SMM Space Open (DOPEN):** When DOPEN = 1 and DLCK = 0, SMM space DRAM is made visible, even when SMIACT# is negated. This is intended to help BIOS initialize SMM space. Software should ensure that DOPEN = 1 is mutually exclusive with DCLS = 1. When DLCK is set to 1, DOPEN is set to 0 and becomes read only.
5	**SMM Space Closed (DCLS):** When DCLS = 1, SMM space DRAM is not accessible to data references, even if SMIACT# is asserted. Code references may still access SMM space DRAM. This allows SMM software to reference "through" SMM space to update the display, even when SMM space is mapped over the VGA range. Software should ensure that DOPEN = 1 is mutually exclusive with DCLS = 1.
4	**SMM Space Locked (DLCK):** When DLCK is set to 1, the TSC sets DOPEN to 0 and both DLCK and DOPEN become read only. DLCK can be set to 1 via a normal configuration space write but can only be cleared by a power-on reset. The combination of DLCK and DOPEN provide convenience with security. The BIOS can use the DOPEN function to initialize SMM space and then use DLCK to "lock down" SMM space in the future so that no application software (or BIOS itself) can violate the integrity of SMM space, even if the program has knowledge of the DOPEN function.
3	**SMRAM Enable (SMRAME):** When SMRAME = 1, the SMRAM function is enabled, providing 128 Kbytes of DRAM accessible at the A0000h address while in SMM (ADS# with SMIACT#).
2:0	**SMM Space Base Segment (DBASESEG):** This field selects the location of SMM space. "SMM DRAM" is not remapped. It is simply "made visible" if the conditions are right to access SMM space. Otherwise, the access is forwarded to PCI. DBASESEG = 010 is the only allowable setting and selects the SMM space as A0000–BFFFFh. All other values are reserved. PCI masters are not allowed access to SMM space.

Table 6 summarizes the operation of SMRAM space cycles targeting SMI space addresses (A and B segments):

Table 6. SMRAM Space Cycles

SMRAME	DLCK	DCLS	DOPEN	SMIACT#	Code Fetch	Data Reference
0	X	X	X	X	PCI	PCI
1	0	0	0	0	DRAM	DRAM
1	0	X	0	1	PCI	PCI
1	0	0	1	X	DRAM	DRAM
1	0	1	0	0	DRAM	PCI
1	0	1	1	X	INVALID	INVALID
1	1	0	X	0	DRAM	DRAM
1	1	X	X	1	PCI	PCI
1	1	1	X	0	DRAM	PCI

4.0 FUNCTIONAL DESCRIPTION

This section provides a functional description of the TSC and TDP.

4.1 Host Interface

The Host Interface of the TSC is designed to support the Pentium processor. The host interface of the TSC supports 50 MHz, 60 MHz, and 66 MHz bus speeds. The 82430FX PCIset supports the Pentium processor with a full 64-bit data bus, 32-bit address bus, and associated internal write-back cache logic. Host bus addresses are decoded by the TSC for accesses to main memory, PCI memory, and PCI I/O. The TSC also supports the pipelined addressing capability of the Pentium processor.

4.2 PCI Interface

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

The TSC/TDPs integrate posted write buffers for CPU memory writes to PCI. Back-to-back sequential memory writes to PCI are converted to burst writes on PCI. This feature allows the CPU to continue posting dword writes at the maximum bandwidth for the Pentium processor for the highest possible transfer rates to the graphics frame buffer.

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

4.3 Secondary Cache Interface

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

The second level cache can be configured for either 256-Kbyte or 512-Kbyte cache sizes using either synchronous burst or pipelined burst SRAMs, or standard asynchronous SRAMs. For the 256-Kbyte configurations, an 8kx8 standard SRAM is used to store the tags. For the 512-Kbyte configurations, a 16kx8 standard SRAM is used to store the tags and the valid bits. A 5V SRAM is used for the Tag.

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

Table 7. SRAM Access Time Requirements

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

4.3.1 CLOCK LATENCIES

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

Table 8. Second Level Cache Latencies with Standard SRAM

Cycle Type	HCLK Count
Burst Read	3-2-2-2
Burst Write (Write Back)	4-3-3-3
Single Read	3
Single Write	4

Table 9. Second Level Cache Latencies with Burst SRAM

Cycle Type	HCLK Count
Burst Read	3-1-1-1
Burst Write (Write Back)	3-1-1-1
Single Read	3
Single Write	3
Pipelined Back-to-Back Burst Reads	3-1-1-1, 1-1-1-1

4.3.2 SNOOP CYCLES

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

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

Reads

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

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

Writes

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

4.3.3 CACHE ORGANIZATION

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

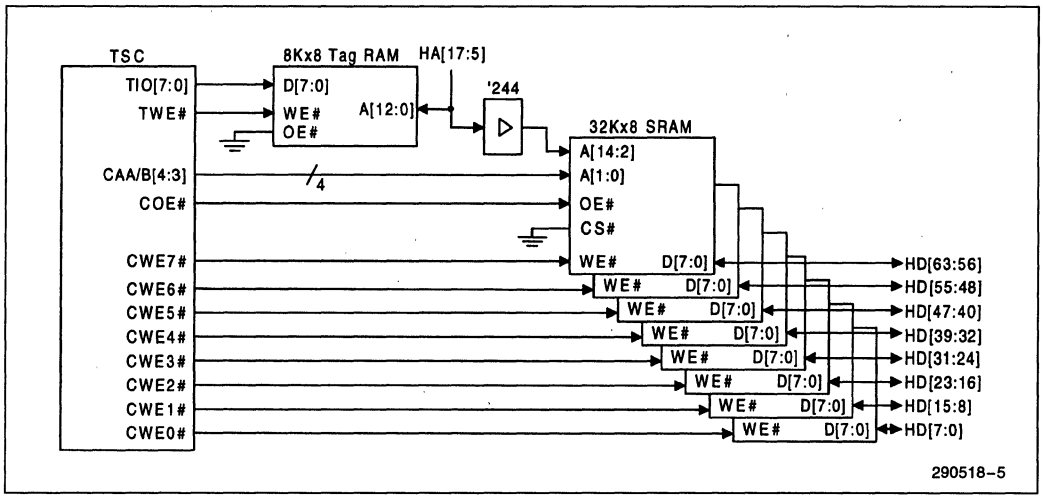


Figure 3. 256-Kbyte Second Level Cache (Standard SRAM)

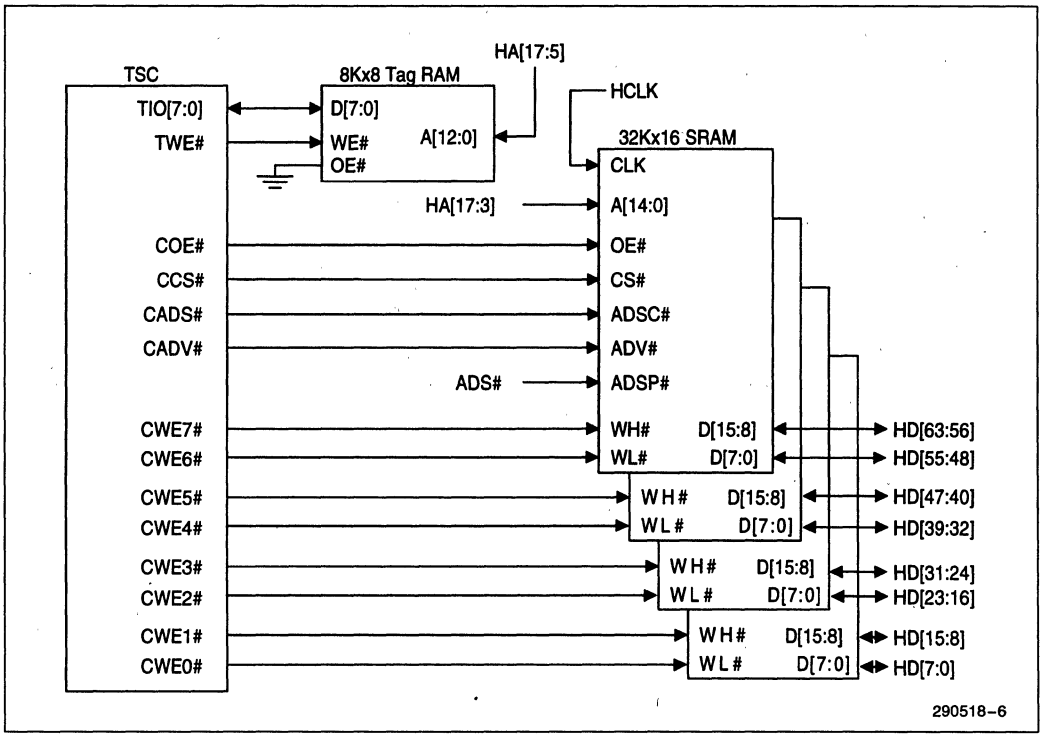


Figure 4. 256-Kbyte Second Level Cache (Burst SRAM)

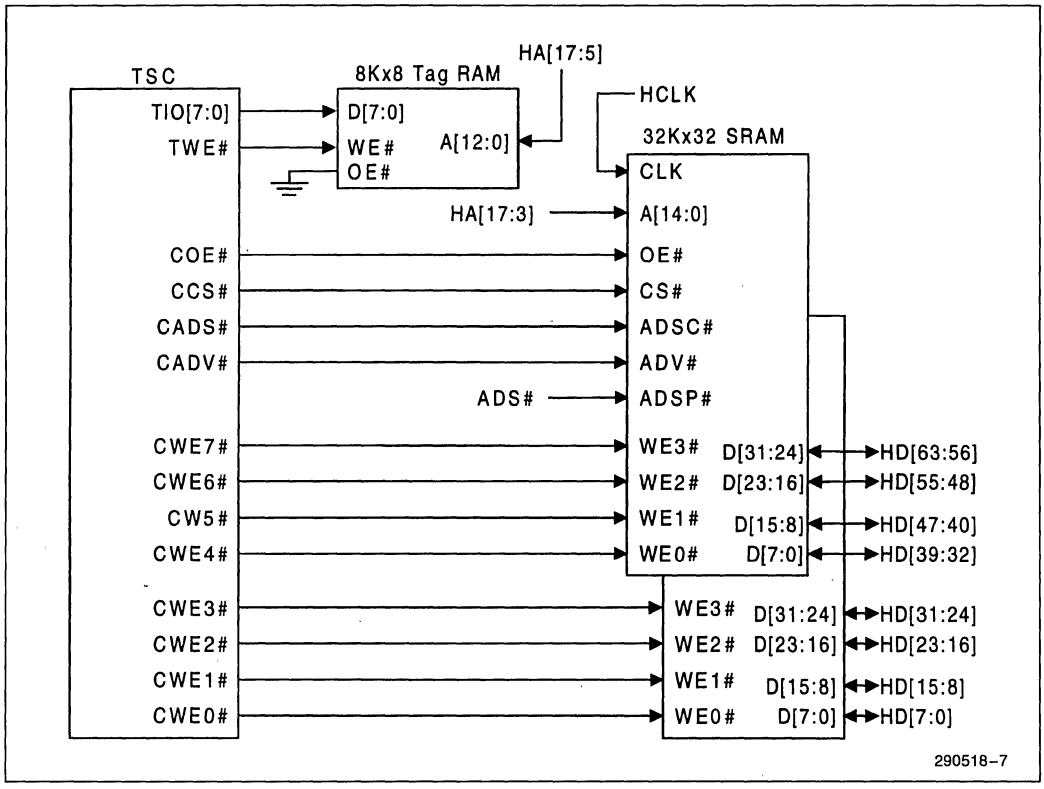


Figure 5. 256-Kbyte Second Level Cache (Burst SRAM)

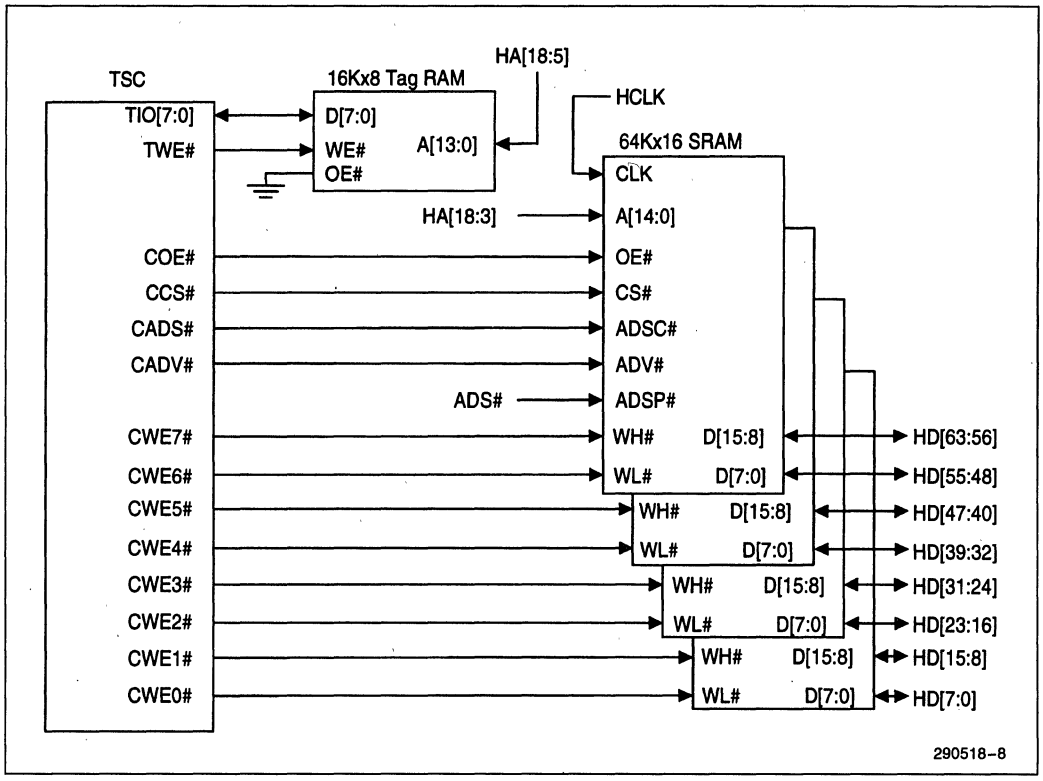


Figure 6. 512-Kbyte Second Level Cache (Burst SRAM)

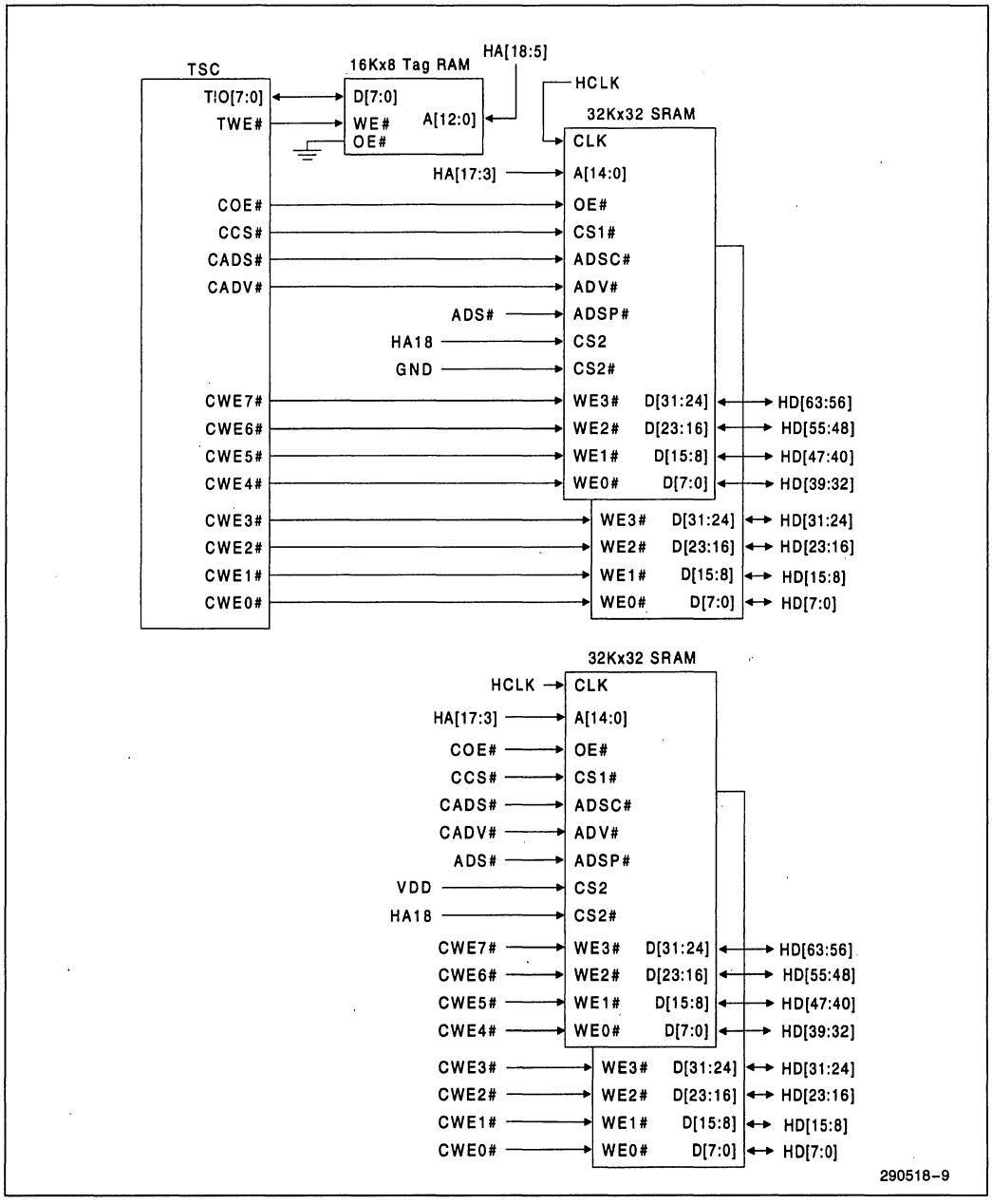


Figure 7. Two Bank 512-Kbyte Second Level Cache (Pipelined Burst SRAM)

4.4 DRAM Interface

The 82430FX PCIset's main memory DRAM interface supports a 64-bit wide memory array and main memory sizes from 4 to 128 Mbytes. The TSC generates the RAS#, CAS#, WE# (using MOE#) and multiplexed addresses for the DRAM array and controls the data flow through the 82438FX TDP's. For CPU-to-DRAM cycles the address flows through the TSC and data flows through the TDP's. For PCI or ISA cycles to memory the address flows through the TSC and data flows to the TDP's through the TSC and PLINK bus. The TSC and TDP DRAM interfaces are synchronous to the CPU clock.

The 82430FX PCIset supports industry standard 32-bit wide memory modules with fast page-mode DRAMs and EDO (Extended Data Out) DRAMs (also known as Hyper Page mode). With twelve multiplexed address lines (MA[11:0]), the TSC supports 512Kx32, 1M32, 2Mx32, and 4Mx32 SIMM's (both symmetrical and asymmetrical addressing). Five RAS# lines permit up to five rows of DRAM and eight CAS# lines provide byte write control over the array. The TSC supports 60 ns and 70 ns DRAMs (both single and double-sided SIMM's). The TSC also provides an automatic RAS# only refresh, at a rate of 1 refresh per 15.6 ms at 66 MHz, 60 MHz, and 50 MHz. A refresh priority queue and "smart refresh" algorithm are used to minimize the performance impact due to refresh.

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

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

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

The SMRAM memory space is controlled by the SMRAM Control Register. This register selects if the SMRAM space is enabled, opened, closed, or locked. SMRAM space is between 640 Kbytes and 768 Kbytes. See section 3.2.17 for more details on SMRAM.

4.4.1 DRAM ORGANIZATION

Figure 8 illustrates a 4-SIMM configuration that supports 4 double-sided SIMM's and motherboard DRAM. RAS0# is used for motherboard memory. This memory should not be implemented with SIMMs.

Except for motherboard memory, a row in the DRAM array is made up of two SIMM's that share a common RAS# line. Within any given row, the two SIMMs must be the same size. For different rows, SIMM densities can be mixed in any order. Each row is controlled by 8 CAS lines. EDO and Standard page mode DRAM's can be mixed between rows or within a row. When DRAM types are mixed (EDO and standard page mode), each row will run optimized for that particular type of DRAM. If DRAM types are mixed within a row, page mode timings must be selected.

SIMMs can be used for the sockets connected to RAS[2:1]# and RAS[4:3]#. The two RAS lines permit double-sided SIMMs to be used in these socket pairs. The following rules apply to the SIMM configuration.

1. SIMM sockets can be populated in any order.
2. SIMM socket pairs need to be populated with the same densities. For example, SIMM sockets RAS[2:1]# should be populated with identical densities. However, SIMM sockets using RAS[4:3]# can be populated with different densities than the SIMM socket pair using RAS[2:1]#.

3. The TSC only recognizes a maximum of 128 Mbytes of main memory, even if populated with more memory.
4. EDO's and standard page mode can both be used.

4.4.2 MAIN MEMORY ADDRESS MAP

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

4.4.3 DRAM ADDRESS TRANSLATION

The multiplexed row/column address to the DRAM memory array is provided by MA[11:0] which are derived from the host address bus or PCI address as defined by Table 10. The TSC has a 4-Kbyte page size. The page offset address is driven on the MA[8:0] lines when driving the column address. The MA[11:0] lines are translated from the address lines A[24:3] for all memory accesses.

Table 10. DRAM Address Translation

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

The types of DRAMs depth configuration supported are:

Depth	Row Width	Column Width
512K	10	9
1M	10	10
2M	11	10
4M	11	11
4M	12	10

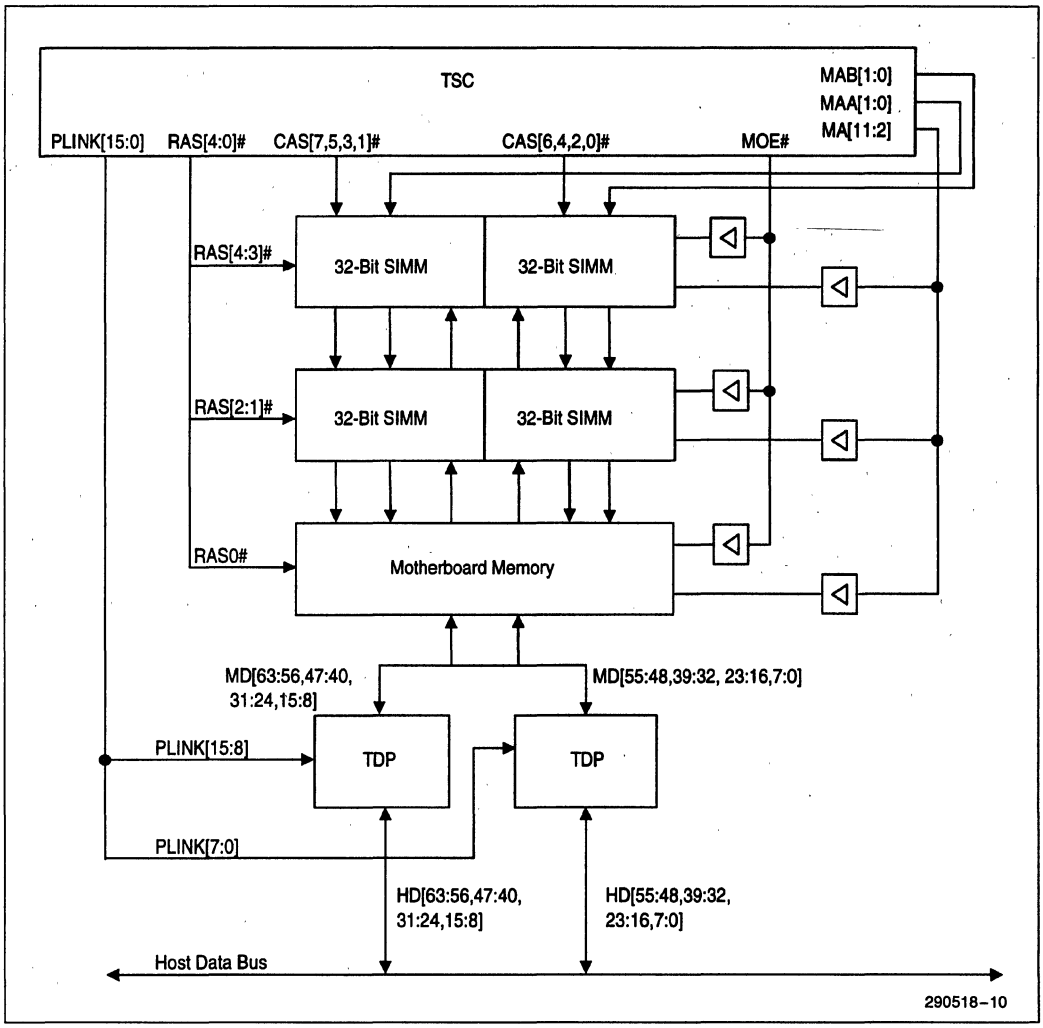


Figure 8. DRAM Array Connections

4.4.4 DRAM PAGE MODE

For any row containing standard page mode DRAM on read cycles, the TSC keeps CAS[7:0]# asserted until data is sampled by the TDPs.

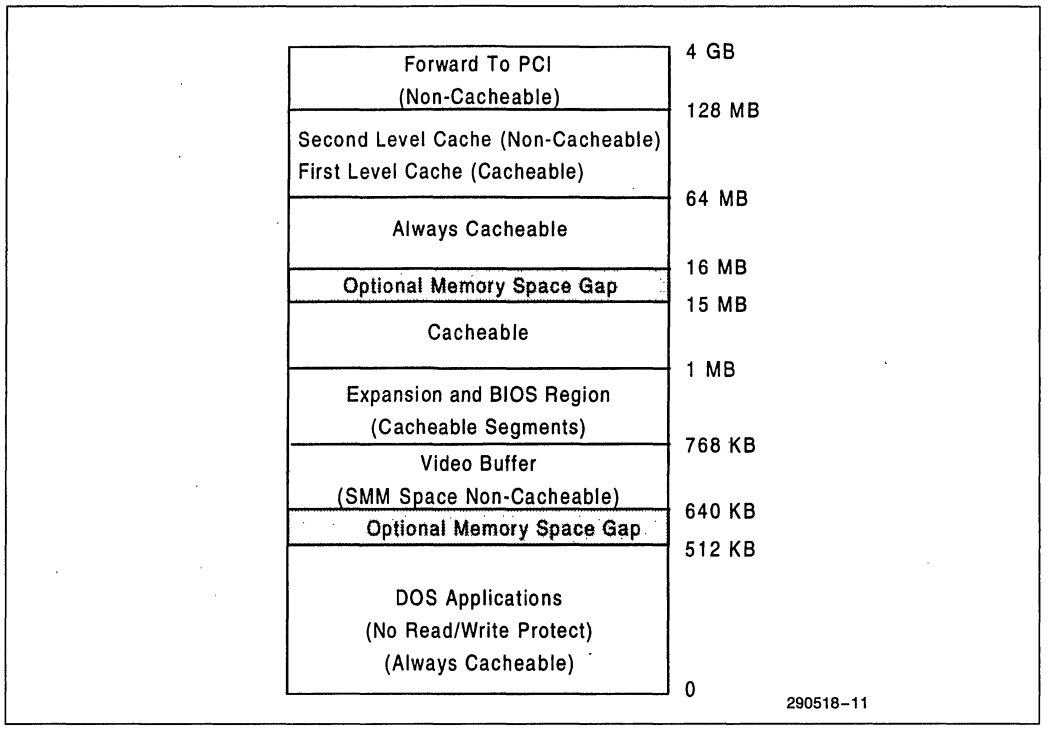


Figure 9. Memory Space Organization

4.4.5 EDO MODE

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

The EDO Detect Mode Enable bit in the DRAM Control Register enables a special timing mode for BIOS to detect the DRAM type on a row by row basis.

4.4.6 DRAM PERFORMANCE

The DRAM performance is controlled by the DRAM Timing Register, processor pipelining, and by the type of DRAM used (EDO or standard page mode). Table 11 lists both EDO and standard page mode optimum timings.

Table 11. CPU to DRAM Performance Summary

Processor Cycle Type (Pipelined)	Clock Count (ADS# to BRDY#)	Comments
Burst read page hit	7-2-2-2	EDO
Read row miss	9-2-2-2 (note 1)	EDO
Read page miss	12-2-2-2	EDO
Back-to-back burst reads page hit	7-2-2-2-3-2-2-2	EDO
Burst read page hit	7-3-3-3	Standard page mode
Burst read row miss	9-3-3-3 (note 1)	Standard page mode
Burst read page miss	12-3-3-3	Standard page mode
Back-to-back burst read page hit	7-3-3-3-3-3-3-3	Standard page mode
Posted write	3-1-1-1	EDO/Standard page mode
Retire data for posted write	Every 3 clocks	EDO/Standard page mode

NOTES:
1. Due to the MA[11:2] to RAS# setup requirements, if a page is open, two clocks are added to the leadoff.
2. Read and write rates to DRAM are programmable via the DRAMT Register.

4.4.7 DRAM REFRESH

The TSC supports RAS# only refresh and generates refresh requests. The rate that requests are generated is determined by the DRAM Control Register. When a refresh request is generated, the request is placed in a four entry queue. The DRAM controller services a refresh request when the refresh queue is not empty and the controller has no other requests pending. When the refresh queue is full, refresh becomes the highest priority request and is serviced next by the DRAM controller, regardless of other pending requests. When the DRAM controller begins to service a refresh request, the request is removed from the refresh queue.

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

4.4.8 SYSTEM MANAGEMENT RAM

The 82430FX PCIset support the use of main memory as System Management RAM (SMRAM), enabling the use of System Management Mode. When this function is disabled, the TSC memory map is defined by the DRB and PAM Registers. When SMRAM is enabled, the TSC reserves the A and B segments of main memory for use as SMRAM.

SMRAM is placed at A0000-BFFFFh via the SMRAM Space Register. Enhanced SMRAM features can also be enabled via this register. PCI masters can not access SMRAM when it is programmed to the A and B segments.

When the TSC detects a CPU stop grant special cycle, it generates a PCI Stop Grant Special cycle with 0002h in the message field (AD[15:0]) and 0012h in the message dependent data field (AD[31:16]) during the first data phase (IRDY# asserted).

4.5 82438FX Data Path (TDP)

The TDP's provide the data path for host-to-main memory, PCI-to-main memory, and host-to-PCI cycles. Two TDP's are required for the 82430FX PCIset system configuration. The TSC controls the data flow through the TDP's with the PCMD[1:0], HOE#, POE#, MOE#, MSTB#, and MADV# signals.

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

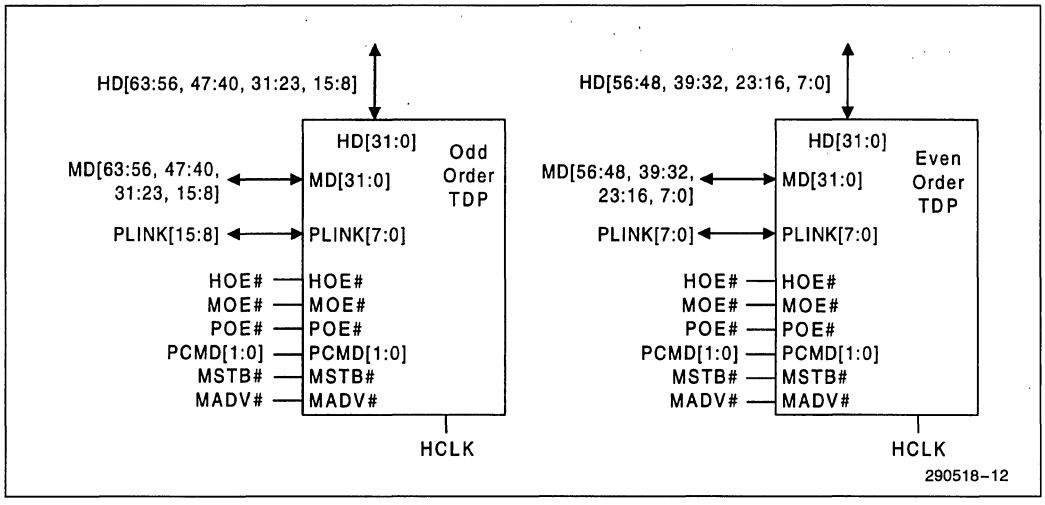


Figure 10. TDP 64-Bit Data Path Partitioning

4.6 PCI Bus Arbitration

The TSC's PCI Bus arbiter allows PCI peer-to-peer traffic concurrent with CPU main memory/second level cache cycles. The arbiter supports five PCI masters (Figure 11). REQ[3:0]#/GNT[3:0]# are used by PCI masters other than the PCI-to-ISA expansion bridge (PIIX). PHLD#/PHLDA# are the arbitration request/grant signals for the PIIX and provide guaranteed access time capability for ISA masters. PHLD#/PHLDA# also optimize system performance based on PIIX known policies.

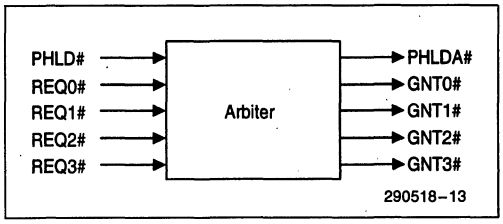


Figure 11. Arbiter

4.6.1 PRIORITY SCHEME AND BUS GRANT

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

A rotating priority scheme is used to determine the highest priority requester in the case of simultaneous requests. If the highest priority input at arbitration time does not have an active request, the next priority active requester is granted the bus. Granting the bus to a lower priority requester does not change the rotation order, but it does advance the priority rotation. The rotation priority chain is fixed (Figure 12). If the highest priority agent does not request the bus, the next agent in the chain is the highest priority, and so forth down the chain.

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

The grant signals (GNTx#) are normally negated after FRAME# assertion or 16 PCLKs from grant assertion, if no cycle has started. Once asserted, PHLDA# is only negated after PHLD# has been negated.

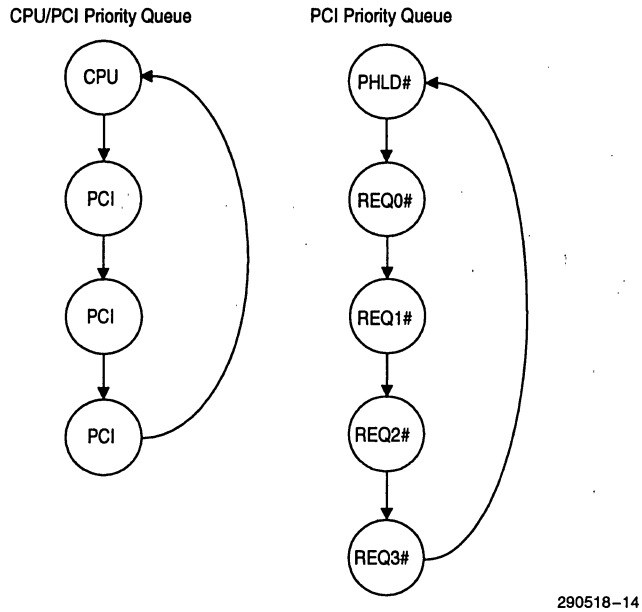
4.6.2 CPU POLICIES

The CPU never explicitly requests the bus. Instead, the arbiter grants the bus to the CPU when:

- the CPU is the highest priority
- PCI agents do not require main memory (peer-to-peer transfers or bus idle) and the PCI Bus is not currently locked by a PCI master

When the CPU is granted as highest priority, the MLT timer is used to guarantee a minimum amount of system resources to the CPU before another requesting PCI agent is granted.

An AHOLD mechanism controls granting the bus to the CPU.



NOTES:
1. For the CPU/PCI priority queue, the CPU is granted the PCI Bus after three consecutive PCI grants.
2. For the PCI priority queue, the last agent granted is always dropped to the bottom of the queue for the next arbitration cycle. However, the order of the chain is always preserved.

Figure 12. Arbitration Priority Rotation

4.7 Clock Generation and Distribution

The TSC and CPU should be clocked from one clock driver output to minimize skew between the CPU and TSC. The TDPs should share another clock driver output.

4.7.1 RESET SEQUENCING

The TSC is asynchronously reset by the PCI reset (RST#). After RST# is negated, the TSC resets the TDP by driving HOE#, MOE#, and POE# to 1 for two HCLKs. The TSC changes HOE#, MOE#, and POE# to their default value after the TDP is reset.

Arbiter (Central Resource) Functions on Reset

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

5.0 PINOUT AND PACKAGE INFORMATION

5.1 82437FX Pinout

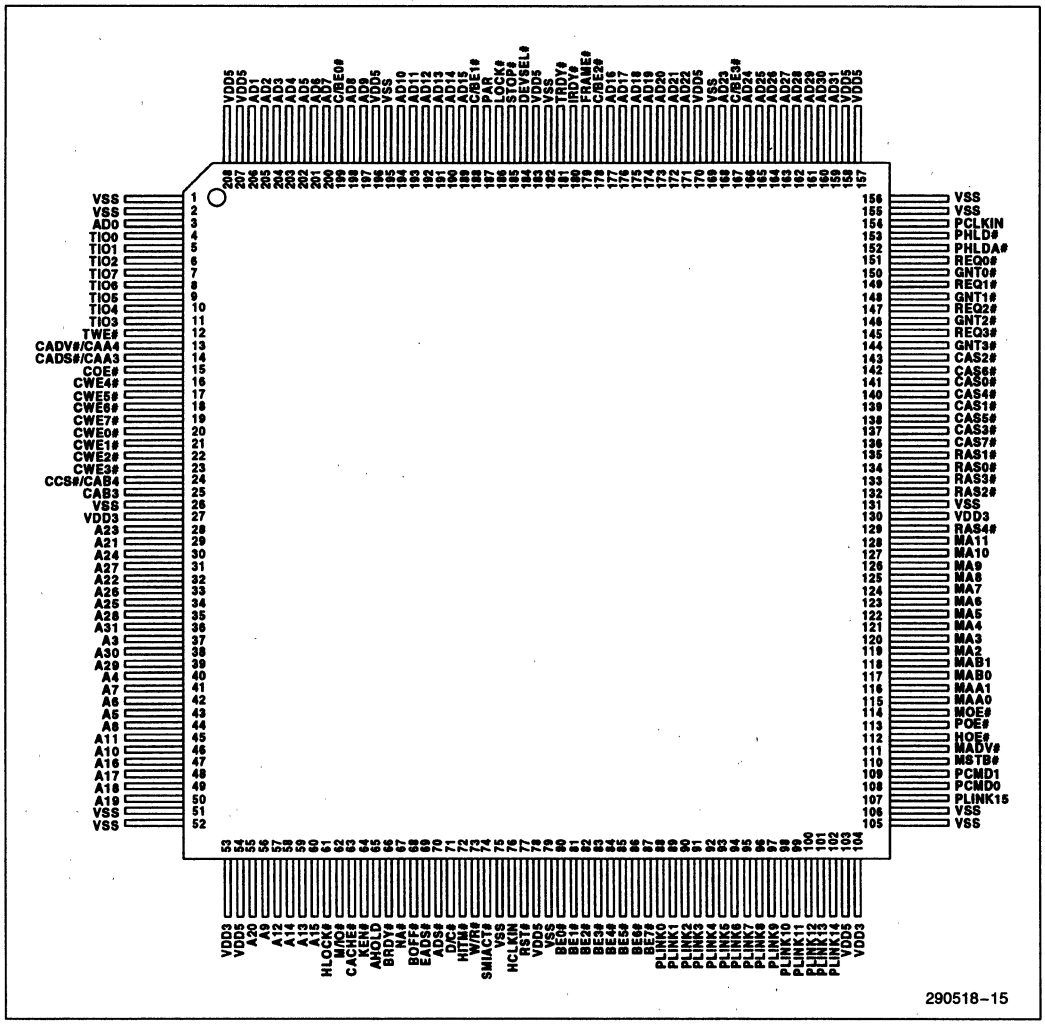


Figure 13. 82437FX Pin Assignment

Table 12. 82437FX Alphabetical Pin Assignment

Name	Pin#	Type
A3	37	I/O
A4	40	I/O
A5	43	I/O
A6	42	I/O
A7	41	I/O
A8	44	I/O
A9	56	I/O
A10	46	I/O
A11	45	I/O
A12	57	I/O
A13	59	I/O
A14	58	I/O
A15	60	I/O
A16	47	I/O
A17	48	I/O
A18	49	I/O
A19	50	I/O
A20	55	I/O
A21	29	I/O
A22	32	I/O
A23	28	I/O
A24	30	I/O
A25	34	I/O
A26	33	I/O
A27	31	I/O
A28	35	I/O
A29	39	I/O
A30	38	I/O
A31	36	I/O

Name	Pin#	Type
AD0	3	I/O
AD1	206	I/O
AD2	205	I/O
AD3	204	I/O
AD4	203	I/O
AD5	202	I/O
AD6	201	I/O
AD7	200	I/O
AD8	198	I/O
AD9	197	I/O
AD10	194	I/O
AD11	193	I/O
AD12	192	I/O
AD13	191	I/O
AD14	190	I/O
AD15	189	I/O
AD16	177	I/O
AD17	176	I/O
AD18	175	I/O
AD19	174	I/O
AD20	173	I/O
AD21	172	I/O
AD22	171	I/O
AD23	168	I/O
AD24	166	I/O
AD25	165	I/O
AD26	164	I/O
AD27	163	I/O
AD28	162	I/O

Name	Pin#	Type
AD29	161	I/O
AD30	160	I/O
AD31	159	I/O
ADS#	70	I
AHOLD	65	O
BE0#	80	I
BE1#	81	I
BE2#	82	I
BE3#	83	I
BE4#	84	I
BE5#	85	I
BE6#	86	I
BE7#	87	I
BOFF#	68	O
BRDY#	66	O
C/BE0#	199	I/O
C/BE1#	188	I/O
C/BE2#	178	I/O
C/BE3#	167	I/O
CAB3	25	O
CACHE#	63	I
CADS#/ CAA3	14	O
CADV#/ CAA4	13	O
CAS0#	141	O
CAS1#	139	O
CAS2#	143	O
CAS3#	137	O
CAS4#	140	O

Table 12. 82437FX Alphabetical Pin Assignment (Continued)

Name	Pin#	Type
CAS5#	138	O
CAS6#	142	O
CAS7#	136	O
CCS#/ CAB4	24	O
COE#	15	O
CWE0#	20	O
CWE1#	21	O
CWE2#	22	O
CWE3#	23	O
CWE4#	16	O
CWE5#	17	O
CWE6#	18	O
CWE7#	19	O
D/C#	71	I
DEVSEL#	184	I/O
EADS#	69	O
FRAME#	179	I/O
GNT0#	150	O
GNT1#	148	O
GNT2#	146	O
GNT3#	144	O
HCLKIN	76	I
HITM#	72	I
HLOCK#	61	I
HOE#	112	O
IRDY#	180	I/O
KEN#	64	O
LOCK#	186	I/O
M/IO#	62	I
MA2	119	O

Name	Pin#	Type
MA3	120	O
MA4	121	O
MA5	122	O
MA6	123	O
MA7	124	O
MA8	125	O
MA9	126	O
MA10	127	O
MA11	128	O
MAA0	115	O
MAA1	116	O
MAB0	117	O
MAB1	118	O
MADV#	111	O
MOE#	114	O
MSTB#	110	O
NA#	67	O
PAR	187	I/O
PCLKIN	154	I
PCMD0	108	O
PCMD1	109	O
PHLD#	153	I
PHLDA#	152	O
PLINK0	88	I/O
PLINK1	89	I/O
PLINK2	90	I/O
PLINK3	91	I/O
PLINK4	92	I/O
PLINK5	93	I/O
PLINK6	94	I/O
PLINK7	95	I/O

Name	Pin#	Type
PLINK8	96	I/O
PLINK9	97	I/O
PLINK10	98	I/O
PLINK11	99	I/O
PLINK12	100	I/O
PLINK13	101	I/O
PLINK14	102	I/O
PLINK15	107	I/O
POE#	113	O
RAS0#	134	O
RAS1#	135	O
RAS2#	132	O
RAS3#	133	O
RAS4#	129	O
REQ0#	151	I
REQ1#	149	I
REQ2#	147	I
REQ3#	145	I
RST#	77	I
SMIACT#	74	I
STOP#	185	I/O
TIO0	4	I/O
TIO1	5	I/O
TIO2	6	I/O
TIO3	11	I/O
TIO4	10	I/O
TIO5	9	I/O
TIO6	8	I/O
TIO7	7	I/O
TRDY#	181	I/O
TWE#	12	O

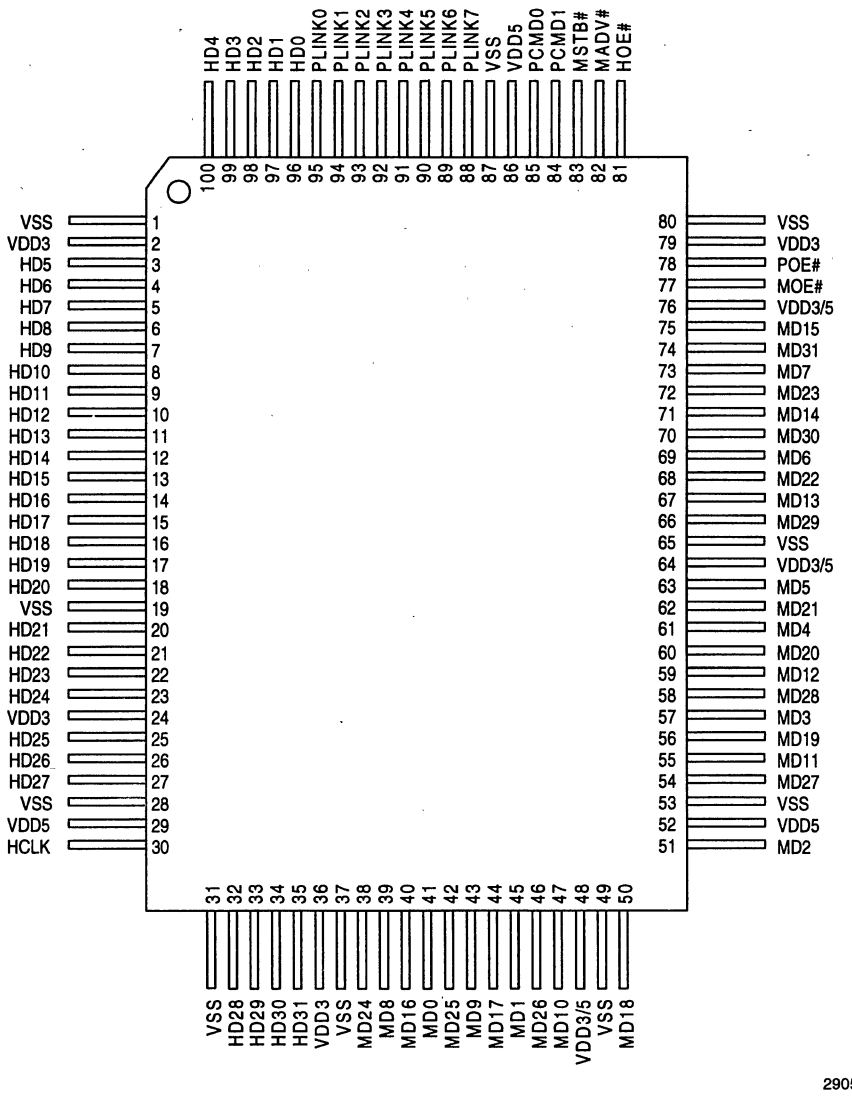
Table 12. 82437FX Alphabetical Pin Assignment (Continued)

Name	Pin#	Type
VDD3	27	V
VDD3	53	V
VDD3	104	V
VDD3	130	V
VDD5	54	V
VDD5	78	V
VDD5	103	V
VDD5	157	V
VDD5	158	V
VDD5	170	V

Name	Pin#	Type
VDD5	183	V
VDD5	196	V
VDD5	207	V
VDD5	208	V
VSS	1	V
VSS	2	V
VSS	26	V
VSS	51	V
VSS	52	V
VSS	75	V

Name	Pin#	Type
VSS	79	V
VSS	105	V
VSS	106	V
VSS	131	V
VSS	155	V
VSS	156	V
VSS	169	V
VSS	182	V
VSS	195	V
W/R#	73	I

5.2 82438FX Pinout



290518-16

NOTES:
VDD3 is connected to the 3.3V rail; VDD5 is connected to the 5V, VDD3/5 is connected to the 3.3V rail for 3.3V DRAMs and to the 5V rail for 5V DRAMs.

Figure 14. 82438FX Pin Assignment

Table 13. 82438FX Alphabetical Pin Assignment

Name	Pin#	Type
HCLK	30	I
HD0	96	I/O
HD1	97	I/O
HD2	98	I/O
HD3	99	I/O
HD4	100	I/O
HD5	3	I/O
HD6	4	I/O
HD7	5	I/O
HD8	6	I/O
HD9	7	I/O
HD10	8	I/O
HD11	9	I/O
HD12	10	I/O
HD13	11	I/O
HD14	12	I/O
HD15	13	I/O
HD16	14	I/O
HD17	15	I/O
HD18	16	I/O
HD19	17	I/O
HD20	18	I/O
HD21	20	I/O
HD22	21	I/O
HD23	22	I/O
HD24	23	I/O
HD25	25	I/O
HD26	26	I/O
HD27	27	I/O
HD28	32	I/O
HD29	33	I/O
HD30	34	I/O
HD31	35	I/O
HOE#	81	I
MADV#	82	I
MD0	41	I/O
MD1	45	I/O
MD2	51	I/O
MD3	57	I/O
MD4	61	I/O
MD5	63	I/O
MD6	69	I/O
MD7	73	I/O
MD8	39	I/O
MD9	43	I/O
MD10	47	I/O
MD11	55	I/O
MD12	59	I/O
MD13	67	I/O
MD14	71	I/O
MD15	75	I/O
MD16	40	I/O
MD17	44	I/O
MD18	50	I/O
MD19	56	I/O
MD20	60	I/O
MD21	62	I/O
MD22	68	I/O
MD23	72	I/O
MD24	38	I/O
MD25	42	I/O
MD26	46	I/O
MD27	54	I/O
MD28	58	I/O
MD29	66	I/O
MD30	70	I/O
MD31	74	I/O
MOE#	77	I
MSTB#	83	I
PCMD0	85	I
PCMD1	84	I
PLINK0	95	I/O
PLINK1	94	I/O
PLINK2	93	I/O
PLINK3	92	I/O
PLINK4	91	I/O
PLINK5	90	I/O
PLINK6	89	I/O
PLINK7	88	I/O
POE#	78	I
VDD3	2	V
VDD3	24	V
VDD5	29	V
VDD3	36	V
VDD3/5	48	V
VDD5	52	V
VDD3/5	64	V
VDD3/5	76	V
VDD3	79	V
VDD5	86	V

Table 13. 82438FX Alphabetical Pin Assignment (Continued)

Name	Pin#	Type
VSS	1	V
VSS	19	V
VSS	28	V
VSS	31	V

Name	Pin#	Type
VSS	37	V
VSS	49	V
VSS	53	V
VSS	65	V

Name	Pin#	Type
VSS	80	V
VSS	87	V

5.3 82437FX Package Dimensions

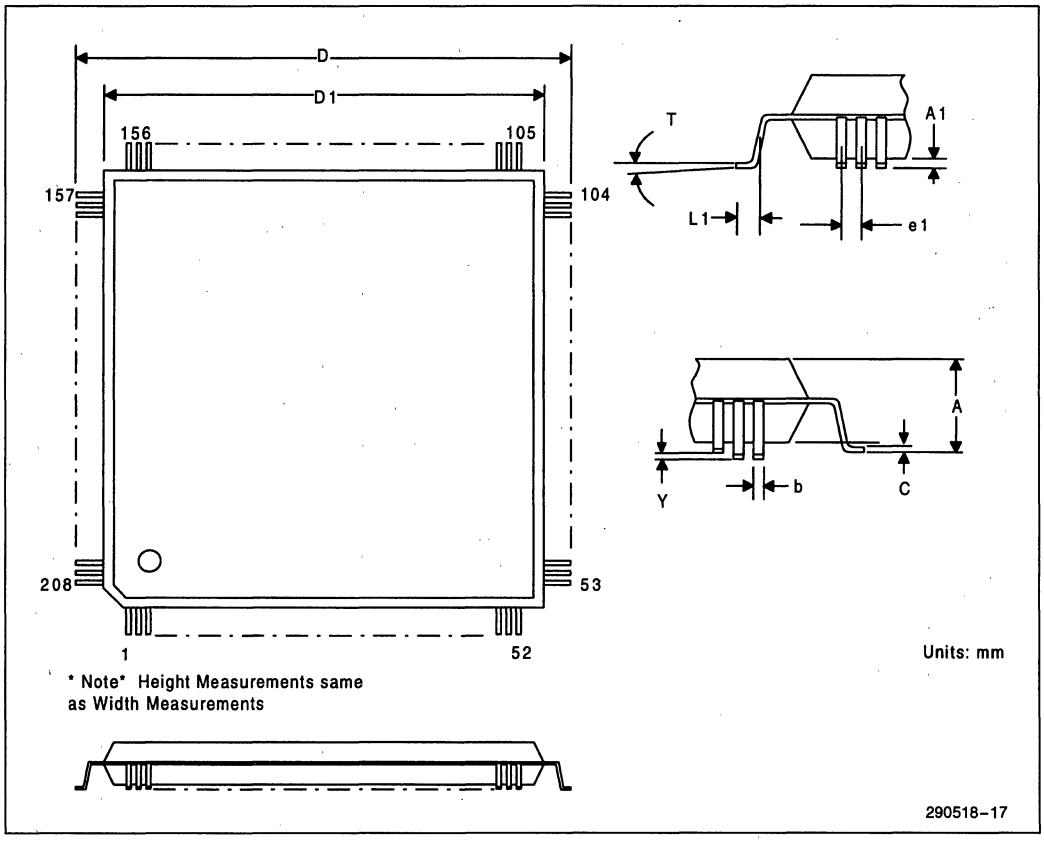


Figure 15. 208 Pin Quad Flat Pack (QFP) Dimensions

Table 14. 208 Pin Quad Flat Pack (QFP) Dimensions

Symbol	Description	Value (mm)
A	Seating Height	4.25 (max)
A1	Stand-off	0.05 (min); 0.40 (max)
b	Lead Width	0.2 ± 0.10
C	Lead Thickness	0.15 +0.1/-0.05
D	Package Length and Width, including pins	30.6 ± 0.4
D1	Package Length and Width, excluding pins	28 ± 0.2
e1	Linear Lead Pitch	0.5 ± 0.1
Y	Lead Coplanarity	0.08 (max)
L1	Foot Length	0.5 ± 0.2
T	Lead Angle	0° - 10°

5.4 82438FX Package Dimensions

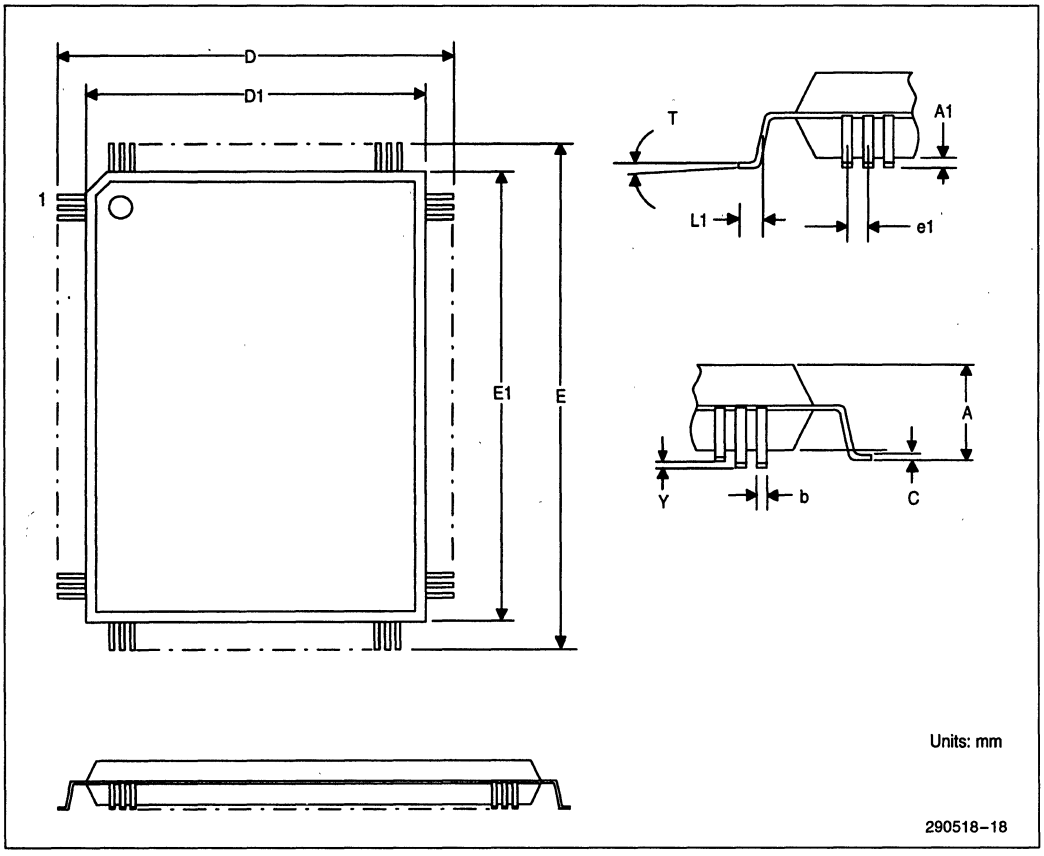


Figure 16. 100 Pin Plastic Quad Flat Pack (PQFP) Dimensions

Table 15. 100 Pin Quad Flat Pack (QFP) Dimensions

Symbol	Description	Value (mm)
A	Seating Height	3.3 (max)
A1	Stand-off	0.0 (min); 0.50 (max)
b	Lead Width	0.3 ± 0.10
C	Lead Thickness	0.15 +0.1/-0.05
D	Package Width, including pins	17.9 ± 0.4
D1	Package Width, excluding pins	14 0.2
E	Package Length, including pins	23.9 ± 0.4
E1	Package Length, excluding pins	20 ± 0.2
e1	Linear Lead Pitch	0.65 ± 0.12
Y	Lead Coplanarity	0.1 (max)
L1	Foot Length	0.8 0.2
T	Lead Angle	0° - 10°

6.0 82437FX TSC TESTABILITY

6.1 Test Mode Description

The test modes are decoded from the REQ#[3:0] and qualified with the RESET# pin. Test mode selection is asynchronous, these signals need to remain in their respective state for the duration of the test modes. The test modes are defined as follows.

Test Mode	RST#	REQ0#	REQ1#	REQ2#	REQ3#
NAND Tree	0	0	0	0	0

6.2 NAND Tree Mode

Tri-states all outputs and bi-directional buffers except for RST#, REQ#[3:0], GNT#[3:1]. The NAND tree follows the pins sequentially around the chip skipping only RESET#, REQ#[3:0], and GNT#[3:1]. The first input of the NAND chain GNT0#, and the NAND chain is routed counter-clockwise around the chip (e.g., GNT0#, PHLDA#, . . .). The only valid outputs during NAND tree mode are GNT1#, GNT2#, and GNT3#. GNT1# and GNT#3 are both final outputs of the NAND tree, and GNT2# is the halfway point of the NAND tree.

To perform a NAND tree test, all pins included in the NAND tree should be driven to 1 with the exception of PCLKIN which should be driven to 0.

Beginning with GNT0# and working counter-clockwise around the chip, each pin can be toggled with a resulting toggle observed on GNT3#, GNT2#, and GNT1#. The GNT2# output is provided so that the NAND tree test can be divided into two sections.

Table 16. NAND Tree Cell Order for the 82437FX

Pin #	Pin Name	Notes
77	RST#	Must be 0 to enter NAND tree mode. This signal must remain 0 during the entire NAND tree test.
145	REQ3#	Must be 0 to enter NAND tree mode. This signal must remain 0 during the entire NAND tree test.
147	REQ2#	Must be 0 to enter NAND tree mode. This signal must remain 0 during the entire NAND tree test.
149	REQ1#	Must be 0 to enter NAND tree mode. This signal must remain 0 during the entire NAND tree test.
151	REQ0#	Must be 0 to enter NAND tree mode. This signal must remain 0 during the entire NAND tree test.
150	GNT0#	Start of the NAND tree chain.
152	PHLDA#	
153	PHOLD#	
154	PCLKIN	PCLKIN needs to be 0 to pass the NAND-tree chain, a logic 1 will block the NAND-tree chain.
159	AD31	
160	AD30	
161	AD29	
162	AD28	
163	AD27	
164	AD26	
165	AD25	
166	AD24	
167	C/BE3#	
168	AD23	
171	AD22	
172	AD21	
173	AD20	
174	AD19	
175	AD18	
176	AD17	
177	AD16	
178	C/BE2#	
179	FRAME#	
180	IRDY#	
181	TRDY#	
184	DEVSEL#	
185	STOP#	
186	LOCK#	
187	PAR	
188	C/BE1#	
189	AD15	
190	AD14	
191	AD13	
192	AD12	
193	AD11	
194	AD10	
197	AD9	
198	AD8	
199	C/BE0#	
200	AD7	
201	AD6	
202	AD5	
203	AD4	
204	AD3	
205	AD2	

Table 16. NAND Tree Cell Order for the 82437FX (Continued)

Pin #	Pin Name	Notes
206	AD1	
3	AD0	
4	TIO0	
5	TIO1	
6	TIO2	
7	TIO7	
8	TIO6	
9	TIO5	
10	TIO4	
11	TIO3	
12	TWE#	
13	CADV# / CAA4	
14	CADS# / CAA3	
15	COE#	
16	CWE4#	
17	CWE5#	
18	CWE6#	
19	CWE7#	
20	CWE0#	
21	CWE1#	
22	CWE2#	
23	CWE3#	
24	CCS# / CAB4	
25	CAB3	
28	A23	
29	A21	
30	A24	
31	A27	
32	A22	

Pin #	Pin Name	Notes
33	A26	
34	A25	
35	A28	
36	A31	
37	A3	
38	A30	
39	A29	
40	A4	
41	A7	
42	A6	
43	A5	
44	A8	
45	A11	
46	A10	
47	A16	
48	A17	
49	A18	
50	A19	
55	A20	
56	A9	
57	A12	
58	A14	
59	A13	
60	A15	
61	HLOCK#	
62	M/IO#	
63	CACHE#	
64	KEN#	
65	AHOLD	
66	BRDY#	

Table 16. NAND Tree Cell Order for the 82437FX (Continued)

Pin #	Pin Name	Notes
67	NA#	
68	BOFF#	
69	EADS#	
70	ADS#	
71	D/C#	
72	HITM#	
73	W/R#	
74	SMIACT#	
76	HCLKIN	
80	BE0#	
81	BE1#	
82	BE2#	
83	BE3#	
84	BE4#	
85	BE5#	
86	BE6#	
87	BE7#	
88	PLINK0	
89	PLINK1	
90	PLINK2	
91	PLINK3	
92	PLINK4	
93	PLINK5	
94	PLINK6	
95	PLINK7	
96	PLINK8	
97	PLINK9	
98	PLINK10	
99	PLINK11	
100	PLINK12	

Pin #	Pin Name	Notes
101	PLINK13	
102	PLINK14	
107	PLINK15	
108	PCMD0	
109	PCMD1	
110	MSTB#	
111	MADV#	
112	HOE#	
113	POE#	
114	MOE#	
115	MAA0	
116	MAA1	
117	MAB0	
118	MAB1	
119	MA2	
120	MA3	
121	MA4	
122	MA5	
123	MA6	
124	MA7	
125	MA8	
126	MA9	
127	MA10	
128	MA11	
129	RAS4#	
132	RAS2#	
133	RAS3#	
134	RAS0#	
135	RAS1#	
136	CAS7#	

Table 16. NAND Tree Cell Order for the 82437FX (Continued)

Pin #	Pin Name	Notes
137	CAS3#	
138	CAS5#	
139	CAS1#	
140	CAS4#	
141	CAS0#	
142	CAS6#	
143	CAS2#	
144	GNT3#	Final output of the NAND tree chain.
146	GNT2#	Half way point of the NAND tree chain.
148	GNT1#	Final output of the NAND-tree chain.

Figure 17 is a schematic of the NAND tree circuitry.

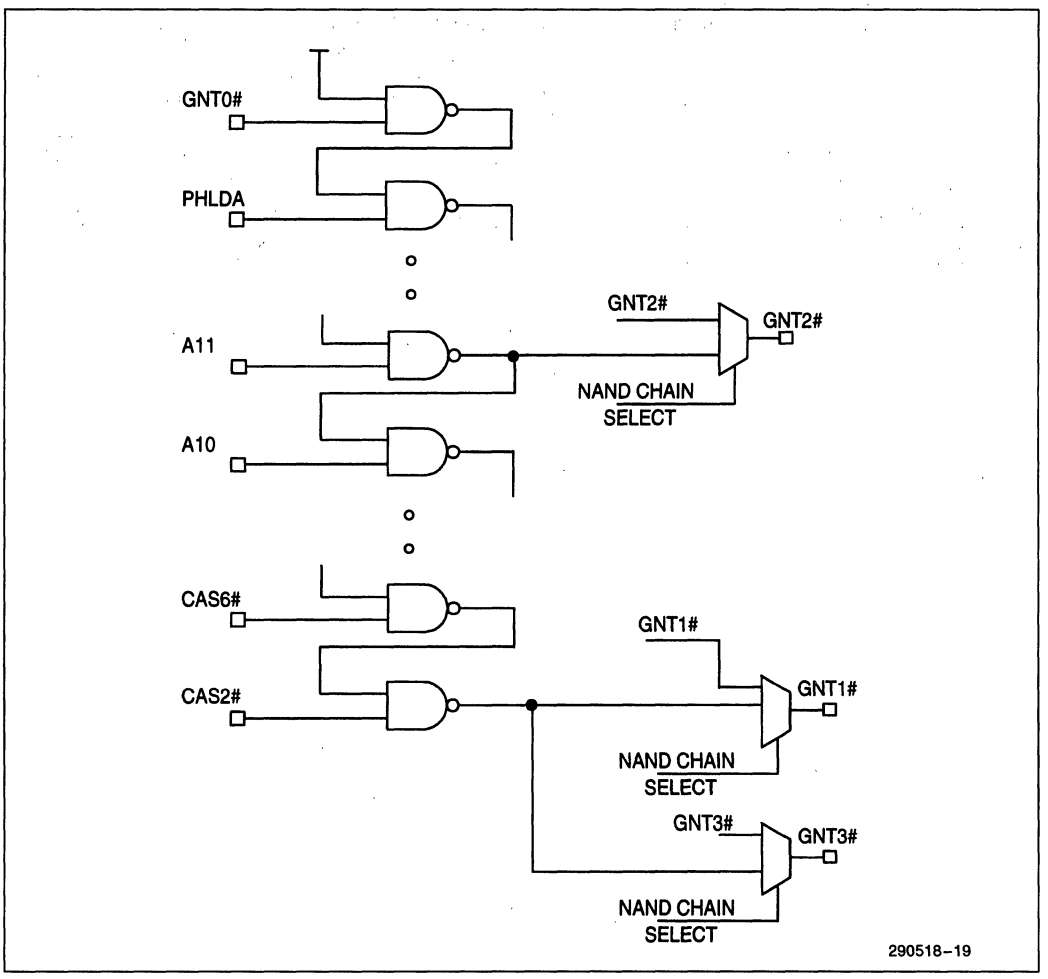


Figure 17. 82437FX NAND Tree Circuitry

NAND Tree Timing Requirements

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

7.0 82438FX TDP TESTABILITY

7.1 Test Mode Description

The test modes are decoded from HOE#, MOE#, POE#, and MSTB#. HCLK must be active for at least one clock to sample the above signals. Once these signals are sampled then HCLK must be asserted to 0 for the duration of the NAND tree test. The test modes are defined as follows.

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

NAND tree mode is exited by starting HCLK with HOE#, MOE#, and POE# **not** equal to "111".

7.2 NAND Tree Mode

Tri-states all outputs and bidirectional buffers except for MD0 which is the output of the NAND tree. The NAND tree follows the pins sequentially around the chip skipping only HCLK and MD0. The first input of the NAND chain is HD5, and the NAND chain is routed counter-clockwise around the chip (e.g., HD5, HD6 . . .). The only valid output during NAND tree mode is MD0.

To perform a NAND tree test, all pins included in the NAND tree should be driven to 1, with the exception of HCLK and MDO. Beginning with HD5 and working counter-clockwise around the chip, each pin can be toggled with a resulting toggle observed on MD0. After changing an input pin to 0, keep it at 0 for the remainder of the NAND tree test.

Table 17. NAND Tree Cell Order for the 82438FX

Pin #	Pin Name	Notes
77	MOE#	Must be 1 to enter NAND tree mode. This pin is included in the NAND tree mode input pin sequence.
78	POE#	Must be 1 to enter NAND tree mode. This pin is included in the NAND tree mode input pin sequence.
81	HOE#	Must be 1 to enter NAND tree mode. This pin is included in the NAND tree mode input pin sequence.
83	MSTB#	Must be 1 to enter NAND tree mode. This pin is included in the NAND tree mode input pin sequence.
30	HCLK	HCLK must clock at least once to sample MOE#, POE#, HOE#, and MSTB# to select NAND tree mode. Then to enter NAND tree mode HCLK must remain 0. To exit NAND tree mode HCLK must be started.
41	MD0	Output of the NAND tree chain.
3	HD5	First signal in the NAND tree chain.
4	HD6	
5	HD7	
6	HD8	
7	HD9	
8	HD10	
9	HD11	
10	HD12	
11	HD13	
12	HD14	

Pin #	Pin Name	Notes
13	HD15	
14	HD16	
15	HD17	
16	HD18	
17	HD19	
18	HD20	
20	HD21	
21	HD22	
22	HD23	
23	HD24	
25	HD25	
26	HD26	
27	HD27	
32	HD28	
33	HD29	
34	HD30	
35	HD31	
38	MD24	
39	MD8	
40	MD16	
42	MD25	
43	MD9	
44	MD17	
45	MD1	
46	MD26	
47	MD10	
50	MD18	
51	MD2	
54	MD27	
55	MD11	

Table 17. NAND Tree Cell Order for the 82438FX (Continued)

Pin #	Pin Name	Notes
56	MD19	
57	MD3	
58	MD28	
59	MD12	
60	MD20	
61	MD4	
62	MD21	
63	MD5	
66	MD29	
67	MD13	
68	MD22	
69	MD6	
70	MD30	
71	MD14	
72	MD23	
73	MD7	
74	MD31	
75	MD15	
77	MOE#	Must be 1 to enter NAND tree mode. This pin is included in the NAND tree mode input pin sequence.
78	POE#	Must be 1 to enter NAND tree mode. This pin is included in the NAND tree mode input pin sequence.
81	HOE#	Must be 1 to enter NAND tree mode. This pin is included in the NAND tree mode input pin sequence.
82	MADV#	
83	MSTB#	Must be 1 to enter NAND tree mode. This pin is included in the NAND tree mode input pin sequence.
84	PCMD1	
85	PCMD0	
88	PLINK7	
89	PLINK6	
90	PLINK5	
91	PLINK4	
92	PLINK3	
93	PLINK2	
94	PLINK1	
95	PLINK0	
96	HD0	
97	HD1	
98	HD2	
99	HD3	
100	HD4	Final signal in the NAND tree chain.

Figure 18 is a schematic of the NAND tree circuitry.

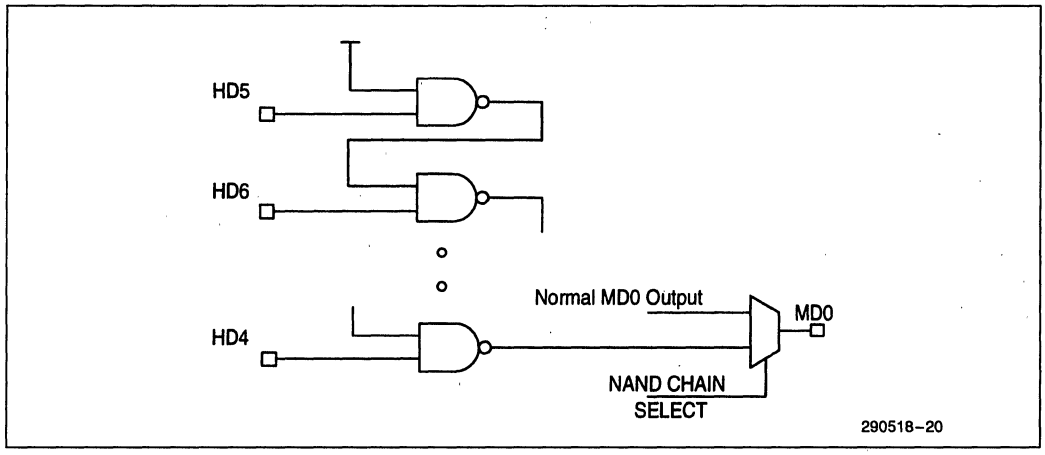


Figure 18. 82438FX NAND Tree Circuitry

NAND Tree Timing Requirements

Allow 500 ns for the input signals to propagate to the NAND tree outputs (input-to-output propagation delay specification).

82371FB PCI ISA IDE XCELERATOR (PIIX)

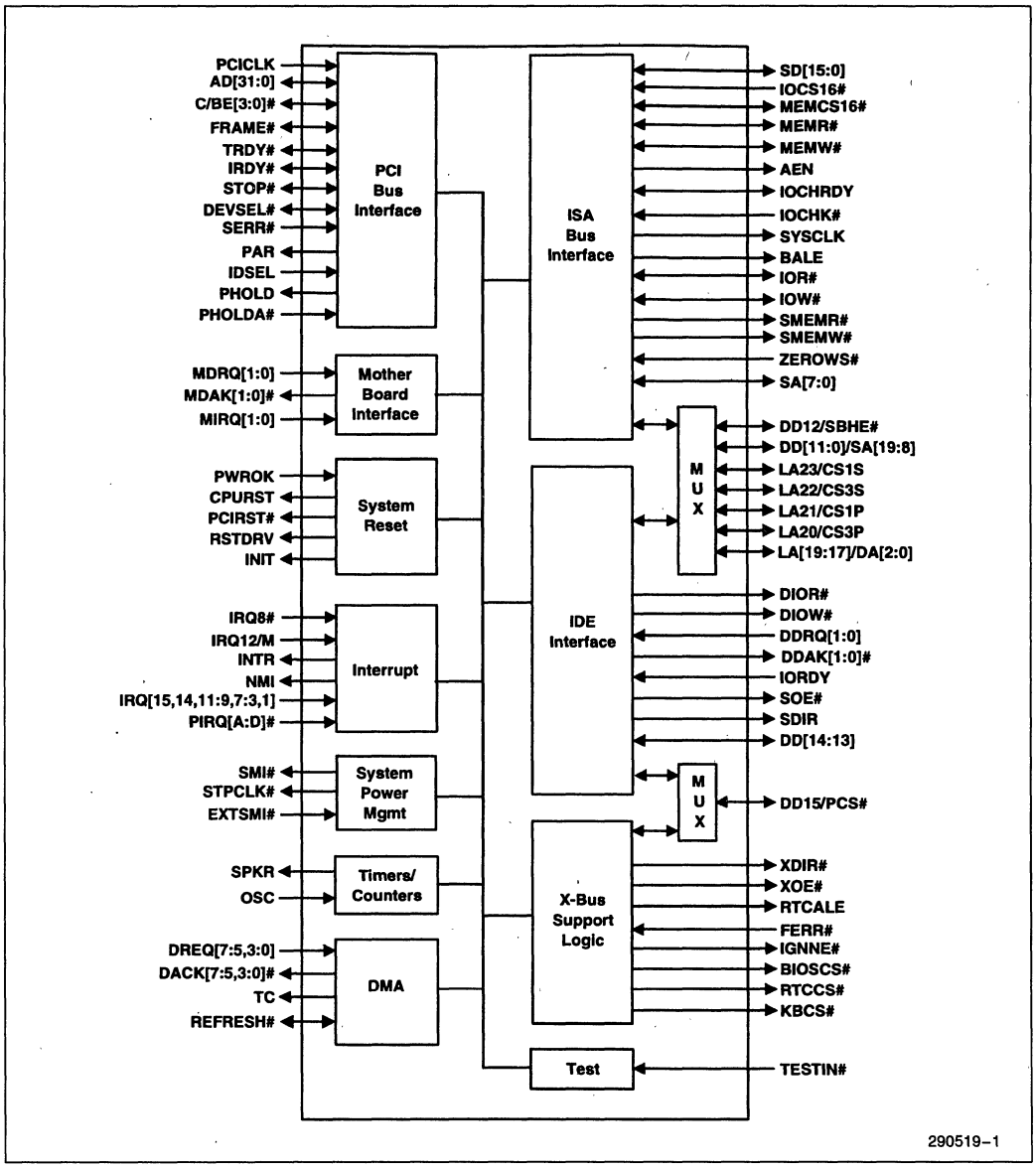
- Provides a Bridge Between the PCI Bus and ISA Bus
- PCI and ISA Master/Slave Interface
  - PCI from 25 to 33 MHz
  - ISA from 7.5 to 8.33 MHz
  - Five ISA Slots
- Fast IDE Interface
  - Supports PIO and Bus Master IDE
  - Supports Up to Mode 4 Timings
  - Transfer Rates to 22 Mbytes/Sec
  - 8 x 32-Bit Buffer for Bus Master IDE PCI Burst Transfers
- Plug-and-Play Port for Motherboard Devices
  - 2 Steerable DMA Channels
  - Fast DMA with 4-Byte Buffer
  - Up to 2 Steerable Interrupt Lines
  - 1 Programmable Chip Select
- Steerable PCI Interrupts for PCI Device Plug-and-Play
- Functionality of One 82C54 Timer
  - System Timer
  - Refresh Request
  - Speaker Tone Output
- Functionality of Two 82C59 Interrupt Controllers
  - 14 Interrupts Supported
  - Independently Programmable for Edge/Level Sensitivity
- Enhanced DMA Functions
  - Two 8237 DMA Controllers
  - Fast Type F DMA
  - Compatible DMA Transfers
  - Seven Independently Programmable Channels
- X-Bus Peripheral Support
  - Chip Select Decode
  - Controls Lower X-Bus Data Byte Transceiver
- System Power Management (Intel SMM Support)
  - Programmable System Management Interrupt (SMI)Hardware Events, Software Events, EXTSMI#
  - Programmable CPU Clock Control (STPCLK#)
  - Fast-On/Off Mode
- Non-Maskable Interrupts (NMI)
  - PCI System Error Reporting
- NAND Tree for Board-Level ATE Testing
- 208-Pin QFP

The 82371FB PCI ISA IDE Xcelerator (PIIX) is a multi-function PCI device implementing a PCI-to-ISA bridge. In addition, the PIIX has an IDE interface with both programmed I/O (PIO) and Bus Master functions. As a PCI-to-ISA bridge, the PIIX integrates many common I/O functions found in ISA-based PC systems—a seven channel DMA controller, two 82C59 interrupt controllers, an 8254 timer/counter, Intel SMM power management support, and control logic for NMI generation. In addition to compatible transfers, each DMA channel supports type F transfers. Chip select decoding is provided for BIOS, real time clock, and keyboard controller. Edge/level interrupts and interrupt steering are supported for PCI plug and play compatibility.

For motherboard plug-and-play compatibility, the PIIX also provides two steerable DMA channels (including type F transfers), up to two steerable interrupt lines, and a programmable chip select. The interrupt lines can be routed to any of the available ISA interrupts.

The PIIXs fast IDE interface supports two IDE connectors for up to four IDE devices providing an interface for IDE hard disks and CD ROMs.

November 1995
Order Number: 290519-001



82371FB PIIX Simplified Block Diagram

82371FB PCI ISA IDE XCELERATOR (PIIX)

CONTENTS

CONTENTS

CONTENTS

CONTENTS

1.0 SIGNAL DESCRIPTION

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

The "#" symbol at the end of a signal name indicates that the active or asserted state occurs when the signal is at a low voltage level. When "#" is not present after the signal name, the signal is asserted when at the high voltage level.

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

Certain signal pins provide two separate functions. At the system level, these pins drive other signals with different functions through external buffers or transceivers. These pins have two different signal names depending on the function. These signal names have been noted in the signal description tables, with the signal whose function is being described in **bold** font. (For example, LA23/**CS1S** is in the section describing CS1S and **LA23**/CS1S is in the section describing LA23).

The following notations are used to describe the signal type:

I *Input* is a standard input-only signal.

O *Totem Pole Output* is a standard active driver.

I/O *Input/Output* is a bi-directional, tri-state signal.

od *Open Drain* allows multiple devices to share as a wire-OR.

t/s *Tri-state* is a bi-directional, tri-state input/output pin.

s/t/s *Sustained Tri-state* is an active low tri-state signal owned and driven by one agent at a time. The agent that drives a s/t/s pin low must drive it high for at least one clock before letting it float. A new agent can not start driving a s/t/s signal any sooner than one clock after the previous owner tri-states it. An external pull-up is required to sustain the inactive state until another agent drives it and must be provided by the central resource.

1.1 PCI Interface Signals

Signal Name	Type	Description
PCICLK	I	**PCI CLOCK:** PCICLK provides timing for all transactions on the PCI Bus. All other PCI signals are sampled on the rising edge of PCICLK, and all timing parameters are defined with respect to this edge. PCI frequencies of 25–33 MHz are supported.
AD[31:0]	I/O	**PCI ADDRESS/DATA:** The standard PCI address and data lines. The address is driven with FRAME# assertion and data is driven or received in following clocks.
C/BE[3:0]#	I/O	**BUS COMMAND AND BYTE ENABLES:** The command is driven with FRAME# assertion. Byte enables corresponding to supplied or requested data is driven on following clocks.
FRAME#	I/O (s/t/s)	**FRAME:** Assertion indicates the address phase of a PCI transfer. Negation indicates that one or more data transfer is desired by the cycle initiator.
TRDY#	I/O (s/t/s)	**TARGET READY:** Asserted when the target is ready for a data transfer.
IRDY#	I/O (s/t/s)	**INITIATOR READY:** Asserted when the initiator is ready for a data transfer.
STOP#	I/O (s/t/s)	**STOP:** Asserted by the target to request the master to stop the current transaction.
IDSEL	I	**INITIALIZATION DEVICE SELECT:** IDSEL is used as a chip select during configuration read and write transactions.

Signal Name	Type	Description
DEVSEL#	I/O (s/t/s)	**DEVICE SELECT:** The PIIX asserts DEVSEL# to claim a PCI transaction through positive or subtractive decoding.
PAR	O	**CALCULATED PARITY SIGNAL:** PAR is "even" parity and is calculated on 36 bits—AD[31:0] plus C/BE[3:0]#.
SERR#	I	**SYSTEM ERROR:** SERR# can be pulsed active by any PCI device that detects a system error condition. Upon sampling SERR# active, the PIIX can be programmed to generate a non-maskable interrupt (NMI) to the CPU.
PHOLD#	O	**PCI HOLD:** The PIIX asserts this signal to request the PCI Bus.
PHLDA#	I	**PCI HOLD ACKNOWLEDGE:** The TSC asserts this signal to grant the PCI Bus to the PIIX.

1.2 Motherboard I/O Device Interface Signals

Signal Name	Type	Description
MDRQ[1:0]	I	**MOTHERBOARD DEVICE DMA REQUEST:** The signals can be connected internally to any of DREQ[3:0,7:5]. Each pair of request/acknowledge signals is controlled by a separate register. Each signal can be configured as steerable interrupts for motherboard devices.
MDAK[1:0]#	O	**MOTHERBOARD DEVICE DMA ACKNOWLEDGE:** These signals can be connected internally to any of DACK[3:0,7:5]. Each pair of request/acknowledge signals is controlled by a separate register. Each signal can be configured as steerable interrupts for motherboard devices.
MIRQ[1:0]	I	**MOTHERBOARD DEVICE INTERRUPT REQUEST:** The MIRQ signals can be internally connected to interrupts IRQ[15,14,12:9,7:3]. Each MIRQx line has a separate Route Control Register. If MIRQx and PIRQx are steered to the same ISA interrupt, the device connected to the MIRQx should produce active high, level-sensitive interrupts. If the Bus Master mode of the IDE interface is used and the secondary IDE channel is used, the interrupt for that channel must be connected to MIRQ0. If an MIRQ line is steered to a given IRQ input to the internal 8259, the corresponding ISA IRQ is masked, unless the Route Control Register is programmed to allow the interrupts to be shared. The should only be done if the device connected to the MIRQ line and the device connected to the ISA IRQ line both produce active high level interrupts.

1.3 IDE Interface Signals

Signal Name	Type	Description
DD[15:0]/ PCS#, SBHE#, SA[19:8]	**I/O** O I/O I/O	**DISK DATA:** These signals directly drive the corresponding signals on up to two IDE connectors (primary and secondary). In addition, these signals are buffered (using 2xALS245's on the motherboard) to produce the SA[19:8] and PCS# and SBHE# signals (see separate descriptions).
DIOR#	O	**DISK I/O READ:** This signal directly drives the corresponding signal on up to two IDE connectors (primary and secondary).
DIOW#	O	**DISK I/O WRITE:** This signal directly drives the corresponding signal on up to two IDE connectors (primary and secondary).
DDRQ[1:0]	I	**DISK DMA REQUEST:** These input signals are directly driven from the DRQ signals on the primary (DDRQ0) and secondary (DDRQ1) IDE connectors. They are used in conjunction with any ISA-Compatible DMA channel.
DDAK[1:0]#	O	**DISK DMA ACKNOWLEDGE:** These signals directly drive the DAK# signals on the primary (DDAK0#) and secondary (DDAK1#) IDE connectors. These signals are used in conjunction with the PCI Bus Master IDE function and are not associated with any ISA-Compatible DMA channel.
IORDY	I	**IO CHANNEL READY:** This input signal is directly driven by the corresponding signal on up to two IDE connectors (primary and secondary).
SOE#	O	**SYSTEM ADDRESS TRANSCEIVER OUTPUT ENABLE:** This signal controls the output enables of the 245 transceivers that interface the DD[15:0] signals to the SA[19:8], SBHE#, and PCS# signals.
SDIR	O	**SYSTEM ADDRESS TRANSCEIVER DIRECTION:** This signal controls the direction of the '245 transceivers that interface the DD[15:0] signals to the SA[19:8], SBHE#, and PCS# signals. Default condition is high (transmit). When an ISA Bus Master is granted use of the bus, the transceivers are turned around to drive the ISA address [19:8] on DD[15:3]. The address can then be latched by the PIIX. In this case, the SDIR signal is low (receive). The SOE# and SDIR signals taken together as a group can assume one of three states: **SOE# SDIR State** 0 1 PCI to ISA transaction 1 1 PCI to IDE 0 0 ISA Bus Master

Signals Buffered from LA[23:17]

These signals are buffered from the LA[23:17] lines by an ALS244 tri-state buffer. The output enable of this buffer is tied asserted. These signals are set up with respect to the IDE command strobes (DIOR# and IOW#) and are valid throughout I/O transactions targeting the ATA Register clock(s).

Signal Name	Type	Description
LA23/ **CS1S**	I/O	**CHIP SELECT:** CS1S is for the ATA command register block and corresponds to the inverted CS1FX# on the secondary IDE connector. CS1S is inverted externally (see PCI Local Bus IDE section).
LA22/ **CS3S**	I/O	**CHIP SELECT:** CS3S is for the ATA control register block and corresponds to the inverted CS3FX# on the secondary IDE connector. CS35 is inverted externally (see PCI Local Bus IDE section).
LA21/ **CS1P**	I/O	**CHIP SELECT:** CS1P is for the ATA command register block and corresponds to the inverted CS3FX# on the primary IDE connector. CS1P is inverted externally (see PCI Local Bus IDE section).
LA20/ **CS3P**	I/O	**CHIP SELECT:** CS3P is for the ATA control register block and corresponds to the inverted CS3FX# on the primary IDE connector. CS3P is inverted externally (see PCI Local Bus IDE section).
LA[19:17]/ **DA[2:0]**	I/O	**DISK ADDRESS:** DA[2:0] are used to indicate which byte in either the ATA command block or control block is being addressed.

1.4 ISA Interface Signals

Signal Name	Type	Description
BALE	O	**BUS ADDRESS LATCH ENABLE:** BALE is an active high signal asserted by the PIIX to indicate that the address (SA[19:0], LA[23:17]) and SBHE# signal lines are valid.
AEN	O	**ADDRESS ENABLE:** AEN is asserted during DMA cycles to prevent I/O slaves from misinterpreting DMA cycles as valid I/O cycles. This signal is also driven high during PIIX initiated refresh cycles. When TC is sampled low on the assertion of PWORK (external DMA mode), the PIIX tri-states this signal.
SYSCLK	O	**ISA SYSTEM CLOCK:** SYSCLK is the reference clock for the ISA Bus and drives the bus directly. SYSCLK is generated by dividing PCICLK by 3 or 4. The SYSCLK frequencies supported are 7.5 MHz and 8.33 MHz. SYSCLK is a divided down version of PCICLK. **Hardware Strapping Option** SYSCLK is tri-stated when PWROK is negated. The value of SYSCLK is sampled on the assertion of PWROK: If sampled high, the ISA clock divisor is 3 (for 25 MHz PCI). If sampled low, the divisor is 4 (for 30 and 33 MHz PCI).
IOCHRDY	I/O	**I/O CHANNEL READY:** Resources on the ISA Bus negate IOCHRDY to indicate that additional time (wait states) is required to complete the cycle. This signal is normally high on the ISA Bus. IOCHRDY is an input when the PIIX owns the ISA Bus and the CPU or a PCI agent is accessing an ISA slave or during DMA transfers. IOCHRDY is output when an external ISA Bus Master owns the ISA Bus and is accessing DRAM or a PIIX register.

Signal Name	Type	Description
IOCS16#	I	**16-BIT I/O CHIP SELECT:** This signal is driven by I/O devices on the ISA Bus to indicate that they support 16-bit I/O bus cycles.
IOCHK#	I	**I/O CHANNEL CHECK:** IOCHK# can be driven by any resource on the ISA Bus. When asserted, it indicates that a parity or an uncorrectable error has occurred for a device or memory on the ISA Bus. If enabled, a NMI is generated to the CPU.
IOR#	I/O	**I/O READ:** IOR# is the command to an ISA I/O slave device that the slave may drive data on to the ISA data bus (SD[15:0]).
IOW#	I/O	**I/O WRITE:** IOW# is the command to an ISA I/O slave device that the slave may latch data from the ISA data bus (SD[15:0]).
LA[23:17]/ CS1S# CS3S# CS1P# CS3P# DA[2:0]	**I/O/** O O O O O	**UNLATCHED ADDRESS:** The LA[23:17] address lines are bi-directional. These address lines allow accesses to physical memory on the ISA Bus up to 16 Mbytes. The LA[23:17] are also used to drive the IDE interface chip selects and address lines via an external ALS244 buffer. See the IDE Interface signal descriptions.
SA[7:0], **SA[19:8]/** DD[11:0]	I/O, **I/O,** I/O,	**SYSTEM ADDRESS BUS:** These bi-directional address lines define the selection with the granularity of one byte within the one Mbyte section of memory defined by the LA[23:17] address lines. The address lines SA[19:17] that are coincident with LA[19:17] are defined to have the same values as LA[19:17] for all memory cycles. For I/O accesses, only SA[15:0] are used.
SBHE#/ DD12	**I/O** I/O	**SYSTEM BYTE HIGH ENABLE:** SBHE# indicates, when asserted, that a byte is being transferred on the upper byte (SD[15:8]) of the data bus. SBHE# is negated during refresh cycles.
MEMCS16#	od	**MEMORY CHIP SELECT 16:** MEMCS16# is a decode of LA[23:17] without any qualification of the command signal lines. ISA slaves that are 16-bit memory devices drive this signal low. The PIIX drives this signal low during ISA master to DRAM Cycles.
MEMR#	I/O	**MEMORY READ:** MEMR# is the command to a memory slave that it may drive data onto the ISA data bus. This signal is also driven by the PIIX during refresh cycles.
MEMW#	I/O	**MEMORY WRITE:** MEMW# is the command to a memory slave that it may latch data from the ISA data bus.
SMEMR#	O	**STANDARD MEMORY READ:** The PIIX asserts SMEMR# to request an ISA memory slave to drive data onto the data lines. If the access is below 1 Mbyte (00000000–000FFFFFh) during DMA compatible, PIIX master, or ISA master cycles, the PIIX asserts SMEMR#. SMEMR# is a delayed version of MEMR#.
SMEMW#	O	**STANDARD MEMORY WRITE:** The PIIX asserts SMEMW# to request an ISA memory slave to accept data from the data lines. If the access is below 1 Mbyte (00000000–000FFFFFh) during DMA compatible, PIIX master, or ISA master cycles, the PIIX asserts SMEMW#. SMEMW# is a delayed version of MEMW#.

Signal Name	Type	Description
ZEROWS#	I	**ZERO WAIT-STATES:** An ISA slave asserts ZEROWS# after its address and command signals have been decoded to indicate that the current cycle can be shortened. A 16-bit ISA memory cycle can be reduced to two SYSCLKs. An 8-bit memory or I/O cycle can be reduced to three SYSCLKs. ZEROWS# has no effect during 16-bit I/O cycles.
SD[15:0]	I/O	**SYSTEM DATA:** SD[15:0] provide the 16-bit data path for devices residing on the ISA Bus. SD[15:8] correspond to the high order byte and SD[7:0] correspond to the low order byte. SD[15:0] are undefined during refresh.

1.5 DMA Signals

Signal Name	Type	Description
DREQ [7:5,3:0]	I	**DMA REQUEST:** The DREQ lines are used to request DMA service from the PIIXs DMA controller or for a 16-bit master to gain control of the ISA expansion bus. The active level (high or low) is programmed via the DMA Command Register. The request must remain active until the appropriate DACKx# signal is asserted.
DACK [7:5,3:0]#	O	**DMA ACKNOWLEDGE:** The DACK output lines indicate that a request for DMA service has been granted by the PIIX or that a 16-bit master has been granted the bus. The active level (high or low) is programmed via the DMA Command Register. These lines should be used to decode the DMA slave device with the IOR# or IOW# line to indicate selection. If used to signal acceptance of a Bus Master request, this signal indicates when it is legal to assert MASTER#. When TC is sampled low on the assertion of PWORK (external DMA mode), the PIIX tri-states these signals.
TC	O	**TERMINAL COUNT:** The PIIX asserts TC to DMA slaves as a terminal count indicator. When all the DMA channels are not in use, TC is negated (low). **Hardware Strapping Option** This strapping option selects between the internal ISA DMA mode and external DMA mode. When TC is sampled high on the assertion of PWROK (ISA DMA mode), the PIIX drives the AEN, TC, and DACK# [7:5,3:0] normally. When TC is sampled low on the assertion of PWROK (external DMA mode), the PIIX tri-states the AEN, TC, and DACK[7:5,3:0]# signals, and also forwards PCI masters I/O accesses to location 0000h to ISA. TC has an internal pull-up resistor. For normal operation, this pin is pulled high by the internal pull-up.
REFRESH#	I/O	**REFRESH:** As an output, REFRESH# indicates when a refresh cycle is in progress. It should be used to enable the SA[15:0] address to the row address inputs of all banks of dynamic memory on the ISA Bus. Thus, when MEMR# is asserted, the entire expansion bus dynamic memory is refreshed. Memory slaves must not drive any data onto the bus during refresh. As an output, this signal is driven directly onto the ISA Bus. This signal is an output only when the PIIX DMA refresh controller is a master on the bus responding to an internally generated request for refresh. As an input, REFRESH# is driven by 16-bit ISA Bus Masters to initiate refresh cycles.

1.6 Timer/Counter Signals

Signal Name	Type	Description
SPKR	O	**SPEAKER DRIVE:** The SPKR signal is the output of counter 2.
OSC	I	**OSCILLATOR:** OSC is the 14.31818 MHz ISA clock signal. It is used by the internal 8254 Timer.

1.7 Interrupt Controller Signals

Signal Name	Type	Description
IRQ[15,14, 11:9,7:3,1]	I	**INTERRUPT REQUEST:** The IRQ signals provide both system board components and ISA Bus I/O devices with a mechanism for asynchronously interrupting the CPU. The assertion mode of these inputs depends on the programming of the two ELCR Registers. The IRQ14 signal (pin 83) must be used by the Bus Master IDE interface function to signal interrupts on the primary IDE channel.
IRQ8#	I	**INTERRUPT REQUEST EIGHT SIGNAL:** IRQ8# is always an active low edge triggered interrupt input (i.e., this interrupt can not be modified by software). Upon PCIRST#, IRQ8# is placed in active low edge sensitive mode.
IRQ12/M	I	**INTERRUPT REQUEST/MOUSE INTERRUPT:** In addition to providing the standard interrupt function (see IRQ[15,14,11:9,7:3,1] signal description), this pin can be programmed (via X-Bus Chip Select Register) to provide a mouse interrupt function.
PIRQ[3:0]#	I	**PROGRAMMABLE INTERRUPT REQUEST:** The PIRQx# signals can be shared with interrupts IRQ[15,14,12:9,7:3] as described in the Interrupt Steering section. Each PIRQx# line has a separate Route Control Register. These signals require external pull-up resisters.
INTR	od	**CPU INTERRUPT:** INTR is driven by the PIIX to signal the CPU that an interrupt request is pending and needs to be serviced. The interrupt controller must be programmed following PCIRST# to ensure that INTR is at a known state.
NMI	od	**NON-MASKABLE INTERRUPT:** NMI is used to force a non-maskable interrupt to the CPU. The PIIX generates an NMI when either SERR# or IOCHK# is asserted, depending on how the NMI Status and Control Register is programmed.

1.8 System Power Management (SMM) Signals

Signal Name	Type	Description
SMI#	od	**SYSTEM MANAGEMENT INTERRUPT:** SMI# is an active low synchronous output that is asserted by the PIIX in response to one of many enabled hardware or software events.
STPCLK#	od	**STOP CLOCK:** STPCLK# is an active low synchronous output that is asserted by the PIIX in response to one of many hardware or software events. STPCLK# connects directly to the CPU and is synchronous to PCICLK.
EXTSMI#	I	**EXTERNAL SYSTEM MANAGEMENT INTERRUPT:** EXTSMI# is a falling edge triggered input to the PIIX indicating that an external device is requesting the system to enter SMM mode. An external pullup should be placed on this signal if it is not used or it is not guaranteed to be always driven.

1.9 X-Bus Signals

Signal Name	Type	Description
XDIR#	O	**X-BUS DIRECTION:** XDIR# is tied directly to the direction control of a 74F245 that buffers the X-Bus data (XD[7:0]). XDIR# is asserted for all I/O read cycles, regardless if the accesses are to a PIIX supported device. XDIR# is only asserted for memory cycles if BIOS space has been decoded. For PCI Master and ISA master-initiated read cycles, XDIR# is asserted from the falling edge of either IOR# or MEMR# (from MEMR# only if BIOS space has been decoded), depending on the cycle type. When the rising edge of IOR# or MEMR# occurs, the PIIX negates XDIR#. For DMA read cycles from the X-Bus, XDIR# is asserted from DACKx# falling and negated from DACKx# rising. At all other times, XDIR# is negated.
XOE#	O	**X-BUS OUTPUT ENABLE:** XOE# is tied directly to the output enable of a 74F245 that buffers the X-Bus data (XD[7:0]) from the system data bus (SD[7:0]). XOE# is asserted when a PIIX supported X-Bus device is decoded, and the devices decode is enabled in the X-Bus Chip Select Enable Register (XBCS Register). XOE# is asserted from the falling edge of the ISA commands (IOR#, IOW#, MEMR#, or MEMW#) for PCI Master and ISA master-initiated cycles. XOE# is negated from the rising edge of the ISA command signals for CPU and PCI Master-initiated cycles and the SA[16:0] and LA[23:17] address for ISA master-initiated cycles. XOE# is not generated during any access to an X-Bus peripheral in which it's decode space has been disabled.
DD15/ **PCS#**	O	**PROGRAMMABLE CHIP SELECT:** PCS# is asserted for ISA I/O cycles that are generated by PCI masters and subtractively decoded to ISA, if the access hits the address range programmed into the PCSC Register. The X-Bus buffer signals are enabled when the chip select is asserted (i.e., it is assumed that the peripheral that is selected via this pin resides on the X-Bus).
BIOSCS#	O	**BIOS CHIP SELECT:** BIOSCS# is asserted during read or write accesses to BIOS. BIOSCS# is driven combinatorially from the ISA addresses SA[16:0] and LA [23:17], except during DMA. During DMA cycles, BIOSCS# is not generated.
KBCS#	O	**KEYBOARD CONTROLLER CHIP SELECT:** KBCS# is asserted during I/O read or write accesses to KBC locations 60h and 64h. This signal is driven combinatorially from the ISA addresses SA[16:0] and LA [23:17]. For DMA cycles, KBCS# is never asserted.
RTCCS#	O	**REAL TIME CLOCK CHIP SELECT:** RTCCS# is asserted during read or write accesses to RTC location 71h. RTCCS# can be tied to a pair of external OR gates to generate the real time clock read and write command signals.
RTCALE	O	**REAL TIME CLOCK ADDRESS LATCH:** RTCALE is used to latch the appropriate memory address into the RTC. A write to port 70h with the appropriate RTC memory address that will be written to or read from, causes RTCALE to be asserted. RTCALE is asserted based on IOW# falling and remains asserted for two SYSCLKs.
FERR#	od	**NUMERIC COPROCESSOR ERROR:** This signal is tied to the coprocessor error signal on the CPU. IGNNE# is only used if the PIIX coprocessor error reporting function is enabled in the XBCSA Register. If FERR# is asserted, the PIIX generates an internal IRQ13 to it's interrupt controller unit. The PIIX then asserts the INTR output to the CPU. FERR# is also used to gate the IGNNE# signal to ensure that IGNNE# is not asserted to the CPU unless FERR# is active. FERR# has a weak internal pull-up used to ensure a high level when the coprocessor error function is disabled.

Signal Name	Type	Description
IGNNE#	od	**IGNORE ERROR:** This signal is connected to the ignore error pin on the CPU. IGNNE# is only used if the PIIX coprocessor error reporting function is enabled in the XBCSA Register. If FERR# is asserted, indicating a coprocessor error, a write to the Coprocessor Error Register (F0h) causes the IGNNE# to be asserted. IGNNE# remains asserted until FERR# is negated. If FERR# is not asserted when the Coprocessor Error Register is written, the IGNNE# signal is not asserted.

1.10 System Reset Signals

Signal Name	Type	Description
PWROK	I	**POWER OK:** When asserted, PWROK is an indication to the PIIX that power and PCICLK have been stable for at least 1 ms. PWROK can be driven asynchronously. When PWROK transitions from low to high, the PIIX asserts CPURST, PCIRST# and RSTDRV.
CPURST	od	**CPU RESET:** The PIIX asserts CPURST to reset the CPU. The PIIX asserts CPURST during power-up and when a hard reset sequence is initiated through the RC Register. CPURST is driven synchronously to the rising edge of PCICLK. If a hard reset is initiated through the RC Register, the PIIX resets it's internal registers to the default state.
PCIRST#	O	**PCI RESET:** The PIIX asserts PCIRST# to reset devices that reside on the PCI Bus. The PIIX asserts PCIRST# during power-up and when a hard reset sequence is initiated through the RC Register. PCIRST# is driven inactive a minimum of 1 ms after PWROK is driven active. PCIRST# is driven active for a minimum of 1 ms when initiated through the RC Register. PCIRST# is driven asynchronously relative to PCICLK.
INIT	O	**INITIALIZATION:** The PIIX asserts INIT if it detects a shut down special cycle on the PCI Bus or if a soft reset is initiated via the RC Register.
RSTDRV	O	**RESET DRIVE:** The PIIX asserts this signal during a hard reset and during power-up to reset ISA Bus devices. RSTDRV is also asserted for a minimum of 1 ms if a hard reset has been programmed in the RC Register.

1.11 Test Signals

Signal Name	Type	Description
TESTIN#	I	**TEST INPUT:** The Test signal is used to tri-state all of the PIIX outputs. This input contains an internal pull-up resistor.

1.12 Signal State During Reset

Table 1 shows the state of all PIIX output and bi-directional signals during a hard reset. A hard reset is initiated when PWROK is asserted or by programming a hard reset through the RC Register.

Table 1. Output and I/O Signal States During Hard Reset

Signal	State
AD[31:0]	Low
C/BE[3:0]#	Low
FRAME#	Tri-state
TRDY#	Tri-state
IRDY#	Tri-state
STOP#	Tri-state
DEVSEL#	Tri-state
PAR	Input
PHOLD#	High
MDAK[1:0]#	High
DD[15:0]/ PCS#,SBHE#, SA[19:8]	Tri-state
SA[7:0]	Undefined
DIOR#	High
DIOW#	High
DDAK[1:0]#	High*
SOE#	High
SDIR	High
LA23/CS1S	Undefined

Signal	State
LA22/CS3S	Undefined
LA21/CS1P	Undefined
LA20/CS3P	Undefined
LA[19:17]/ DA[2:0]	Undefined
BALE	Low
AEN	Depends on strapping option
SYSCLK	Strapping option
IOCHRDY	Tri-state
IOR#	High
IOW#	High
MEMCS16#	Open drain
MEMR#	Tri-state
MEMW#	Tri-state
SMEMR#	High
SMEMW#	High
SD[15:0]	Tri-state
DACK[7:5, 3:0]#	Depends on strapping option

Signal	State
TC	High*
REFRESH#	Tri-state
SPKR	Low
INTR	Open drain
NMI	Open drain
SMI#	Open drain
STPCLK#	Open drain
XDIR#	High
XOE#	High
BIOSCS#	High
KBCS#	High
RTCCS#	High
RTCALE	Low
FERR#	Open drain
IGNNE#	Open drain
CPURST	Open drain
PCIRST#	Low
INIT	Open drain
RSTDRV	High

* DDAK[0] and TC are pulled high with an internal pull-up.

2.0 REGISTER DESCRIPTION

The PIIX internal registers are organized into five groups—PCI Configuration Registers (function 0), PCI Configuration Registers (function 1), ISA-Compatible Registers, PCI Bus Master IDE Registers, and Power Management Registers. These registers are discussed in this section.

Some of the PIIX registers contain reserved bits. Software must deal correctly with fields that are reserved. On reads, software must use appropriate masks to extract the defined bits and not rely on reserved bits being any particular value. On writes, software must ensure that the values of reserved bit positions are preserved. That is, the values of reserved bit positions must first be read, merged with the new values for other bit positions and then written back.

During a hard reset, the PIIX sets its internal registers to predetermined **default** states. The default values are indicated in the individual register descriptions.

The following notation is used to describe register access attributes:

RO Read Only. If a register is read only, writes have no effect.

WO Write Only. If a register is write only, reads have no effect.

R/W Read/Write. A register with this attribute can be read and written. Note that individual bits in some read/write registers may be read only.

R/WC Read/Write Clear. A register bit with this attribute can be read and written. However, a write of 1 clears (sets to 0) the corresponding bit and a write of 0 has no effect.

2.1 Register Access

Table 2, Table 3, and Table 4, show the I/O assignments for the PCI Configuration Registers (function 0), ISA Compatible Registers, and PCI Configuration Registers (function 1). The CPU and PCI masters have access to all PIIX internal registers. In addition, ISA masters have access to some of the ISA-Compatible Registers (see Table 4). Table 5 show the I/O assignments for the Bus Master IDE Interface Registers.

PCI Configuration Registers (Functions 0 and 1)

The PIIX is a multi-function device on the PCI Bus implementing two functions—PCI-to-ISA Bridge (function 0) and IDE Interface (function 1). These functions can be independently configured with two sets of PCI Configuration Registers in compliance with the PCI Local Bus Specification, Revision 2.0. The two sets of configuration registers are accessed by the CPU through a mechanism defined for multi-functional PCI devices. The PIIX does not assert DEVSEL# for PCI Configuration cycles that target functions 2 through 7.

ISA Compatible Registers

The ISA-Compatible Registers (e.g., DMA Registers, timer/counter registers, X-Bus Registers, and NMI Registers) are accessed through normal I/O space. Except for the DMA Registers, the PIIX positively decodes accesses to the ISA-Compatible Registers. The PIIX subtractively decodes accesses to all I/O space registers contained within the ISA-Compatible DMA function. This permits another device in the system to implement the compatible DMA function.

PCI master accesses to the ISA-Compatible Registers can be 8, 16, 24, or 32 bits. However, the PIIX only responds to the least significant byte. On writes the other bytes are not loaded and on reads the other bytes have invalid data. The PIIX responds as an 8-bit ISA I/O slave when accessed by an ISA master. See the PCI Local Bus IDE section for accesses to the IDE Register blocks located in the IDE device.

In general, accesses from CPU or PCI masters to the internal PIIX registers are not broadcast to the ISA Bus. Exceptions to this are read/write accesses to 70h and F0h and write accesses to 80h, 84–86h, 88h, 8C–8Eh, 90h, 94–96h, 98h, and 9C–9Eh. These accesses are broadcast to the ISA Bus. Note that aliasing of the 90–9Fh to 80–8Fh can be enabled/disabled via the ISA Controller Recovery Timer Register.

Power Management Registers

There are two power management registers located in normal I/O space. These registers are accessed (by PCI Bus Masters) with 8-bit accesses. The other power management registers are located in PCI configuration space for function 0.

PCI Bus Master IDE Registers

The PCI Bus Master IDE function uses 16 bytes located in normal I/O space, allocated via the BMIBA Register (a PCI base address register). All Bus Master IDE I/O space registers can be accessed as 8, 16, or 32 bit quantities.

Table 2. PCI Configuration Registers—Function 0 (PCI to ISA Bridge)

Configuration Offset	Mnemonic	Register	Register Access
00–01h	VID	Vendor Identification	RO
02–03h	DID	Device Identification	RO
04–05h	PCICMD	PCI Command	R/W
06–07h	PCISTS	PCI Device Status	R/WC
08h	RID	Revision Identification	RO
09h	PI	Programming Interface	RO
0Ah	SUBC	Sub Class Code	RO
0Bh	BCC	Base Class Code	RO
0C–0Dh		Reserved	
0Eh	HEDT	Header Type	RO
0F–4Bh		Reserved	
4Ch	IORT	ISA I/O Controller Recovery Timer	R/W
4Dh		Reserved	
4Eh	XBCS	X-Bus Chip Select Enable	R/W
4F–5Fh		Reserved	
60–63h	PIRQRC[A:D]	PCI IRQ Route Control	R/W
64–68h		Reserved	
69h	TOM	Top of Memory	R/W
6A–6Bh	MSTAT	Miscellaneous Status	R/W
6C–6Fh		Reserved	
70–71h	MBIRQ[1:0]	Motherboard IRQ Route Control	R/W
72–75h		Reserved	
76–77h	MBDMA[1:0]	Motherboard Device DMA Control	R/W
78–79h	PCSC	Programmable Chip Select Control	R/W
7A–9Fh		Reserved	
A0h	SMICNTL	SMI Control	R/W
A1h		Reserved	
A2–A3h	SMIEN	SMI Enable	R/W
A4–A7h	SEE	System Event Enable	R/W

Table 2. PCI Configuration Registers—Function 0 (PCI to ISA Bridge) (Continued)

Configuration Offset	Mnemonic	Register	Register Access
A8h	FTMR	Fast-Off Timer	R/W
A9h		Reserved	
AA–ABh	SMIREQ	SMI Request	R/W
ACh	CTLTMR	Clock Scale STPCLK# Low Timer	R/W
ADh		Reserved	
AEh	CTHTMR	Clock Scale STPCLK# High Timer	R/W
AF–FFh		Reserved	

Table 3. PCI Configuration Registers—Function 1 (IDE Interface)

Configuration Offset	Mnemonic	Register	Register Access
00–01h	VID	Vendor Identification	RO
02–03h	DID	Device Identification	RO
04–05h	PCICMD	Command	R/W
06–07h	PCISTS	PCI Device Status	R/WC
08h	RID	Revision Identification	RO
09h	PI	Programming Interface	RO
0Ah	SUBC	Sub Class Code	RO
0Bh	BCC	Base Class Code	RO
0Ch		Reserved	
0Dh	MLT	Master Latency Timer	R/W
0Eh	HEDT	Header Type	RO
0F–1Fh		Reserved	
20–23h	BMIBA	Bus Master Interface Base Address	R/W
24–3Fh		Reserved	
40–43h	IDETIM	IDE Timing Modes	R/W
44–FFh		Reserved	

Table 4. ISA-Compatible and Power Management Registers

Address (hex)	Address (bits)				Type	Name	Access
	FEDC	BA98	7654	3210			
0000h[3]	0000	0000	000x	0000	r/w	DMA1 CH0 Base and Current Address	PCI
0001h[3]	0000	0000	000x	0001	r/w	DMA1 CH0 Base and Current Count	PCI
0002h[3]	0000	0000	000x	0010	r/w	DMA1 CH1 Base and Current Address	PCI
0003h[3]	0000	0000	000x	0011	r/w	DMA1 CH1 Base and Current Count	PCI
0004h[3]	0000	0000	000x	0100	r/w	DMA1 CH2 Base and Current Address	PCI
0005h[3]	0000	0000	000x	0101	r/w	DMA1 CH2 Base and Current Count	PCI
0006h[3]	0000	0000	000x	0110	r/w	DMA1 CH3 Base and Current Address	PCI
0007h[3]	0000	0000	000x	0111	r/w	DMA1 CH3 Base and Current Count	PCI
0008h[3]	0000	0000	000x	1000	r/w	DMA1 Status(r) Command(w) Register	PCI
0009h[3]	0000	0000	000x	1001	wo	DMA1 Write Request	PCI
000Ah[3]	0000	0000	000x	1010	wo	DMA1 Write Single Mask Bit	PCI
000Bh[3]	0000	0000	000x	1011	wo	DMA1 Write Mode	PCI
000Ch[3]	0000	0000	000x	1100	wo	DMA1 Clear Byte Pointer	PCI
000Dh[3]	0000	0000	000x	1101	wo	DMA1 Master Clear	PCI
000Eh[3]	0000	0000	000x	1110	wo	DMA1 Clear Mask	PCI
000Fh[3]	0000	0000	000x	1111	r/w	DMA1 Read/Write All Mask Register Bits	PCI
0020h	0000	0000	001x	xx00	r/w	INT 1 Control	PCI/ISA
0021h	0000	0000	001x	xx01	r/w	INT 1 Mask	PCI/ISA
0040h	0000	0000	010x	0000	r/w	Timer Counter 1 - Counter 0 Count	PCI/ISA
0041h	0000	0000	010x	0001	r/w	Timer Counter 1 - Counter 1 Count	PCI/ISA
0042h	0000	0000	010x	0010	r/w	Timer Counter 1 - Counter 2 Count	PCI/ISA
0043h	0000	0000	010x	0011	wo	Timer Counter 1 Command Mode	PCI/ISA
0060h[1]	0000	0000	0110	0000	r	Reset XBus IRQ12/M and IRQ1	PCI/ISA
0061h	0000	0000	0110	0001	r/w	NMI Status and Control	PCI/ISA
0070h[1]	0000	0000	0111	0xx0	wo	CMOS RAM Address and NMI Mask Reg	PCI/ISA
0080h[2,3]	0000	0000	100x	0000	r/w	DMA Page (Reserved)	PCI/ISA
0081h[3]	0000	0000	100x	0001	r/w	DMA Channel 2 Page	PCI/ISA
0082h[3]	0000	0000	1000	0010	r/w	DMA Channel 3 Page	PCI/ISA
0083h[3]	0000	0000	100x	0011	r/w	DMA Channel 1 Page	PCI/ISA
0084h[2,3]	0000	0000	100x	0100	r/w	DMA Page (Reserved)	PCI/ISA
0085h[2,3]	0000	0000	100x	0101	r/w	DMA Page (Reserved)	PCI/ISA

Table 4. ISA-Compatible and Power Management Registers (Continued)

Address (hex)	Address (bits)				Type	Name	Access
	FEDC	BA98	7654	3210			
0086h[2,3]	0000	0000	100x	0110	r/w	DMA Page (Reserved)	PCI/ISA
0087h[3]	0000	0000	100x	0111	r/w	DMA Channel 0 Page	PCI/ISA
0088h[2,3]	0000	0000	100x	0100	r/w	DMA Page (Reserved)	PCI/ISA
0089h[3]	0000	0000	100x	1001	r/w	DMA Channel 6 Page	PCI/ISA
008Ah[3]	0000	0000	100x	1010	r/w	DMA Channel 7 Page	PCI/ISA
008Bh[3]	0000	0000	100x	1011	r/w	DMA Channel 5 Page	PCI/ISA
008Ch[2,3]	0000	0000	100x	1100	r/w	DMA Page (Reserved)	PCI/ISA
008Dh[2,3]	0000	0000	100x	1101	r/w	DMA Page (Reserved)	PCI/ISA
008Eh[2,3]	0000	0000	100x	1110	r/w	DMA Page (Reserved)	PCI/ISA
008Fh[3]	0000	0000	100x	1111	r/w	DMA Low Page Register Refresh	PCI/ISA
00A0h	0000	0000	101x	xx00	r/w	INT 2 Control	PCI/ISA
00A1h	0000	0000	101x	xx01	r/w	INT 2 Mask	PCI/ISA
00B2h	0000	0000	1011	0010	r/w	Advanced Power Management Control	PCI
00B3h	0000	0000	1011	0011	r/w	Advanced Power Management Status	PCI
00C0h[3]	0000	0000	1100	000x	r/w	DMA2 CH0 Base and Current Address	PCI
00C2h[3]	0000	0000	1100	001x	r/w	DMA2 CH0 Base and Current Count	PCI
00C4h[3]	0000	0000	1100	010x	r/w	DMA2 CH1 Base and Current Address	PCI
00C6h[3]	0000	0000	1100	011x	r/w	DMA2 CH1 Base and Current Count	PCI
00C8h[3]	0000	0000	1100	100x	r/w	DMA2 CH2 Base and Current Address	PCI
00CAh[3]	0000	0000	1100	101x	r/w	DMA2 CH2 Base and Current Count	PCI
00CCh[3]	0000	0000	1100	110x	r/w	DMA2 CH3 Base and Current Address	PCI
00CEh[3]	0000	0000	1100	111x	r/w	DMA2 CH3 Base and Current Count	PCI
00D0h[3]	0000	0000	1101	000x	r/w	DMA2 Status(r) Command(w)	PCI
00D2h[3]	0000	0000	1101	001x	wo	DMA2 Write Request	PCI
00D4h[3]	0000	0000	1101	010x	wo	DMA2 Write Single Mask Bit	PCI
00D6h[3]	0000	0000	1101	011x	wo	DMA2 Write Mode	PCI
00D8h[3]	0000	0000	1101	100x	wo	DMA2 Clear Byte Pointer	PCI
00DAh[3]	0000	0000	1101	101x	wo	DMA2 Master Clear	PCI
00DCh[3]	0000	0000	1101	110x	wo	DMA2 Clear Mask	PCI
00DEh[3]	0000	0000	1101	111x	r/w	DMA2 Read/Write All Mask Register Bits	PCI
00F0h[1]	0000	0000	1111	0000	wo	Coprocessor Error	PCI/ISA
04D0h	0000	0100	1101	0000	r/w	INT-1 Edge/Level Control	PCI/ISA

Table 4. ISA-Compatible and Power Management Registers (Continued)

Address (hex)	Address (bits)				Type	Name	Access
	FEDC	BA98	7654	3210			
04D1h	0000	0100	1101	0001	r/w	INT-2 Edge/Level Control	PCI/ISA
0CF9h	0000	1100	1111	1001	r/w	Reset Control	PCI

NOTES:
1. Read and write accesses to these locations are always broadcast to the ISA Bus.
2. Write accesses to these locations are broadcast to the ISA Bus. Read Accesses are not. If programmed in the ISA Controller Recovery Timer Register, the PIIX will not alias the 90h–9Fh address range with the following addresses; 80h, 84h–86h, 88h, and 8C–8Eh. In this case, accesses to the 90h–9Fh address range for the previously specified addresses are forwarded to the ISA Bus for both reads and writes and are ignored during ISA Master cycles (i.e., they are no-longer considered PIIX registers). Note that port 92 is always a distinct ISA Register and is always forwarded to the ISA Bus.
3. ISA-Compatible DMA Register I/O space accesses are always subtractively decoded.

Table 5. PCI Bus Master IDE I/O Registers

Offset From Base Address	Mnemonic	Register	Register Access
00h	BMICP	Bus Master IDE Command (Primary)	R/W
01h		Reserved	
02h	BMISP	Bus Master IDE Status (Primary)	R/WC
03h		Reserved	
04–07h	BMIDTPP	Bus Master IDE Descriptor Table Pointer (Primary)	R/W
08h	BMICS	Bus Master IDE Command (Secondary)	R/W
09h		Reserved	
0Ah	BMISS	Bus Master IDE Status (Secondary)	R/WC
0Bh		Reserved	
0C–0Fh	BMIDTPS	Bus Master IDE Descriptor Table Pointer (Secondary)	R/W

NOTE:
The base address is programmable via the BMIBA Register (20-23h; function 1).

2.2 PCI Configuration Registers—Function 0 (PCI to ISA Bridge)

2.2.1 VID—VENDOR IDENTIFICATION REGISTER (FUNCTION 0)

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

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

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

2.2.2 DID—DEVICE IDENTIFICATION REGISTER (FUNCTION 0)

Address Offset: 02–03h
Default Value: 122Eh
Attribute: Read Only

The DID Register contains the device identification number. This register, along with the VID Register, define the PIIX. Writes to this register have no effect.

Bit	Description
15:0	**Device Identification Number:** This is a 16-bit value assigned to the PIIX.

2.2.3 PCICMD—COMMAND REGISTER (FUNCTION 0)

Address Offset: 04–05h
Default Value: 0007h
Attribute: Read/Write

Bit	Description
15:10	**Reserved:** Read as 0.
9	**Fast Back-to-Back Enable:** (Not Implemented). This bit is hardwired to 0.
8:5	**Reserved:** Read as 0.
4	**Postable Memory Write Enable:** (Not Implemented). This bit is hardwired to 0.
3	**Special Cycle Enable (SCE):** 1 = Enable (the PIIX recognizes shutdown special cycle). 0 = Disable (the PIIX ignores all PCI special cycles).
2	**Bus Master Enable (BME):** (Not Implemented). The PIIX does not support disabling it's Bus Master capability. This bit is hardwired to 1.
1	**Memory Access Enable (MAE):** (Not Implemented). The PIIX does not support disabling access to main memory. This bit is hardwired to 1.
0	**I/O Space Access Enable (IOSE):** The PIIX does not support disabling it's response to PCI I/O cycles. This bit is hardwired to 1.

2.2.4 PCISTS—PCI DEVICE STATUS REGISTER (FUNCTION 0)

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

The PCISTS Register reports the occurrence of a PCI master-abort by the PIIX or a PCI target-abort when the PIIX is a master. The register also indicates the PIIX DEVSEL# signal timing.

Bit	Description
15	**Detected Parity Error (PERR):** (Not Implemented). Read as 0.
14	**Signaled SERR# Status (SERRS):** (Not Implemented). Read as 0.
13	**Master-Abort Status (MA)—R/W:** When the PIIX, as a master (for the ISA bridge function), generates a master-abort, MA is set to 1. Software sets MA to 0 by writing 1 to this bit location.
12	**Received Target-Abort Status (RTA)—R/W:** When the PIIX is a master on the PCI Bus (for the ISA bridge function) and receives a target-abort, this bit is set to 1. Software sets RTA to 0 by writing 1 to this bit location.
11	**Signaled Target-Abort Status (STA)—R/W:** This bit is set when the PIIX ISA bridge function is targeted with a transaction that the PIIX terminates with a target abort. Software sets STA to 0 by writing 1 to this bit location.
10:9	**DEVSEL# Timing Status (DEVT)—RO:** The PIIX always generates DEVSEL# with medium timing for ISA functions. Thus, DEVT = 01. This DEVSEL# timing does not include configuration cycles.
8	**PERR# Response:** (Not Implemented). Read as 0.
7	**Fast Back to Back—RO:** This bit indicates to the PCI Master that PIIX as a target is capable of accepting fast back-to-back transactions.
6:0	**Reserved:** Read as 0s.

2.2.5 RID—REVISION IDENTIFICATION REGISTER (FUNCTION 0)

Address Offset: 08h
Default Value: Refer to stepping information
Attribute: Read Only

This 8 bit register contains device stepping information. Writes to this register have no effect.

Bit	Description
7:0	**Revision ID Byte:** This register is hardwired to the default value.

2.2.6 PI—PROGRAMMING INTERFACE REGISTER (FUNCTION 0)

Address Offset: 09h
Default Value: 00h
Attribute: Read Only

This register contains the device programming interface information related to the Sub Class Code Register and Base Class Code Register definition for this function.

Bit	Description
7:0	**Programming Interface:** 00h-hardwired as a PCI-to-ISA bridge.

2.2.7 SUBC—SUB CLASS CODE REGISTER (FUNCTION 0)

Address Offset: 0Ah
Default Value: 01h
Attribute: Read Only

This register indicates the function sub class in relation to the Base Class Code Register.

Bit	Description
7:0	**Sub-Class Code (SCC):** 01h = PCI-to-ISA bridge.

2.2.8 BCC—BASE CLASS CODE REGISTER (FUNCTION 0)

Address Offset: 0Bh
Default Value: 06h
Attribute: Read Only

This register contains the Base Class Code of the PIIX.

Bit	Description
7:0	**Base Class Code (BASEC):** 06h = bridge device.

2.2.9 HEDT—HEADER TYPE REGISTER (FUNCTION 0)

Address Offset: 0Eh
Default Value: 80h
Attribute: Read Only

The HEDT Register identifies the PIIX as a multi-function device.

Bit	Description
7:0	**Device Type (DEVCET):** 80h = multi-function device.

2.2.10 IORT—ISA I/O RECOVERY TIMER REGISTER (FUNCTION 0)

Address Offset: 4Ch
Default Value: 4Dh
Attribute: Read/Write

The I/O recovery mechanism in the PIIX is used to add additional recovery delay between CPU or PCI master originated 8-bit and 16-bit I/O cycles to the ISA Bus. The PIIX automatically forces a minimum delay of 3.5 SYSCLKs between back-to-back 8- and 16-bit I/O cycles to the ISA Bus. This delay is measured from the rising edge of the I/O command (IOR# or IOW#) to the falling edge of the next I/O command. If a delay of greater than 3.5 SYSCLKs is required, the ISA I/O Recovery Time Register can be programmed to increase the delay in increments of SYSCLKs. No additional delay is inserted for back-to-back I/O "sub cycles" generated as a result of byte assembly or disassembly. This register defaults to 8 and 16-bit recovery enabled with one SYSCLK clock added to the standard I/O recovery.

Bit	Description
7	**DMA Reserved Page Register Aliasing Control (DMAAC):** When DMAAC = 0, the PIIX aliases I/O accesses in the 90–9Fh range to the 80–8Fh range. In this case, the PIIX only forwards write accesses to these locations to the ISA Bus. When DMAAC = 1, the PIIX disables aliasing for the following registers; 80h, 84-86h, 88h, and 8C-8Eh. When disabled, the PIIX forwards read and write accesses to those registers to the ISA Bus. Note that port 92h is always a distinct ISA Register in the 90–9Fh range and is always forwarded to the ISA Bus. When DMAAC = 1, ISA master accesses to the 90–9Fh range are ignored by the PIIX. Also, when DMAAC = 1, the PIIX does not re-load the power management Fast-Off-Timer with its original value for accesses to the 90–9Fh address range.
6	**8-Bit I/O Recovery Enable:** 1 = Enable the recovery time programmed in bits[5:3]. 0 = Disable recovery times in bits[5:3] and the recovery timing of 3.5 SYSCLKs is inserted.
5:3	**8-Bit I/O Recovery Times:** When bit 6 = 1, this 3-bit field defines the recovery time for 8-bit I/O. **Bit[5:3] SYSCLK Bit[5:3] SYSCLK** 001 1 101 5 010 2 110 6 011 3 111 7 100 4 000 8
2	**16-Bit I/O Recovery Enable:** 1 = Enable, the recovery times programmed in bits[1:0]. 0 = Disable, programmable recovery times in bits[1:0] and the recovery timing of 3.5 SYSCLKs is inserted.
1:0	**16-Bit I/O Recovery Times:** When bit 2 = 1, this 2-bit field defines the recovery time for 16-bit I/O. **Bit[1:0] SYSCLK** 01 1 10 2 11 3 00 4

2.2.11 XBCS—X-BUS CHIP SELECT REGISTER (FUNCTION 0)

Address Offset: 4Eh
Default Value: 03h
Attribute: Read/Write

This register enables/disables accesses to the RTC, keyboard controller and BIOS. Disabling any of these bits prevents the chip select and X-Bus output enable control signal (XOE#) for that device from being generated. This register also provides coprocessor error and mouse functions.

Bit	Description
7	**Extended BIOS Enable:** When bit 7 = 1 (enabled), PCI master accesses to locations FFF80000–FFFDFFFFh are forwarded to ISA and result in the generation of BIOSCS# and XOE#. When forwarding the additional 384-Kbyte region at the top of 4 Gbytes, the PIIX allows the PCI address A[23:20] to propagate to the ISA LA[23:20] lines as all 1's, aliasing this 384-Kbyte region to the top of the 16-Mbyte space. To avoid contention, ISA add-in memory must not be present in this region (00F80000–00FDFFFFh). When bit 7 = 0, the PIIX does not generate BIOSCS# or XOE#.
6	**Lower BIOS Enable:** When bit 6 = 1 (enabled), PCI master, or ISA master accesses to the lower 64-Kbyte BIOS block (E0000–EFFFFh) at the top of 1 Mbyte, or the aliases at the top of 4 Gbyte (FFFE0000–FFFEFFFFh) result in the generation of BIOSCS# and XOE#. When forwarding the region at the top of 4 Gbytes to the ISA Bus, the ISA LA[23:20] lines are all 1's, aliasing this region to the top of the 16 Mbyte space. To avoid contention, ISA addin memory must not be present in this region (00F80000–00FDFFFFh). When bit 6 = 0, the PIIX does not generate BIOSCS# or XOE# during these accesses and does not forward the accesses to ISA.
5	**Coprocessor Error Function Enable:** 1 = Enable; the FERR# input, when asserted, triggers IRQ13 (internal). FERR# is also used to gate the IGNNE# output.
4	**IRQ12/M Mouse Function Enable:** 1 = Mouse function; 0 = Standard IRQ12 interrupt function.
3	**Reserved.**
2	**BIOSCS# Write Protect Enable:** 1 = Enable (BIOSCS# is asserted for BIOS memory read and write cycles in decoded BIOS region); 0 = Disable (BIOSCS# is only asserted for BIOS read cycles).
1	**Keyboard Controller Address Location Enable:** 1 = Enable KBCS# and XOE# for address locations 60h and 64h. 0 = Disable KBCS#/XOE# for accesses to these locations.
0	**RTC Address Location Enable:** 1 = Enable RTCCS#/RTCALE and XOE# for accesses to address locations 70–77h. 0 = Disable RTCCS#/RTCALE and XOE# for these accesses.

2.2.12 PIRQRC[A:D]—PIRQx ROUTE CONTROL REGISTERS (FUNCTION 0)

Address Offset: 60h (PIRQRCA#)–63h (PIRQRCD#)
Default Value: 80h
Attribute: Read/Write

These registers control the routing of the PIRQ[A:D]# signals to the IRQ inputs of the interrupt controller. Each PIRQx# can be independently routed to any one of 11 interrupts. All four PIRQx# lines can be routed to the same IRQx input. Note that the IRQ that is selected through bits[3:0] must be set to level sensitive mode in the corresponding ELCR Register. When a PIRQ signal is routed to an interrupt controller IRQ, the PIIX masks the corresponding IRQ signal.

Bit	Description
7	**Interrupt Routing Enable:** 0 = Enable; 1 = Disable
6:4	**Reserved.** Read as 0s.
3:0	**Interrupt Routing:** When bit 7 = 0, this field selects the routing of the PIRQx to one of the interrupt controller interrupt inputs.

Bits[3:0]	IRQ Routing	Bits[3:0]	IRQ Routing	Bits[3:0]	IRQ Routing
0000	Reserved	0110	IRQ6	1011	IRQ11
0001	Reserved	0111	IRQ7	1100	IRQ12
0010	Reserved	1000	Reserved	1101	Reserved
0011	IRQ3	1001	IRQ9	1110	IRQ14
0100	IRQ4	1010	IRQ10	1111	IRQ15
0101	IRQ5				

2.2.13 TOM—TOP OF MEMORY REGISTER (FUNCTION 0)

Address Offset: 69h
Default Value: 02h
Attribute: Read/Write

This register enables the forwarding of ISA and DMA memory cycles to the PCI Bus and sets the top of main memory accessible by ISA or DMA devices. In addition, this register controls the forwarding of ISA or DMA accesses to the lower BIOS region (E0000-EFFFFh) and the 512–640-Kbyte main memory region (80000-A0000h). The Top of Memory Configuration Register must be set by the BIOS.

Bit	Description
7:4	**Top Of Memory:** The top of memory can be assigned in 1-Mbyte increments from 1–16 Mbytes. ISA or DMA accesses within this region, and not in the memory hole region, are forwarded to PCI.

Bits[7:4]	Top of Memory	Bits[7:4]	Top of Memory	Bits[7:4]	Top of Memory
0000	1 Mbyte	0110	7 Mbyte	1011	12 Mbyte
0001	2 Mbyte	0111	8 Mbyte	1100	13 Mbyte
0010	3 Mbyte	1000	9 Mbyte	1101	14 Mbyte
0011	4 Mbyte	1001	10 Mbyte	1110	15 Mbyte
0100	5 Mbyte	1010	11 Mbyte	1111	16 Mbyte
0101	6 Mbyte				

NOTE:
The PIIX only supports a main memory hole at the top of 16 Mbytes. Thus, if a 1-Mbyte memory hole is created for the TSC's DRAM controller between 15 and 16 Mbytes, the PIIX Top of Memory should be set at 15 Mbytes.

3	**ISA/DMA Lower BIOS Forwarding Enable:** 1 = Enable (forwarded to PCI, if XBCS Register bit 6 = 0); 0 = Disable (contained to ISA). Note that if the XBCS Register bit 6 = 1, ISA/DMA accesses in this region are always contained to ISA.
2	**Reserved.**
1	**ISA/DMA 512–640 Kbyte Region Forwarding Enable:** 1 = Enable (forwarded to PCI); 0 = Disable (contained to ISA).
0	**Reserved.**

2.2.14 MSTAT—MISCELLANEOUS STATUS REGISTER (FUNCTION 0)

Address Offset: 6B–6Ah
Default Value: Undefined
Attribute: Read Only (bit 2 read/write)

This register reports the hardware strapping options selected for internal ISA DMA or external DMA mode and the ISA clock divisor.

Bit	Description
15:3	**Reserved.**
2	**PCI Header Type Bit Enable:** This bit controls the "Header Type Bit" in PIIX register 0Eh which defines the PIIX as a multifunction device. This bit defaults to 1 (multifunction device), and should be left in the default state. This bit is read/write.
1	**Internal ISA DMA or External DMA Mode Status (IEDMAS):** This bit reports the strapping option selected on the TC signal. This bit is 0 for normal DMA operation. This bit indicates strapping at the TC pin during reset (pulled high at reset for a value of 0).
0	**ISA Clock Divisor Status:** This bit reports the strapping option on the SYSCLK signal. 1 = clock divisor of 3 (PCICLK = 25 MHz). 0 = Clock divisor of 4 (PCICLK = 33 MHz). Note that, for PCICLK = 30 MHz, a clock divisor of 4 must be selected and produces a SYSCLK of 7.5 Mhz.

2.2.15 MBIRQ[1:0]—MOTHERBOARD DEVICE IRQ ROUTE CONTROL REGISTERS (FUNCTION 0)

Address Offset: 70h—MBIRQ0; 71h—MBIRQ1
Default Value: 80h
Attribute: Read/Write

These registers control the routing of motherboard device interrupts (MIRQ[1:0]) to the internal IRQ inputs of the interrupt controller. Each MIRQx# can be independently routed to any one of the interrupts. If the Bus Master mode of the IDE interface is used and there is a secondary IDE channel, the interrupt for that channel must be connected to MIRQ0.

Note that when a MIRQ line and a PIRQ# line are steered to the same ISA interrupt, the device connected to the MIRQ line must be set for active high, level-sensitive interrupts. In this case, the ISA interrupt will be masked. Bit 6 of that Motherboard Device IRQ Route Control Register must be programmed to 0.

Bit	Description
7	**Interrupt Routing Enable:** 0 = Enable routing; 1 = Disable routing.
6	**MIRQx/IRQx Sharing Enable:** 0 = Disable sharing; 1 = Enable sharing. When sharing is disabled and bit 7 of this register is 0, the interrupt specified by bits[3:0] is masked. Interrupt sharing should only be enabled when the device connected to the MIRQ line and the device connected to the ISA IRQ line both produce active high, level-sensitive interrupts.
5:4	**Reserved:** Read as 0s.
3:0	**Interrupt Routing:** When bit 7 = 0, this field selects the routing of the MBIRQx to one of the interrupt controller interrupt inputs.

Bits[3:0]	IRQ Routing	Bits[3:0]	IRQ Routing	Bits[3:0]	IRQ Routing
0000	Reserved	0110	IRQ6	1011	IRQ11
0001	Reserved	0111	IRQ7	1100	IRQ12
0010	Reserved	1000	Reserved	1101	Reserved
0011	IRQ3	1001	IRQ9	1110	IRQ14
0100	IRQ4	1010	IRQ10	1111	IRQ15
0101	IRQ5				

2.2.16 MBDMA[1:0]—MOTHERBOARD DEVICE DMA CONTROL REGISTERS (FUNCTION 0)

Address Offset: 76h—MBDMA0#; 77h—MBDMA1#
Default Value: 04h
Attribute: Read/Write

These registers control the routing of motherboard device DMA signals (MDRQ[1:0] and MDAK[1:0]) to the DREQ and DACK# signals on the 8237 DMA controller unit. This register also enables a fast transfer mode (type F, 3 SYSCLK) for motherboard devices.

When a MDRQ/MDAK# pair is programmed for a given 8237 DMA channel and DMC=1, the MDRQ/MDAK# signals are masked. If both motherboard DMAs are used, the motherboard DMAs should be programmed to different compatible DMA channels. Programming both motherboard DMAs to the same compatible DMA channel results in unpredictable device operation.

When DMC=1, this register enables type F transfers and the 4-byte DMA buffer for an ISA peripheral on a given channel. When DMC=0, this register steers the corresponding MDRQ/MDAK# signals to a compatible ISA channel for a motherboard peripheral and also enable type F transfers and the 4-byte DMA buffer.

Bit	Description
7	**Type F and DMA Buffer Enable (FAST):** 1=Enable for the channel selected by bits[2:0]. 0=Disable for the channel selected by bits[2:0].
6:4	**Reserved.** Read as 0s.
3	**Disable Motherboard Channel (DMC):** When this bit 3=0, the MDRQ/MDAK# pair associated with this channel is routed to the compatible ISA channel determined by the CHNL field (bits[2:0]). When bit 3=1, the ISA DREQ/DACK# pair is used for that channel.
2:0	**DMA Channel Select (CHNL):** This field selects the DMA channel connected to the MDRQ/MDAK# pair. **Bits[2:0] DMA Channel Bits[2:0] DMA Channel** 000 0 100 default (disabled) 001 1 101 5 010 2 110 6 011 3 111 7

2.2.17 PCSC—PROGRAMMABLE CHIP SELECT CONTROL REGISTER (FUNCTION 0)

Address Offset: 78–79h
Default Value: 0002h
Attribute: Read/Write

This register controls the assertion of the PCS# programmable chip select signal. The PCS# signal is asserted for subtractively decoded I/O cycles generated by PCI masters that fall in the range specified by this register. The address is programmable to any 16 bit I/O space location and the range is programmable to be 4, 8 or 16 bytes. A split range is precluded. The upper 16 address bits (AD[31:16]) must be 0 for the address to be decoded and the PCS# signal asserted. The PCS# signal is never asserted for ISA Bus Masters access.

Bit	Description
15:2	**PCS Address (PCSADDR):** This field defines a 16 bit I/O space address (4 byte range) that causes the PCS# signal to assert. Address bits [3:2] may be masked (considered "don't care") by programming bits [1:0] of this register.
1:0	**PCS Address Mask:** When bit 1 = 1, PCSADDR3 is masked. When bit 0 = 1, PCSADDR2 is masked. **Bits[1:0] Range** 00 4 bytes (default) 01 8 bytes, contiguous 10 Disabled 11 16 bytes, contiguous

2.2.18 SMICNTL—SMI CONTROL REGISTER (FUNCTION 0)

Address Offset: A0h
Default Value: 08h
Attribute: Read/Write

The SMICNTL Register provides Fast-Off Timer control, STPCLK# enable/disable, and CPU clock scaling. This register also enables/disables the system management interrupt (SMI).

Bit	Description
7:5	**Reserved.**
4:3	**Fast-Off Timer Freeze (CTMRFRZ):** This field enables/disables the Fast-Off Timer and when enabled, selects the timer count granularity as shown below: **Bits[4:3]** / **Count Granularity (33 MHz PCICLK)** / **Count Granularity (30 MHz PCICLK)** / **Count Granularity (25 MHz PCICLK)** 00 / 1 Minute / 1.1 Minute / 1.32 Minute 01 / Disabled (default) / Disabled (default) / Disabled (default) 10 / 1 PCICLK / 1 PCICLK / 1 PCICLK 11 / 1 msec / 1.1 msec / 1.32 msec
2	**STPCLK# Scaling Enable (CSTPCLKSC):** 1 = Enable; 0 = Disable. When enabled (and bit 1 = 1), the high and low times for the STPCLK# signal are controlled by the Clock Scaling STPCLK# High Timer and Clock Scaling STPCLK# Low Timer Registers.
1	**STPCLK# Signal Enable (CSTPCLKE):** 1 = Enable; 0 = Disable. When enabled, an APMC Register read causes STPCLK# to be asserted. When disabled, the STPCLK# signal is disabled and is negated (high). Software can set this bit to 0 by writing 0 to it.
0	**SMI# Gate (CSMIGATE):** 1 = Enable; 0 = Disable. When enabled, a system management interrupt condition asserts the SMI# signal. When disabled, the SMI# signal is masked and negated. This bit only affects the SMI# signal and does not affect the detection/recording of SMI events (i.e., this bit does not affect the SMI status bits in the SMIREQ Register). Thus, if an SMI is pending when this bit is set to 1, the SMI# signal is asserted.

2.2.19 SMIEN—SMI ENABLE REGISTER (FUNCTION 0)

Address Offset: A2–A3h
Default Value: 0000h
Attribute: Read/Write

This register enables the generation of SMI (asserting the SMI# signal) for the associated hardware events (bits[5:0]), and software events (bit 7). When a hardware event is enabled, the occurrence of a corresponding event results in the assertion of SMI#, if enabled via the SMICNTL Register. The SMI# is asserted independent of the current power state (Power-On or Fast-Off). The default for all sources in this register is disabled.

Bit	Description
15:8	**Reserved.**
7	**APMC Write SMI Enable:** 1 = Enable; 0 = Disable.
6	**EXTSMI# SMI Enable:** 1 = Enable; 0 = Disable.
5	**Fast-Off Timer SMI Enable:** 1 = Enable; 0 = Disable. When enabled, the timer generates an SMI when it decrements to 0.
4	**IRQ12 SMI Enable (PS/2 Mouse Interrupt):** 1 = Enable; 0 = Disable.
3	**IRQ8 SMI Enable (RTC Alarm Interrupt):** 1 = Enable; 0 = Disable.
2	**IRQ4 SMI Enable (COM2/COM4 Interrupt or Mouse):** 1 = Enable; 0 = Disable.
1	**IRQ3 SMI Enable (COM1/COM3 Interrupt or Mouse):** 1 = Enable; 0 = Disable.
0	**IRQ1 SMI Enable (Keyboard Interrupt):** 1 = Enable; 0 = Disable.

2.2.20 SEE—SYSTEM EVENT ENABLE REGISTER (FUNCTION 0)

Address Offset: A4–A7h
Default Value: 00000000h
Attribute: Read/Write

This register enables hardware events as system events and break events for power management control. The default for each system/break event in this register is disabled. Bits[31,29,15:3,1:0] generate both system and break events and bit 30 generates break events only.

System Events: Activity by these events can keep the system from powering down. When a system event is enabled, the corresponding hardware event activity prevents a Fast-Off power-down condition by reloading the Fast-Off Timer with its initial count.

Break Events: These events can awaken a powered down system. When a break event is enabled, the corresponding hardware event activity powers up the system by negating STPCLK#.

Bit	Description
31	**Fast-Off SMI Enable (FSMIEN):** 1 = Enable; 0 = Disable.
30	**INTR Enable (FINTREN):** 1 = Enable; 0 = Disable. When enabled, INTR is used as a global break event. In this case, any IRQ that is generated causes the system to powerup via the negation of STPCLK#, regardless of the state of bits[15:3,1:0] in this register.
29	**Fast-Off NMI Enable (FNMIEN):** 1 = Enable; 0 = Disable.
28:16	**Reserved.**
15:3	**Fast-Off IRQ[15:3] Enable (FIRQ[15:3]EN):** 1 = Enable; 0 = Disable.
2	**Reserved.**
1:0	**Fast-Off IRQ[1:0] Enable (FIRQ[1:0]EN):** 1 = Enable; 0 = Disable.

2.2.21 FTMR—FAST-OFF TIMER REGISTER (FUNCTION 0)

Address Offset: A8h
Default Value: 0Fh
Attribute: Read/Write

The Fast-Off Timer indicates (through an SMI) that the system has been idle for a preprogrammed period of time. When the timer expires, an SMI special cycle is generated. The count time interval is programmable (via the SMICNTL Register). The granularity of the counter is programmable via the SMICNTL Register.

NOTE:

1. Before writing to the FTMR Register, the Fast-Off Timer must be stopped via bits[4:3] of the SMICNTL Register.

Bit	Description
7:0	**Fast-Off Timer Value:** Bits[7:0] contain one less than the actual count-down value. Thus, if X is programmed into this register, the countdown value is X+1. The X+1 value is loaded into the counter when an enabled system event occurs. When the Fast-Off Timer reaches 00h, an SMI is generated and the timer is re-loaded with the X+1 value. When the Fast-Off Timer is enabled (via the SMICNTL Register), the timer counts down from this value. A read from the FTMR Register returns the value last written.

2.2.22 SMIREQ—SMI REQUEST REGISTER (FUNCTION 0)

Address Offset: AA–ABh
Default Value: 00h
Attribute: Read/Write

The SMIREQ Register contains status bits indicating which enabled event caused a SMI.

NOTES:

1. The SMIREQ bits are set, cleared, or read independently of each other and independently of the CSMIGATE bit in the SMICNTL Register.
2. If software attempts to set a status bit to 0 at the same time that the PIIX is setting it to 1, the bit is set to 1.
3. Each of the SMIREQ bits is set by the PIIX in response to the activation of the corresonding SMI event. If the SMI event is still active when the corresponding SMIREQ bit is set to 0, the PIIX does not set the status bit back to 1 (i.e., there is only one status indication per active SMI event).
4. When an IRQx signal is asserted, the corresponding IRQx status bit is set to 1. If the IRQx signal is still active when software sets the corresponding status bit to 0, the status bit is not set back to 1. The IRQx may be negated before software sets the status bit to 0. However, if the status bit is set to 0 at the same time a new IRQx is activated, the status bit remains at 1. This indicates to the SMI handler that a new SMI event has been detected.
5. If an IRQx is set in level mode and shared by two devices, the IRQ should not be enabled as an SMI# event. The PIIX's SMIREQ bits are essentially set with an edge. When the second IRQ occurs on a shared IRQ, there is not second edge and the SMI# will not be generated for the second IRQ.

Bit	Description
15:8	**Reserved.**
7	**APM SMI Status (RAPMC):** The PIIX sets this bit to 1 to indicate that a write to the APM Control Register caused an SMI. Software sets this bit to 0 by writing 0 to it.
6	**EXTSMI# SMI Status (REXT):** The PIIX sets this bit to 1 to indicate that EXTSMI# caused an SMI. Software sets this bit to 0 by writing 0 to it.
5	**Fast-Off Timer Expired Status (RFOT):** The PIIX sets this bit to 1 to indicate that the Fast-Off Timer expired and caused an SMI. Software sets this bit to 0 by writing 0 to it. Note that the timer re-starts counting 1 the next clock after it expires.
4	**IRQ12 Request SMI Status (RIRQ12):** The PIIX sets this bit to 1 to indicate that IRQ12 caused an SMI. Software sets this bit to 0 by writing 0 to it.
3	**IRQ8# Request SMI Status (RIRQ8):** The PIIX sets this bit to 1 to indicate that IRQ8# caused an SMI. Software sets this bit to 0 by writing 0 to it.
2	**IRQ4 Request SMI Status (RIRQ4):** The PIIX sets this bit to 1 to indicate that IRQ4 caused an SMI. Software sets this bit to 0 by writing 0 to it.
1	**IRQ3 Request SMI Status (RIRQ3):** The PIIX sets this bit to 1 to indicate that IRQ3 caused an SMI. Software sets this bit to 0 by writing 0 to it.
0	**IRQ1 Request SMI Status (RIRQ1):** The PIIX sets this bit to 1 to indicate that IRQ1 caused an SMI. Software sets this bit to 0 by writing 0 to it.

2.2.23 CTLTMR—CLOCK SCALE STPCLK# LOW TIMER (FUNCTION 0)

Address Offset: ACh
Default Value: 00h
Attribute: Read/Write

The value in this register defines the duration of the STPCLK# asserted period when bit 2 in the SMICNTL Register is set to 1. The value in this register is loaded into the STPCLK# Timer when STPCLK# is asserted. The STPCLK# timer is a divide of PCI clocks and is, therefore, frequency dependent.

The base count for a value of 0 is as follows:

50.0 MHz timebase is 42 microseconds
60.0 MHz timebase is 35 microseconds
66.6 MHz timebase is 32 microseconds

The numbers above are derived from the formula: # of PCI clocks STPCLK# asserted (or negated) = 1+ 1056* (programmed value + 1) where "programmed value" = the value programmed in the clock scale STPCLK# low or high timers, register offset 0ACh and 0AEh.

Bit	Description
7:0	**Clock Scaling STPCLK# Low Timer Value:** Bits[7:0] define the duration of the STPCLK# asserted period during clock throttling.

2.2.24 CTHTMR—CLOCK SCALE STPCLK# HIGH TIMER (FUNCTION 0)

Address Offset: AEh
Default Value: 00h
Attribute: Read/Write

The value in this register defines the duration of the STPCLK# negated period when bit 2 in the SMICNTL Register is set to 1. The value in this register is loaded into the STPCLK# timer when STPCLK# is negated. The STPCLK# timer is a divide of PCI clicks and is, therefore, frequency dependent. See the STPCLK# Low Timer description in Section 2.2.23..

Bit	Description
7:0	**Clock Scaling STPCLK# High Timer Value:** Bits[7:0] define the duration of the STPCLK# negated period during clock throttling.

2.3 PCI Configuration Registers—Function 1 (IDE Interface)

The PIIX is a multi-function device, as indicated by bit 7 of the Header Type Register. The PCI IDE interface function uses Function 1.

2.3.1 VID—VENDOR ID REGISTER (FUNCTION 1)

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

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

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

2.3.2 DID—DEVICE IDENTIFICATION REGISTER (FUNCTION 1)

Address Offset: 02–03h
Default Value: 1230h
Attribute: Read Only

The DID Register contains the device identification number. This register, along with the VID Register, define the PIIX. Writes to this register have no effect.

Bit	Description
15:0	**Device Identification Number:** This is a 16-bit value assigned to the PIIX.

2.3.3 PCICMD—COMMAND REGISTER (FUNCTION 1)

Address Offset: 04–05h
Default Value: 0000h
Attribute: Read/Write

The PCICMD Register controls access to the I/O Space Registers.

Bit	Description
15:10	**Reserved:** Read 0.
9	**Fast Back to Back Enable (FBE): (Not Implemented).** This bit is hardwired to 0.
8:5	**Reserved:** Read as 0.
4	**Memory Write and Invalidate Enable (MWI): (Not Implemented).** This bit is hardwired to 0.
3	**Special Cycle Enable (SCE): (Not Implemented).** This bit is hardwired to 0.
2	**Bus Master Enable (BME):** 1 = Enables the PIIX to be an IDE Bus Master. 0 = Disables the PIIX from generating PCI accesses for the IDE Bus Master function. This bit must be programmed to 1 by BIOS for Bus Master IDE operation.
1	**Memory Space Enable (MSE): (Not Implemented).** This bit is hardwired to 1.
0	**I/O Space Enable (IOSE):** This bit controls access to the I/O Space Registers. When IOSE = 1, access to the Legacy IDE ports (both primary and secondary) and the PCI Bus Master IDE I/O Registers is enabled. The Base Address Register for the PCI Bus Master IDE I/O Registers should be programmed before this bit is set to 1.

2.3.4 PCISTS—PCI DEVICE STATUS REGISTER (FUNCTION 1)

Address Offset: 06–07h
Default Value: 0280h
Attribute: Read/Write

PCISTS is a 16-bit status register for the IDE interface function. The register also indicates the PIIX's DEVSEL# signal timing.

Bit	Description
15	**Detected Parity Error (PERR): (Not Implemented).** Read as 0.
14	**SERR# Status (SERRS): (Not Implemented).** Read as 0.
13	**Master-Abort Status (MAS)—R/W:** When the Bus Master IDE interface function, as a master, generates a master abort, MA is set to 1. Software sets MA to 0 by writing 1 to this bit.
12	**Received Target-Abort Status (RTA)—R/W:** When the Bus Master IDE interface function is a master on the PCI Bus and receives a target abort, this bit is set to 1. Software sets RTA to 0 by writing 1 to this bit.
11	**Signaled Target Abort Status (STA)—R/W:** This bit is set when the PIIX IDE interface function is targeted with a transaction that the PIIX terminates with a target abort. Software resets STA to 0 by writing 1 to this bit.
10:9	**DEVSEL# Timing Status (DEVT)—RO:** For the PIIX, DEVT = 01 indicating medium timing for DEVSEL# assertion when performing a positive decode. DEVSEL# timing does not include configuration cycles.
8	**Data Parity Detected (DPD): (Not Implemented).** Read as 0.
7	**Fast Back-to-Back Capable (FBC)—RO:** Hardwired to 1. This bit indicates to the PCI Master that PIIX, as a target, is capable of accepting fast back-to-back transactions.
6:0	**Reserved:** Read as 0s.

2.3.5 RID—REVISION IDENTIFICATION REGISTER (FUNCTION 1)

Address Offset: 08h
Default Value: Refer to stepping information
Attribute: Read Only

This 8-bit register contains device stepping information. Writes to this register have no effect.

Bit	Description
7:0	**Revision ID Byte:** The register is hardwired to the default value during manufacturing.

2.3.6 PI—PROGRAMMING INTERFACE REGISTER (FUNCTION 1)

Address Offset: 09h
Default Value: 80h
Attribute: Read Only

This register contains the device programming interface information related to the Sub Class Code Register and Base Class Code Register definition for this function.

Bit	Description
7:0	**Programming Interface:** 80h = Capable of IDE Bus Master operation.

2.3.7 SUBC—SUB CLASS CODE REGISTER (FUNCTION 1)

Address Offset: 0Ah
Default Value: 01h
Attribute: Read Only

This register indicates the function sub-class in relation to the Base Class Code Register.

Bit	Description
7:0	**Sub Class Code (SUBC):** 01h = IDE controller.

2.3.8 BCC—BASE CLASS CODE REGISTER (FUNCTION 1)

Address Offset: 0Bh
Default Value: 01h
Attribute: Read Only

This register contains the Base Class Code of the IDE function on the PIIX.

Bit	Description
7:0	**Base Class Code (BASEC):** 01h = Mass storage device.

2.3.9 MLT—MASTER LATENCY TIMER REGISTER (FUNCTION 1)

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

MLT controls the amount of time PIIX, as a Bus Master, can burst data on the PCI Bus. The count value is an 8 bit quantity. However, MLT[3:0] are reserved and 0 when determining the count value. MLT is cleared and suspended when PIIX is not asserting FRAME#. When PIIX asserts FRAME#, the counter begins counting. If PIIX finishes its transaction before the count expires, the MLT count is ignored. If the count expires before the transaction completes (count = # of clocks programmed in MLT), PIIX initiates a transaction termination as soon as its PHLDA# is removed. The number of clocks programmed in the MLT represents the guaranteed time slice (measured in PCI clocks) allotted to PIIX. The default value of MLT is 00h, or 0 PCI clocks.

Bit	Description
7:4	**Master Latency Timer Count Value:** PIIX-initiated PCI burst cycles can last indefinitely, as long as PHLDA# remains active. However, if PHLDA# is negated after the burst cycle is initiated, PIIX limits the burst cycle to the number of PCI Bus clocks specified by this field.
3:0	**Reserved.**

2.3.10 HEDT—HEADER TYPE REGISTER (FUNCTION 1)

Address Offset: 0Eh
Default Value: 80h
Attribute: Read Only

The HEDT Register identifies the PIIX as a multi-function device.

Bit	Description
7:0	**Device Type (DEVICET):** 80h = Multi-function device.

2.3.11 BMIBA—BUS MASTER INTERFACE BASE ADDRESS REGISTER (FUNCTION 1)

Address Offset: 20–23h
Default Value: 00000001h
Attribute: Read/Write

This register selects the base address of a 16-byte I/O space to provide a software interface to the Bus Master functions. Only 12 bytes are actually used (6 bytes for primary and 6 bytes for secondary).

Bit	Description
31:16	**Reserved:** Hardwired to 0.
15:4	**Bus Master Interface Base Address:** These bits provide the base address for the Bus Master Interface Registers and correspond to AD[15:4].
3:2	**Reserved:** Hardwired to 0.
1	**Reserved.**
0	**Resource Type Indicator (RTE)—RO:** This bit is hardwired to 1 indicating that the base address field in this register maps to I/O space.

2.3.12 IDETM—IDE TIMING REGISTER (FUNCTION 1)

Address Offset: Primary Channel = 40–41h; Secondary Channel = 42–43h
Default Value: 0000h
Attribute: Read/Write Only

This register controls the PIIX's IDE interface and selects the timing characteristics of the PCI Local Bus IDE cycle.

Bit	Description
15	**IDE Decode Enable (IDE):** 1 = Enable; 0 = Disable. When enabled, I/O transactions on PCI targeting the IDE ATA Register blocks (command block and control block) are positively decoded on PCI and driven on the PIIX IDE interface. When disabled, PIIX subtractively decodes these accesses to ISA.
14	**Reserved.**
13:12	**IORDY Sample Point (ISP):** This field selects the number of clocks between DIOx# assertion and the first IORDY sample point. **Bits[13:12] Number Of Clocks** 00 5 01 4 10 3 11 2
11:10	**Reserved.**
9:8	**Recovery Time (RTC):** This field selects the minimum number of clocks between the last IORDY# sample point and the DIOx# strobe of the next cycle. **Bits[9:8] Number Of Clocks** 00 4 01 3 10 2 11 1
7	**DMA Timing Enable Only (DTE1):** When DTE1 = 1, fast timing mode is enabled for DMA data transfers for drive 1. Note that PIO transfers to the IDE data port still run in compatible timing.
6	**Prefetch and Posting Enable (PPE1):** When PPE1 = 1, prefetch and posting to the IDE data port is enabled for drive 1.
5	**IORDY Sample Point Enable Drive Select 1 (IE1):** When IE1 = 0, IORDY sampling is disabled for Drive 1. The internal IORDY signal is forced asserted guaranteeing that IORDY is sampled asserted at the first sample point as specified by the ISP field in this register. When IE1 = 1 and the currently selected drive (via a copy of bit 4 of 1x6h) is Drive 0, all accesses to the enabled I/O address range sample IORDY. The IORDY sample point is specified by the ISP field in this register.

Bit	Description
4	**Fast Timing Bank Drive Select 1 (TIME1):** When TIME1 = 0, accesses to the data port of the enabled I/O address range use the 16 bit compatible timing PCI local bus path. When TIME1 = 1 and the currently selected drive (via a copy of bit 4 of 1x6h) is Drive 1, accesses to the data port of the enabled I/O address range use the fast timing bank PCI local bus IDE path. Accesses to the data port use fast timing only if bit 7 of this register (DTE1) is zero. Accesses to all non-data ports of the enabled I/O address range use the 8 bit compatible timing PCI local bus path.
3	**DMA Timing Enable Only (DTE0):** When DTE0 = 1, fast timing mode is enabled for DMA data transfers for drive 0. Note that PIO transfers to the IDE data port still run in compatible timing.
2	**Prefetch and Posting Enable (PPE0):** 1 = Enable; 0 = Disable. When enabled, prefetch and posting to the IDE data port is enabled for drive 0.
1	**IORDY Sample Point Enable Drive Select 0 (IE0):** When IE0 = 0, IORDY sampling is disabled for Drive 0. The internal IORDY signal is forced asserted guaranteeing that IORDY is sampled asserted at the first sample point as specified by the ISP field in this register. When IE0 = 1 and the currently selected drive (via a copy of bit 4 of 1x6h) is Drive 0, all accesses to the enabled I/O address range sample IORDY. The IORDY sample point is specified by the ISP field in this register.
0	**Fast Timing Bank Drive Select 0 (TIME0):** When TIME0 = 0, accesses to the data port of the enabled I/O address range uses the 16 bit compatible timing PCI local bus path. When TIME0 = 1 and the currently selected drive (via a copy of bit 4 of 1x6h) is Drive 0, accesses to the data port of the enabled I/O address range use the fast timing bank PCI local bus IDE path. Accesses to the data port use fast timing only if bit 3 of this register (DTE0) is 0. Accesses to all non-data ports of the enabled I/O address range use the 8 bit compatible timing PCI local bus path.

2.4 ISA-Compatible Registers

The ISA-Compatible Registers contain the DMA, timer/counter, and interrupt registers. This group also contains the X-Bus, coprocessor, NMI, and reset registers.

2.4.1 DMA REGISTERS

The PIIX contains DMA circuitry that incorporates the functionality of two 82C37 DMA controllers (DMA1 and DMA2). The DMA Registers control the operation of the DMA controllers and are all accessible from the Host CPU via the PCI Bus interface. In addition, some of the registers are accessed from the ISA Bus via ISA I/O space. Unless otherwise stated, a CPURST sets each register to its default value.

2.4.1.1 DCOM—DMA Command Register

I/O Address: Channels 0–3—08h; Channels 4–7—0D0h
Default Value: 00h (CPURST or Master Clear)
Attribute: Write Only

This 8-bit register controls the configuration of the DMA. Note that disabling channels 4-7 also disables channels 0–3, since channels 0–3 are cascaded into channel 4.

Bit	Description
7	**DACK# ACTIVE Level (DACK#[3:0,(7:5)]):** 1 = Active high; 0 = Active low.
6	**DREQ Sense Assert Level (DREQ[3:0,(7:5)]):** 1 = Active low; 0 = Active high.
5	**Reserved:** Must be 0.
4	**DMA Group Arbitration Priority:** 1 = Rotating priority; 0 = Fixed priority
3	**Reserved:** Must be 0.
2	**DMA Channel Group Enable:** 1 = Disable; 0 = Enable.
1:0	**Reserved:** Must be 0.

2.4.1.2 DCM—DMA Channel Mode Register

I/O Address: Channels 0–3 = 0Bh; Channels 4–7 = 0D6h
Default Value: Bits[7:2] = 0, Bits[1:0] = undefined (CPURST or Master Clear)
Attribute: Write Only

Each channel has a 16-bit DMA Channel Mode Register. The Channel Mode Registers provide control over DMA transfer type, transfer mode, address increment/decrement, and autoinitialization.

Bit	Description
7:6	**DMA Transfer Mode:** Each DMA channel can be programmed in one of four different modes: **Bits[7:6] Transfer Mode** 00 Demand mode 01 Single mode 10 Block mode 11 Cascade mode
5	**Address Increment/Decrement Select:** 0 = Increment; 1 = Decrement.
4	**Autoinitialize Enable:** 1 = Enable; 0 = Disable.

Bit	Description
3:2	**DMA Transfer Type:** When Bits[7:6] = 11, the transfer type bits are irrelevant. **Bits[3:2] Transfer Type** 00 Verify transfer 01 Write transfer 10 Read transfer 11 Illegal
1:0	**DMA Channel Select:** Bits[1:0] select the DMA Channel Mode Register written to by bits[7:2]. **Bits[1:0] Channel** 00 Channel 0 (4) 01 Channel 1 (5) 10 Channel 2 (6) 11 Channel 3 (7)

2.4.1.3 DR—DMA Request Register

I/O Address: Channels 0–3—09h; Channels 4–7—0D2h
Default Value: Bits[1:0] = undefined, Bits[7:2] = 0 (CPURST or Master Clear)
Attribute: Write Only

The Request Register is used by software to initiate a DMA request. The DMA responds to the software request as though DREQx is asserted. These requests are non-maskable and subject to prioritization by the priority encoder network. For a software request, the channel must be in Block Mode. The Request Register status for DMA1 and DMA is output on bits[7:4] of a Status Register read.

Bit	Description
7:3	**Reserved**: Must be 0.
2	**DMA Channel Service Request:** 0 = Resets the individual software DMA channel request bit. 1 = Sets the request bit. Generation of a TC also sets this bit to 0.
1:0	**DMA Channel Select:** Bits[1:0] select the DMA channel mode Register to program with bit 2. **Bits[1:0] Channel** 00 Channel 0 01 Channel 1 (5) 10 Channel 2 (6) 11 Channel 3 (7)

2.4.1.4 Mask Register—Write Single Mask Bit

I/O Address: Channels 0–3—0Ah; Channels 4–7—0D4h
Default Value: Bits[1:0] = undefined; Bit 2 = 1; Bits[7:3] = 0 (CPURST or a Master Clear)
Attribute: Write Only

A channel's mask bit is automatically set when the Current Byte/Word Count Register reaches terminal count (unless the channel is programmed for autoinitialization). Setting the entire register disables all DMA requests until a clear mask register instruction allows them to occur. This instruction format is similar to the format used with the DMA Request Register. Masking DMA channel 4 (DMA controller 2, channel 0) also masks DMA channels [3:0].

Bit	Description
7:3	**Reserved**: Must be 0.
2	**Channel Mask Select:** 1 = Disable DREQ for the selected channel. 0 = Enable DREQ for the selected channel.
1:0	**DMA Channel Select:** Bits[1:0] select the DMA Channel Mode Register for bit 2. **Bits[1:0]** **Channel** 00 Channel 0 (4) 01 Channel 1 (5) 10 Channel 2 (6) 11 Channel 3 (7)

2.4.1.5 Mask Register—Write All Mask Bits

I/O Address: Channels 0–3—0Fh; Channels 4–7—0DEh
Default Value: Bit[3:0] = 1; Bit[7:4] = 0 (CPURST or Master Clear)
Attribute: Read/Write

A channel's mask bit is automatically set to 1 when the Current Byte/Word Count Register reaches terminal count (unless the channel is programmed for autoinitialization). Setting bits[3:0] to 1 disables all DMA requests until a clear mask register instruction enables the requests. Note that, masking DMA channel 4 (DMA controller 2, channel 0), masks DMA channels [3:0]. Also note that masking DMA controller 2 with a write to port 0DEh also masks DREQ assertions from DMA controller 1.

Bit	Description
7:4	**Reserved**: Must be 0.
3:0	**Channel Mask Bits:** 1 = Disable the corresponding DREQ(s); 0 = Enable the corresponding DREQ(s). **Bit** **Channel** 0 0 (4) 1 1 (5) 2 2 (6) 3 3 (7)

2.4.1.6 DS—DMA Status Register

I/O Address: Channels 0–3—08h; Channels 4–7—0D0h
Default Value: 00h
Attribute: Read Only

Each DMA controller has a read-only DMA Status Register that indicates which channels have reached terminal count and which channels have a pending DMA request.

Bit	Description
7:4	**Channel Request Status:** When a valid DMA request is pending for a channel (on its DREQ signal line), the corresponding bit is set to 1. When a DMA request is not pending for a particular channel, the corresponding bit is set to 0. The source of the DREQ may be hardware or a software request. Note that channel 4 does not have DREQ or DACK lines, so the response for a read of DMA2 status for channel 4 is irrelevant. **Bit** **Channel** 4 0 5 1 (5) 6 2 (6) 7 3 (7)
3:0	**Channel Terminal Count Status:** 1 = TC is reached; 0 = TC is not reached. **Bit** **Channel** 0 0 1 1 (5) 2 2 (6) 3 3 (7)

2.4.1.7 DMA Base and Current Address Registers (8237 Compatible Segment)

I/O Address: DMA Channel 0000h DMA Channel 40C0h
DMA Channel 1002h DMA Channel 50C4h
DMA Channel 2004h DMA Channel 60C8h
DMA Channel 3006h DMA Channel 70CCh
Default Value: XXXXh (CPURST or Master Clear)
Attribute: Read/Write

This register works in conjunction with the Low Page Register. After an autoinitialization, this register retains the original programmed value. Autoinitialize takes place after a TC. The address register is automatically incremented or decremented after each transfer. This register is read/written in successive 8-bit bytes. The programmer must issue the "Clear Byte Pointer Flip-Flop" command to reset the internal byte pointer and correctly align the write prior to programming the Current Address Register. Autoinitialize takes place only after a TC.

Bit	Description
15:0	**Base and Current Address [15:0]:** These bits represent address bits[15:0] used when forming the 24-bit address for DMA transfers.

2.4.1.8 DMA Base and Current Byte/Word Count Registers (Compatible Segment)

I/O Address:	DMA Channel 0001h	DMA Channel 40C2h
	DMA Channel 1003h	DMA Channel 50C6h
	DMA Channel 2005h	DMA Channel 60CAh
	DMA Channel 3007h	DMA Channel 70CEh
Default Value:	XXXXh (CPURST or Master Clear)	
Attribute:	Read/Write	

This register determines the number of transfers to be performed. The actual number of transfers is one more than the number programmed in the Current Byte/Word Count Register When the value in the register is decremented from zero to FFFFh, a TC is generated. Autoinitialize can only occur when a TC occurs. If it is not autoinitialized, this register has a count of FFFFh after TC.

For transfers to/from an 8-bit I/O, the Byte/Word count indicates the number of bytes to be transferred. This applies to DMA channels 0–3. For transfers to/from a 16-bit I/O, with shifted address, the Byte/Word count indicates the number of 16-bit words to be transferred. This applies to DMA channels 5–7.

Bit	Description
15:0	**Base and Current Byte/ Word Count:** These bits represent the 16 byte/word count bits used when counting down a DMA transfer.

2.4.1.9 DMA Memory Low Page Registers

I/O Address:	DMA Channel 0087h	DMA Channel 508Bh
	DMA Channel 1083h	DMA Channel 6089h
	DMA Channel 2081h	DMA Channel 708Ah
	DMA Channel 3082h	
Default Value:	XXh (CPURST or Master Clear)	
Attribute:	Read/Write	

This register works in conjunction with the Current Address Register. After an autoinitialization, this register retains the original programmed value. Autoinitialize takes place after a TC.

Bit	Description
7:0	**DMA Low Page [23:16]:** These bits represent address bits[23:16] of the 24-bit DMA address.

2.4.1.10 DMA Clear Byte Pointer Register

I/O Address: Channels 0–3—00Ch; Channels 4–7—0D8h
Default Value: All bits undefined
Attribute: Write Only

Writing to this register executes the Clear Byte Pointer Command. This command is executed prior to reading/writing a new address or word count to the DMA. The command initializes the byte pointer flip-flop to a known state so that subsequent accesses to register contents address upper and lower bytes in the correct sequence. The Clear Byte Pointer Command (or CPURST or the Master Clear Command) clears the internal latch used to address the upper or lower byte of the 16-bit Address and Word Count Registers.

Bit	Description
7:0	**Clear Byte Pointer:** No specific pattern. Command enabled with a write to the I/O port address.

2.4.1.11 DMC—DMA Master Clear Register

I/O Address: Channel 0–3—00Dh; Channel 4–7—0DAh
Default Value: All bits undefined
Attribute: Write Only

This software instruction has the same effect as the hardware reset.

Bit	Description
7:0	**Master Clear:** No specific pattern. Command enabled with a write to the I/O port address.

2.4.1.12 DCLM—DMA Clear Mask Register

I/O Address: Channel 0–3—00Eh; Channel 4–7—0DCh
Default Value: All bits undefined
Attribute: Write Only

Bit	Description
7:0	**Clear Mask Register:** No specific pattern. Command enabled with a write to the I/O port address.

2.4.2 TIMER/COUNTER REGISTER DESCRIPTION

2.4.2.1 TCW—Timer Counter Control Word Register

I/O Address: 043h
Default Value: All bits undefined
Attribute: Write Only

The Timer Control Word Register specifies the counter selection, the operating mode, the counter byte programming order and size of the count value, and whether the counter counts down in a 16-bit or binary-coded decimal (BCD) format. After writing the control word, a new count can be written at any time. The new value takes effect according to the programmed mode.

Bit	Description
7:6	**Counter Select:** The Read Back Command is selected when bits[7:6] are both 1. **Bit[7:6]** — **Function** 00 — Counter 0 01 — Counter 1 10 — Counter 2 11 — Read Back Command
5:4	**Read/Write Select:** The Counter Latch Command is selected when bits[5:4] are both 0. **Bit[5:4]** — **Function** 00 — Counter Latch Command 01 — R/W Least Significant Byte 10 — R/W Most Significant Byte 11 — R/W LSB then MSB
3:1	**Counter Mode Selection:** Bits[3:1] select one of six possible counter modes. **Bit[3:1]** — **Mode** — **Function** 000 — 0 — Out signal on end of count (= 0) 001 — 1 — Hardware retriggerable one-shot X10 — 2 — Rate generator (divide by n counter) X11 — 3 — Square wave output 100 — 4 — Software triggered strobe 101 — 5 — Hardware triggered strobe
0	**Binary/BCD Countdown Select:** 0 = Binary countdown. The largest possible binary count is 2^{16}. 1 = Binary coded decimal (BCD) count is used. The largest BCD count allowed is 10^4.

Read Back Command

The Read Back Command is used to determine the count value, programmed mode, and current states of the OUT pin and Null count flag of the selected counter or counters. The Read Back Command is written to the Timer Control Word Register which latches the current states of the above mentioned variables. The value of the counter and its status may then be read by I/O access to the counter address. Note that the Timer Counter Register bit definitions are different during the Read Back Command than for a normal Timer Counter Register write.

Bit	Description
7:6	**Read Back Command:** When bits[7:6] = 11, the Read Back Command is selected during a write to the Timer Control Word Register. Following the Read Back Command, I/O reads from the selected counter's I/O addresses produce the current latch status, the current latched count, or both if bits 4 and 5 are both 0.
5	**Latch Count of Selected Counters:** When bit 5 = 0, the current count value of the selected counters will be latched. When bit 5 = 1, the count will not be latched.
4	**Latch Status of Selected Counters:** When bit 4 = 0, the status of the selected counters will be latched. When bit 4 = 1, the status will not be latched. The status byte format is described in Section 4.3.3, Interval Timer Status Byte Format Register.
3	**Counter 2 Select:** When bit 3 = 1, Counter 2 is selected for the latch command selected with bits 4 and 5. When bit 3 = 0, status and/or count will not be latched.
2	**Counter 1 Select:** When bit 2 = 1, Counter 1 is selected for the latch command selected with bits 4 and 5. When bit 2 = 0, status and/or count will not be latched.
1	**Counter 0 Select:** When bit 1 = 1, Counter 0 is selected for the latch command selected with bits 4 and 5. When bit 1 = 0, status and/or count will not be latched.
0	**Reserved**: Must be 0.

Counter Latch Command

The Counter Latch Command latches the current count value at the time the command is received. If a Counter is latched once and then, some time later, latched again before the count is read, the second Counter Latch Command is ignored. The count read will be the count at the time the first Counter Latch Command was issued. If the counter is programmed for two byte counts, two bytes must be read. The two bytes do not have to be read successively (read, write, or programming operations for other counters may be inserted between the reads). Note that the Timer Counter Register bit definitions are different during the Counter Latch Command than for a normal Timer Counter Register write. Note that if a counter is programmed to read/write two-byte counts, a program must not transfer control between reading the first and second byte to another routine that also reads from that same counter. Otherwise, an incorrect count will be read.

Bit	Description
7:6	**Counter Selection:** Bits 6 and 7 are used to select the counter for latching. **Bit[7:6] Function** 00 latch counter 0 01 latch counter 1 10 latch counter 2 11 Read Back Command
5:4	**Counter Latch Command:** When bits[5:4] = 00, the Counter Latch Command is selected during a write to the Timer Control Word Register. Following the Counter Latch Command, I/O reads from the selected counter's I/O addresses produce the current latched count.
3:0	**Reserved**: Must be 0.

2.4.2.2 Interval Timer Status Byte Format Register

I/O Address: Counter 0—040h; Counter 1—041h; Counter 2—042h
Default Value: Bits[6:0] = X; Bit 7 = 0
Attribute: Read Only

Each counter's status byte can be read following an Interval Timer Read Back Command. If latch status is chosen (bit 4 = 0, Read Back Command) as a read back option for a given counter, the next read from the counter's Counter Access Ports Register returns the status byte.

Bit	Description
7	**Counter OUT Pin State:** 1 = Pin is 1; 0 = Pin is 0.
6	**Count Register Status:** This bit indicates when the last count written to the Count Register (CR) has been loaded into the counting element (CE). 0 = Count has been transferred from CR to CE and is available for reading. 1 = Count has not been transferred from CR to CE and is not yet available for reading.
5:4	**Read/Write Selection Status:** Bits[5:4] reflect the read/write selection made through bits[5:4] of the Control Register. **Bit[5:4] Function** 00 Counter Latch Command 01 R/W Least Significant Byte (LSB) 10 R/W Most Significant Byte (MSB) 11 R/W LSB then MSB
3:1	**Mode Selection Status:** Bits[3:1] return the counter mode programming. **Bit[3:1] Mode Selected** — **Bit[3:1] Mode Selected** 000 0 — X11 3 001 1 — 100 4 X10 2 — 101 5
0	**Countdown Type Status:** 0 = Binary countdown; 1 = Binary coded decimal (BCD) countdown.

2.4.2.3 Counter Access Ports Register

I/O Address: Counter 0—040h; Counter 1—041h; Counter 2—042h
Default Value: All bits undefined
Attribute: Read/Write

Each of these I/O ports is used for writing count values to the Count Registers; reading the current count value from the counter by either an I/O read, after a counter-latch command, or after a Read Back Command; and reading the status byte following a Read Back Command.

Bit	Description
7:0	**Counter Port bit[x]**: Each counter I/O port address is used to program the 16-bit Count Register. The order of programming, either LSB only, MSB only, or LSB then MSB, is defined with the Interval Counter Control Register. The counter I/O port is also used to read the current count from the Count Register and return counter programming status following a Read Back Command.

2.4.3 INTERRUPT CONTROLLER REGISTERS

The PIIX contains an ISA-Compatible interrupt controller that incorporates the functionality of two 82C59 interrupt controllers. The interrupt registers control the operation of the interrupt controller.

2.4.3.1 ICW1—Initialization Command Word 1 Register

I/O Address: INT CNTRL-1—020h; INT CNTRL-2—0A0h
Default Value: All bits undefined
Attribute: Write Only

A write to Initialization Command Word 1 starts the interrupt controller initialization sequence. Addresses 020h and 0A0h are referred to as the base addresses of CNTRL-1 and CNTRL-2, respectively. An I/O write to the CNTRL-1 or CNTRL-2 base address with bit 4 equal to 1 is interpreted as ICW1. For PIIX-based ISA systems, three I/O writes to "base address + 1" must follow the ICW1. The first write to "base address + 1" performs ICW2, the second write performs ICW3, and the third write performs ICW4.

ICW1 starts the initialization sequence during which the following automatically occur:

1. The Interrupt Mask Register is cleared.
2. IRQ7 input is assigned priority 7.
3. The slave mode address is set to 7.
4. Special Mask Mode is cleared and Status Read is set to IRR.
5. If IC4 was set to 0, then all functions selected by ICW4 are set to 0. However, ICW4 must be programmed in the PIIX implementation of this interrupt controller, and IC4 must be set to 1.

Bit	Description
7:5	**ICW/OCW Select:** These bits should be 000 when programming the PIIX.
4	**ICW/OCW Select:** Bit 4 must be 1 to select ICW1. After the fixed initialization sequence to ICW1, ICW2, ICW3, and ICW4, the controller base address is used to write to OCW2 and OCW3. Bit 4 is 0 on writes to these registers. A 1 on this bit at any time will force the interrupt controller to interpret the write as an ICW1. The controller will then expect to see ICW2, ICW3, and ICW4.
3	**Edge/Level Bank Select (LTIM):** This bit is disabled. Its function is replaced by the Edge/Level Triggered Control (ELCR) Registers.
2	**ADI:** Ignored for the PIIX.
1	**Single or Cascade (SNGL):** This bit must be programmed to 0.
0	**ICW4 Write Required (IC4):** This bit must be set to 1.

2.4.3.2 ICW2—Initialization Command Word 2 Register

I/O Address: INT CNTRL-1—021h; INT CNTRL-2—0A1h
Default Value: All bits undefined
Attribute: Write Only

ICW2 is used to initialize the interrupt controller with the five most significant bits of the interrupt vector address.

Bit	Description
7:3	**Interrupt Vector Base Address:** Bits[7:3] define the base address in the interrupt vector table for the interrupt routines associated with each interrupt request level input.
2:0	**Interrupt Request Level:** Must be programmed to all 0s.

2.4.3.3 ICW3—Initialization Command Word 3 Register (Master Controller)

I/O Address: INT CNTRL-1—021h
Default Value: All bits undefined
Attribute: Write Only

The meaning of ICW3 differs between CNTRL-1 and CNTRL-2. On CNTRL-1, the master controller, ICW3 indicates which CNTRL-1 IRQ line physically connects the INTR output of CNTRL-2 to CNTRL-1.

Bit	Description
7:3	**Reserved:** Must be programmed to all 0s.
2	**Cascaded Mode Enable:** This bit must be programmed to 1 selecting cascade mode.
1:0	**Reserved:** Must be programmed to all 0s.

2.4.3.4 ICW3—Initialization Command Word 3 Register (Slave Controller)

I/O Address: INT CNTRL-2—0A1h
Default Value: All bits undefined
Attribute: Write Only

On CNTRL-2, the slave controller, ICW3 is the slave identification code broadcast by CNTRL-1.

Bit	Description
7:3	**Reserved:** Must be programmed to all 0s.
2:0	**Slave Identification Code:** Must be programmed to 010b.

2.4.3.5 ICW4—Initialization Command Word 4 Register

I/O Address: INT CNTRL-1—021h; INT CNTRL-2—0A1h
Default Value: 01h
Attribute: Write Only

Both PIIX interrupt controllers must have ICW4 programmed as part of their initialization sequence.

Bit	Description
7:5	**Reserved:** Must be programmed to all 0s.
4	**Special Fully Nested Mode (SFNM):** Bit 4, SFNM, should normally be disabled by writing 0 to this bit. If SFNM = 1, the special fully nested mode is programmed.
3	**Buffered Mode (BUF):** Must be programmed to 0 selecting non-buffered mode.
2	**Master/Slave in Buffered Mode:** Should always be programmed to 0. Bit not used.
1	**AEOI (Automatic End of Interrupt):** This bit should normally be programmed to 0. This is the normal end of interrupt. If this bit is 1, the automatic end of interrupt mode is programmed.
0	**Microprocessor Mode:** Must be programmed to 1 indicating an Intel Architecture-based system.

2.4.3.6 OCW1—Operational Control Word 1 Register

I/O Address: INT CNTRL-1—021h; INT CNTRL-2—0A1h
Default Value: 00h
Attribute: Read/Write

OCW1 sets and clears the mask bits in the Interrupt Mask Register (IMR). Each interrupt request line may be selectively masked or unmasked any time after initialization. The IMR stores the interrupt line mask bits. The IMR operates on the IRR. Masking of a higher priority input does not affect the interrupt request lines of lower priority. Unlike status reads of the ISR and IRR, for reading the IMR, no OCW3 is needed. The output data bus contains the IMR when an I/O read is active and the I/O address is 021h or 0A1h (OCW1). All writes to OCW1 must occur following the ICW1-ICW4 initialization sequence, since the same I/O ports are used for OCW1, ICW2, ICW3 and ICW4.

Bit	Description
7:0	**Interrupt Request Mask (Mask [7:0]):** When 1 is written to any bit in this register, the corresponding IRQx line is masked. For example, if bit 4 is set to 1, then IRQ4 is masked. Interrupt requests on IRQ4 do not set channel 4's Interrupt Request Register (IRR) bit as long as the channel is masked. When 0 is written to any bit in this register, the corresponding IRQx is unmasked. Note that masking IRQ2 on CNTRL-1 also masks the interrupt requests from CNTRL-2, which is physically cascaded to IRQ2.

2.4.3.7 OCW2—Operational Control Word 2 Register

I/O Address: INT CNTRL-1—020h; INT CNTRL-2—0A0h
Default Value: Bits[4:0] = undefined; Bits[7:5] = 001
Attribute: Write Only

OCW2 controls both the Rotate Mode and the End of Interrupt Mode. Following a CPURST or ICW initialization, the controller enters the fully nested mode of operation. Both rotation mode and specific EOI mode are disabled following initialization.

Bit	Description
7:5	**Rotate and EOI Codes:** R, SL, EOI - These three bits control the Rotate and End of Interrupt modes and combinations of the two. A chart of these combinations is listed above under the bit definition. **Bits[7:5] Function** — **Bits[7:5] Function** 001 Non-specific EOI Cmd — 000 Rotate in Auto EOI Mode (Clear) 011 Specific EOI Cmd — 111 *Rotate on Specific EOI Cmd 101 Rotate on Non-Specific EOI Cmd — 110 *Set Priority Cmd 100 Rotate in Auto EOI Mode (Set) — 010 No Operation * L0 - L2 Are Used
4:3	**OCW2 Select:** Must be programmed to 00 selecting OCW2.
2:0	**Interrupt Level Select (L2, L1, L0):** L2, L1, and L0 determine the interrupt level acted upon when the SL bit is active (bit 6). When the SL bit is inactive, bits[2:0] do not have a defined function; programming L2, L1 and L0 to 0 is sufficient in this case. **Bits[2:0] Interrupt Level** — **Bits[2:0] Interrupt Level** 000 IRQ 0(8) — 100 IRQ 4(12) 001 IRQ 1(9) — 101 IRQ 5(13) 010 IRQ 2(10) — 110 IRQ 6(14) 011 IRQ 3(11) — 111 IRQ 7(15)

2.4.3.8 OCW3—Operational Control Word 3 Register

I/O Address: INT CNTRL-1—020h; INT CNTRL-2—0A0h
Default Value: Bits[6,0] = 0; Bits[7,4:2] = Undefined; Bits[5,1] = 1
Attribute: Read/Write

OCW3 serves three important functionsEnable Special Mask Mode, Poll Mode control, and IRR/ISR Register read control.

Bit	Description
7	**Reserved:** Must be 0.
6	**Special Mask Mode (SMM):** If ESMM = 1 and SMM = 1, the interrupt controller enters Special Mask Mode. If ESMM = 1 and SMM = 0, the interrupt controller is in normal mask mode. When ESMM = 0, SMM has no effect.
5	**Enable Special Mask Mode (ESMM):** 1 = Enable SMM bit; 0 = Disable SMM bit.
4:3	**OCW3 Select:** Must be programmed to 01 selecting OCW3.

Bit	Description
2	**Poll Mode Command:** 0 = Disable Poll Mode Command. When bit 2 = 1, the next I/O read to the interrupt controller is treated as an interrupt acknowledge cycle indicating highest priority request.
1:0	**Register Read Command:** Bits[1:0] provide control for reading the In-Service Register (ISR) and the Interrupt Request Register (IRR). When bit 1 = 0, bit 0 doesl not affect the register read selection. When bit 1 = 1, bit 0 selects the register status returned following an OCW3 read. If bit 0 = 0, the IRR will be read. If bit 0 = 1, the ISR will be read. Following ICW initialization, the default OCW3 port address read will be "read IRR". To retain the current selection (read ISR or read IRR), always write 0 to bit 1 when programming this register. The selected register can be read repeatedly without reprogramming OCW3. To select a new status register, OCW3 must be reprogrammed prior to attempting the read. **Bits[1:0] Function** 00 No Action 01 No Action 10 Read IRQ Register 11 Read IS Register

2.4.3.9 ELCR1—Edge/Level Triggered Register

I/O Address: INT CNTRL-1—4D0h
Default Value: 00h
Attribute: Read/Write

ELCR1 Register allows IRQ3–IRQ7 to be edge or level programmable on an interrupt-by-interrupt basis. IRQ0, IRQ1 and IRQ2 are not programmable and are always edge sensitive.

Bit	Description
7	**IRQ7 ECL:** 0 = edge triggered mode; 1 = level sensitive mode.
6	**IRQ6 ECL:** 0 = edge triggered mode; 1 = level sensitive mode.
5	**IRQ5 ECL:** 0 = edge triggered mode; 1 = level sensitive mode.
4	**IRQ4 ECL:** 0 = edge triggered mode; 1 = level sensitive mode.
3	**IRQ3 ECL:** 0 = edge triggered mode; 1 = level sensitive mode.
2:0	**Reserved:** Must be 0.

2.4.3.10 ELCR2—Edge/Level Triggered Register

I/O Address: INT CNTRL-2—4D1h
Default Value: 00h
Attribute: Read/Write

ELCR2 Register allows IRQ[15,14,12:9] to be edge or level programmable on an interrupt by interrupt basis. Note that IRQ[13,8#] are not programmable and are always edge sensitive.

Bit	Description
7	**IRQ15 ECL:** 0 = edge triggered mode; 1 = level sensitive mode.
6	**IRQ14 ECL:** 0 = edge triggered mode; 1 = level sensitive mode.
5	**Reserved:** Must be 0.
4	**IRQ12 ECL:** 0 = edge triggered mode; 1 = level sensitive mode.
3	**IRQ11 ECL:** 0 = edge triggered mode; 1 = level sensitive mode.
2	**IRQ10 ECL:** 0 = edge triggered mode; 1 = level sensitive mode.
1	**IRQ9 ECL:** 0 = edge triggered mode; 1 = level sensitive mode.
0	**Reserved:** Must be 0.

2.4.4 X-BUS, COPROCESSOR, AND RESET REGISTERS

2.4.4.1 Reset X-Bus IRQ12 and IRQ1 Register

I/O Address: 60h
Default Value: N/A
Attribute: Read Only

This register clears the mouse interrupt function and the keyboard interrupt (IRQ1). Reads to this address are monitored by the PIIX. When the mouse interrupt function is enabled (X-Bus Chip Select Register), the mouse interrupt function is provided on the IRQ12/M input signal. In this mode, a mouse interrupt generates an interrupt through IRQ12 to the Host CPU. A read of 60h releases IRQ12. Reads/writes flow through to the ISA Bus.

Bit	Description
7:0	**Reset IRQ12 and IRQ1:** No specific pattern. A read of address 60h executes the command.

2.4.4.2 Coprocessor Error Register

I/O Address: F0h
Default Value: N/A
Attribute: Write Only

Writing to this register causes the PIIX to assert IGNNE#. The PIIX also negates IRQ13 (internal to the PIIX). Note that IGNNE# is not asserted unless FERR# is active. Reads/writes flow through to the ISA Bus.

Bit	Description
7:0	No special pattern required: A write to address F0h executes the command.

2.4.4.3 RC—Reset Control Register

I/O Address: CF9h
Default Value: 00h
Attribute: Read/Write

Bits 1 and 2 in this register are used by the PIIX to generate a hard reset or a soft reset.

Bit	Description
7:4	**Reserved.**
3	**Reserved.**
2	**Reset CPU (RCPU):** This bit is used to initiate a hard or soft reset to the CPU. To perform a reset, this bit should be set to 0 and then set to 1. This "0" to "1" transition initiates the reset. During a hard reset, the PIIX asserts CPURST, PCIRST#, and RSTDRV. The PIIX initiates a hard reset when this register is programmed for a hard reset or PWROK transitions from low to high. This bit cannot be read as 1.
1	**System Reset (SRST):** This bit is used in conjunction with bit 2 in this register to initiate a hard reset. When SRST = 1, the PIIX initiates a hard reset to the CPU when bit 2 in this register transitions from 0 to 1. When SRST = 0, the PIIX initiates a soft reset when bit 2 in this register transitions from 0 to 1.
0	**Reserved.**

2.4.5 NMI REGISTERS

The NMI logic incorporates two different 8-bit registers. The CPU reads the NMISC Register to determine the NMI source (bits set to 1). After the NMI interrupt routine processes the interrupt, software clears the NMI status bits by setting the corresponding enable/disable bit to 1. The NMI Enable and Real-Time Clock Register can mask the NMI signal and disable/enable all NMI sources.

To ensure that all NMI requests are serviced, the NMI service routine software flow should be as follows:

1. NMI is detected by the processor on the rising edge of the NMI input.
2. The processor will read the status stored in port 061h to determine what sources caused the NMI. The processor may then set to 0 the register bits controlling the sources that it has determined to be active. Between the time the processor reads the NMI sources and sets them to 0, an NMI may have been generated by another source. The level of NMI will then remain active. This new NMI source will not be recognized by the processor because there was no edge on NMI.
3. The processor must then disable all NMIs by setting bit 7 of port 070H to 1 and then enable all NMIs by setting bit 7 of port 070H to 0. This will cause the NMI output to transition low then high if there are any pending NMI sources. The CPU's NMI input logic will then register a new NMI.

2.4.5.1 NMISC—NMI Status Control Register

I/O Address: 061h
Default Value: 00h
Attribute: Read/Write, Read Only

This register reports the status of different system components, control the output of the speaker counter (Counter 2), and gate the counter output that drives the SPKR signal.

Bit	Description
7	**SERR# NMI Source Status—RO:** Bit 7 is set if a system board agent (PCI devices or main memory) detects a system board error and pulses the PCI SERR# line. This interrupt source is enabled by setting bit 2 to 0. To reset the interrupt, set bit 2 to 0 and then set it to 1. When writing to port 061h, bit 7 must be 0.
6	**IOCHK# NMI Source Status—RO:** Bit 6 is set if an expansion board asserts IOCHK# on the ISA Bus. This interrupt source is enabled by setting bit 3 to 0. To reset the interrupt, set bit 3 to 0 and then set it to 1. When writing to port 061h, bit 6 must be 0.
5	**Timer Counter 2 OUT Status—RO:** The Counter 2 OUT signal state is reflected in bit 5. The value on this bit following a read is the current state of the Counter 2 OUT signal. Counter 2 must be programmed following a CPURST for this bit to have a determinate value. When writing to port 061h, bit 5 must be 0.
4	**Refresh Cycle Toggle—RO:** The Refresh Cycle Toggle signal toggles from either 0 to 1 or 1 to 0 following every refresh cycle. When writing to port 061h, bit 4 must be 0.
3	**IOCHK# NMI Enable—R/W:** 1 = Clear and disable; 0 = Enable IOCHK# NMIs.
2	**PCI SERR# Enable—R/W:** 1 = Clear and Disable; 0 = Enable.
1	**Speaker Data Enable—R/W:** 0 = SPKR output is 0; 1 = the SPKR output is the Counter 2 OUT signal value.
0	**Timer Counter 2 Enable—R/W:** 0 = Disable; 1 = Enable.

2.4.5.2 NMI Enable and Real-Time Clock Address Register

I/O Address: 070h
Default Value: Bit[6:0] = undefined; Bit 7 = 1
Attribute: Write Only

This port is shared with the real-time clock. Do not modify the contents of this register without considering the effects on the state of the other bits. Reads and writes to this register address flow through to the ISA Bus.

Bit	Description
7	**NMI Enable:** 1 = Disable; 0 = Enable.
6:0	**Real Time Clock Address:** Used by the Real Time Clock on the Base I/O component to address memory locations. Not used for NMI enabling/disabling.

2.5 System Power Management Registers

This section describes two power management registers—APMS and APMC Registers. These registers are located in normal I/O space and must be accessed (via the PCI Bus) with 8 bit accesses.

2.5.1 APMC—ADVANCED POWER MANAGEMENT CONTROL PORT

I/O Address: 0B2h
Default Value: 00h
Attribute: Read/Write

This register passes data (APM Commands) between the OS and the SMI handler. In addition, writes can generate an SMI and reads can cause STPCLK# to be asserted. The PIIX operation is not affected by the data in this register.

Bit	Description
7:0	**APM Control Port (APMC):** Writes to this register store data in the APMC Register and reads return the last data written. In addition, writes generate an SMI, if bit 7 of the SMIEN Register and bit 0 of the SMICNTL Register are both is set to 1. Reads cause the STPCLK# signal to be asserted, if bit 1 of the SMICNTL Register is set to 1. Reads do not generate an SMI.

2.5.2 APMS—ADVANCED POWER MANAGEMENT STATUS PORT

I/O Address: 0B3h
Default Value: 00h
Attribute: Read/Write

This register passes status information between the OS and the SMI handler. The PIIX operation is not affected by the data in this register.

Bit	Description
7:0	**APM Status Port (APMS):** Writes store data in this register and reads return the last data written.

2.6 PCI Bus Master IDE Registers

The PCI Bus Master IDE function uses 16 bytes of I/O space, allocated via the BMIBA Register (A PCI Base Address Register). All Bus Master IDE I/O Space Registers can be accessed as byte, word, or Dword quantities. The description of the 16 bytes of I/O registers follows:

2.6.1 BMICOM—BUS MASTER IDE COMMAND REGISTER

Address Offset: Primary Channel—Base + 00h; Secondary Channel—Base + 08h

Default Value: 00h

Attribute: Read / Write

Bit	Description
7:4	**Reserved**
3	**Bus Master Read/Write Control (RWCON):** 0 = Reads; 1 = Writes. This bit must NOT be changed when the Bus Master function is active.
2:1	**Reserved**
0	**Start/Stop Bus Master (SSBM):** 1 = Start; 0 = Stop. When this bit is set to 1, Bus Master operation starts. The controller transfers data between the IDE device and memory only when this bit is set. Master operation can be stopped by writing 0 to this bit. This results in all state information being lost (i.e., master mode operation cannot be stopped and then resumed). If this bit is set to 0 while Bus Master operation is still active (i.e., Bit 0 = 1 in the Bus Master IDE Status Register for that IDE channel) and the drive has not yet finished its data transfer (bit 2 = 0 in the channel's Bus Master IDE Status Register), the Bus Master command is aborted and data transferred from the drive may be discarded before being written to system memory. This bit is intended to be set to 0 after the data transfer is completed, as indicated by either bit 0 or bit 2 being set in the IDE Channel's Bus Master IDE Status Register.

2.6.2 BMISTA—BUS MASTER IDE STATUS REGISTER

Address Offset: Primary Channel—Base + 02h; Secondary Channel—Base + 0Ah

Default Value: 00h

Attribute: Read/Write Clear

This register provides status information about the IDE device and state of the IDE DMA transfer. Table 6 describes IDE Interrupt Status and Bus Master IDE Active bit states after a DMA transfer has been started.

Bit	Description
7	**Reserved:** This bit is hardwired to 0.
6	**Drive 1 DMA Capable (DMA1CAP)—R/W:** 1 = Drive 1 is capable of DMA transfers. This bit is a software controlled status bit that indicates IDE DMA device capability and does not affect hardware operation.
5	**Drive 0 DMA Capable (DMA0CAP)—R/W:** 1 = Drive 0 is capable of DMA transfers. This bit is a software controlled status bit that indicates IDE DMA device capability and does not affect hardware operation.
4:3	**Reserved.**

Bit	Description
2	**IDE Interrupt Status—R/WC:** This bit, when set to 1, indicates when an IDE device has asserted its interrupt line. When bit 2 = 1, all read data from the IDE device has been transferred to main memory and all write data has been transferred to the IDE device. Software sets this bit to 0 by writing 1 to it. **The IRQ14 signal (pin 83) must be used for the primary channel and MIRQ0 is used for the secondary channel.** If the interrupt status bit is set to 0 by writing 1 to this bit while the interrupt line (IRQ14 or MIRQ0) is still at the active level, this bit remains 0 until another assertion edge is detected on the interrupt line.
1	**IDE DMA Error—R/WC:** This bit is set to 1 when the PIIX encounters a target abort or master abort while transferring data on the PCI Bus. Software sets this bit to 0 by writing 1 to it.
0	**Bus Master IDE Active (BMIDEA)—RO:** The PIIX sets this bit to 1 when bit 0 in the BMICOM Register is set to 1. The PIIX sets this bit to 0 when the last transfer for a region is performed (where EOT for that region is set in the region descriptor). The PIIX also sets this bit to 0 when bit 0 of the BMICOM Register is set to 0. When this bit is read as 0, all data transferred from the drive during the previous Bus Master command is visible in system memory, unless the Bus Master command was aborted.

Table 6. Interrupt/Activity Status Combinations

Bit 2	Bit 0	Description:
0	1	DMA transfer is in progress. No interrupt has been generated by the IDE device.
1	0	The IDE device generated an interrupt and the Physical Region Descriptors exhausted. This is normal completion where the size of the physical memory regions is equal to the IDE device transfer size.
1	1	The IDE device generated an interrupt. The controller has not reached the end of the physical memory regions. This is a valid completion case when the size of the physical memory regions is larger than the IDE device transfer size.
0	0	Error condition. If the IDE DMA Error bit is 1, there is a problem transferring data to/from memory. Specifics of the error have to be determined using bus-specific information. If the Error bit is 0, the PRD's specified a smaller size than the IDE transfer size.

2.6.3 BMIDTP—BUS MASTER IDE DESCRIPTOR TABLE POINTER REGISTER

Address Offset: Primary Channel—Base + 04h; Secondary Channel—Base + 0Ch
Default Value: 00000000h
Attribute: Read/Write

This register provides the base memory address of the descriptor table. The Descriptor Table must be Dword aligned and not cross a 4-Kbyte boundary in memory.

Bit	Description
31:2	**Descriptor Table Base Address:** Bits[31:2] correspond to A[31:2].
1:0	**Reserved.**

3.0 FUNCTIONAL DESCRIPTION

This section describes each of the major functions on the PIIX including the memory and I/O address map, DMA controller, interrupt controller, timer/counter, and power management. The PCI, ISA, X-Bus, and IDE interfaces.

3.1 Memory and I/O Address Map

The PIIX interfaces to two system buses—PCI and ISA Buses. The PIIX provides positive decode for certain I/O and memory space accesses on these buses as described in this section. ISA masters and DMA devices have access to PCI memory and some of the internal PIIX registers as described in the Register Description section. ISA masters and DMA devices do not have access to host or PCI I/O space.

3.1.1 I/O ACCESSES

The PIIX positively decodes accesses to the PCI Configuration Registers (PCI only), power management registers (PCI only), and Bus Master IDE interface registers (PCI only). The PIIX also positively decodes the ISA-Compatible Registers (PCI and ISA), except for the DMA Register I/O space which is subtractively decoded. For details concerning accessing these registers, see Register Description section.

The PIIX also provides positive decode for BIOS, X-Bus, and system event decode for SMM support. In addition, the PIIX positively decodes PCI Bus accesses to registers located on the IDE device, when enabled. For IDE port accesses, see PCI Local Bus IDE section.

3.1.2 MEMORY ADDRESS MAP

For PCI accesses to ISA memory, accesses below 16 Mbyte (including BIOS space) that are not claimed by a PCI device (subtractive decode) are forwarded to ISA. For write accesses that are not claimed by an ISA slave, the cycle completes normally (i.e. 8-bit, 6 SYSCLK cycle). For read accesses that are not claimed by an ISA slave, the PIIX returns data corresponding to the state of the ISA Bus and completes the cycle normally (i.e. 8-bit, 6 SYSCLK cycle).

For ISA/DMA accesses to main memory, all accesses to memory locations 0–512 Kbytes (512–640 Kbytes, if enabled), or accesses above 1 Mbyte and below the top of memory are forwarded to the PCI Bus (Table 7). The Top of Memory is equal to the value programmed in the Top of Memory Register (bits [7:3]). All remaining ISA originated accesses are confined to the ISA Bus.

Table 7. DMA and ISA Master Accesses to Main Memory

Memory Space	Response
Top of main memory to 128 Mbytes	Confine to ISA
1 Mbyte to top of main memory	Forward to main memory[1]
1 Mbyte minus 128 Kbytes to 1 Mbyte minus 64 Kbytes	Confine to ISA[2]
640 Kbytes to 1 Mbyte minus 128 Kbytes	Confine to ISA
512–640 Kbytes	Confine to ISA[3]
0–512 Kbytes	Forward to PCI

NOTES:
1. Except accesses to programmed memory hole.
2. Forward to main memory if bit 6 = 0 in the XBCS Register and bit 3 = 1 in the TOM Register.
3. Forward to main memory if bit 1 = 0 in the TOM Register.

3.1.3 BIOS MEMORY

The PIIX supports 512 Kbytes of BIOS space. This includes the normal 128-Kbyte space plus an additional 384-Kbyte BIOS space (known as the extended BIOS area). The XBCS Register provides BIOS space access control. Access to the lower 64-Kbyte block of the 128-Kbyte space and the extended BIOS space can be individually enabled/disabled. In addition, write protection can be programmed for the entire BIOS space.

PCI Access to BIOS Memory

The 128-Kbyte BIOS memory space is located at 000E0000–000FFFFFh (top of 1 Mbyte) and is aliased at FFFE0000h (top of 4 Gbytes). This 128-Kbyte block is split into two 64-Kbyte blocks. Accesses to the top 64 Kbytes (000F0000–000FFFFFh) are forwarded to the ISA Bus and BIOSCS# is always generated. Accesses to the bottom 64 Kbytes (000E0000–000EFFFFh) are forwarded to the ISA Bus and BIOSCS# is only generated when this BIOS region is enabled. If this BIOS region is enabled (bit 6 = 1 in the XBCS Register), accesses to the aliased region at the top of 4 Gbytes (FFFE0000h - FFFEFFFFh) are forwarded to ISA and BIOSCS# generated. If disabled, these accesses are not forwarded to ISA and BIOSCS# is not generated.

The additional 384-Kbyte region resides at FFF80000–FFFDFFFFh. If this BIOS region is enabled (bit 7 = 1 in the XBCS Register), these accesses (FFF80000h–FFFDFFFFh) are forwarded to ISA and BIOSCS# generated. If disabled, these accesses are not forwarded to ISA and BIOSCS# not generated.

ISA Access to BIOS Memory

The PIIX confines all ISA-initiated BIOS accesses to the top 64 Kbytes of the 128-Kbyte region (F0000–FFFFFh) to the ISA Bus, even if BIOS is shadowed in main memory. Accesses to the bottom 64 Kbytes of the 128-Kbyte BIOS region (E0000–EFFFFh) are confined to the ISA Bus, when this region is enabled. When the BIOS region is disabled, accesses are forwarded to main memory.

Accesses to the top 64-Kbyte BIOS region always generates BIOSCS#. Accesses to the bottom 64-Kbyte BIOS region generate BIOSCS#, when this region is enabled.

3.2 PCI Interface

The PIIX incorporates a fully PCI Bus-compatible master and slave interface. As a PCI master, the PIIX runs cycles on behalf of DMA, ISA masters, or a Bus Master IDE. As a PCI slave, the PIIX accepts cycles initiated by PCI masters targeted for the PIIX's internal register set or the ISA Bus. The PIIX directly supports the PCI interface running at either 25 MHz, 30 MHz, or 33 MHz.

3.2.1 PCI COMMAND SET

Bus commands indicate to the slave the type of transaction the master is requesting. Bus commands are encoded on the C/BE[3:0]# lines during the address phase of a PCI cycle.

Table 8. PCI Commands

C/BE[3:0]#	Command Type As Slave	Supported As Slave	Supported As Master
0000	Interrupt Acknowledge	Yes	No
0001	Special Cycle	Yes[4]	No
0010	I/O Read	Yes	No
0011	I/O Write	Yes	No
0100	Reserved	No[3]	No
0101	Reserved	No[3]	No
0110	Memory Read	Yes	Yes
0111	Memory Write	Yes	Yes
1000	Reserved	No[3]	No
1001	Reserved	No[3]	No
1010	Configuration Read	Yes	No
1011	Configuration Write	Yes	No
1100	Memory Read Multiple	No[2]	No
1101	Reserved	No[3]	No
1110	Memory Read Line	No[2]	No
1111	Memory Write and Invalidate	No[1]	No

NOTES:
1. Treated as Memory Write.
2. Treated as Memory Read.
3. Reserved cycles are considered invalid by the PIIX and are completely ignored. All internal address decoding is ignored and DEVSEL# is never to be asserted.
4. The PIIX responds to 1 type of special cyclea Shut Down special cycle. All other special cycles are ignored by the PIIX.

3.2.2 TRANSACTION TERMINATION

The PIIX supports the standard PCI cycle terminations as described in the PCI Local Bus specification.

PIIX as a Master—Master-Initiated Termination

The PIIX supports three forms of master-initiated termination:

1. Normal termination of a completed transaction.
2. Normal termination of an incomplete transaction due to timeout (applies to line buffer operations-IDE Bus Master).
3. Abnormal termination due to the slave not responding to the transaction (Abort).

PIIX as a Master—Response to Target-Initiated Termination

As a master, the PIIX responds in one of three ways to a target-termination—Target-Abort, Retry, or Disconnect.

PIIX as a Target—Target-Initiated Termination

The PIIX supports three forms of target-initiated termination—Disconnect, Retry, Target Abort.

3.2.3 PARITY SUPPORT

As a master, the PIIX generates address parity for read/write cycles and data parity when the PIIX is providing the data. As a slave, the PIIX generates data parity for read cycles. The PIIX does not check parity and does not generate SERR#. However, the PIIX does generate an NMI when another PCI device asserts SERR# (if enabled).

PAR is the calculated parity signal. PAR is even parity and is calculated on 36 bitsAD[31:0] signals plus C/BE[3:0]#. PAR is always calculated on 36 bits, regardless of the valid byte enables. PAR is only guaranteed to be valid one PCI clock after the corresponding address or data phase.

3.2.4 PCI ARBITRATION

The PIIX requests the use of the PCI Bus on behalf of ISA devices (Bus Masters and DMA) and IDE DMA slave devices using the PHOLD# and PHLDA# signals. These signals connect to the TSC where the PCI arbiter is located.

ISA devices (Bus Master or DMA) assert DREQ to gain access to the ISA Bus. In response, the PIIX asserts PHOLD#. The PIIX keeps DACK negated until the PIIX has ownership of the PCI Bus and Memory. The PCI arbiter asserts PHLDA# to the PIIX when the above conditions are met. The PIIX gives ownership of the ISA Bus (PCI and Memory) to the ISA device after sampling PHLDA# asserted.

1

3.3 ISA Interface

The PIIX incorporates a fully ISA Bus compatible master and slave interface. The PIIX directly drives five ISA slots without external data buffers. External transceivers are used on the SA[19:8] and SBHE# signals to permit these signals to be used with the IDE interface (Figure 1). The ISA interface also provides byte swap logic, I/O recovery support, wait state generation, and SYSCLK generation.

The ISA interface supports the following types of cycles:

- PCI master-initiated I/O and memory cycles to the ISA Bus
- DMA compatible cycles between main memory and ISA I/O and between ISA I/O and ISA memory
- Enhanced DMA cycles between PCI memory and ISA I/O (for motherboard devices only)
- ISA refresh cycles initiated by either the PIIX or an external ISA master
- ISA master-initiated memory cycles to PCI and ISA master-initiated I/O cycles to the internal PIIX registers, as shown in ISA-Compatible Register table in the Register Description section.

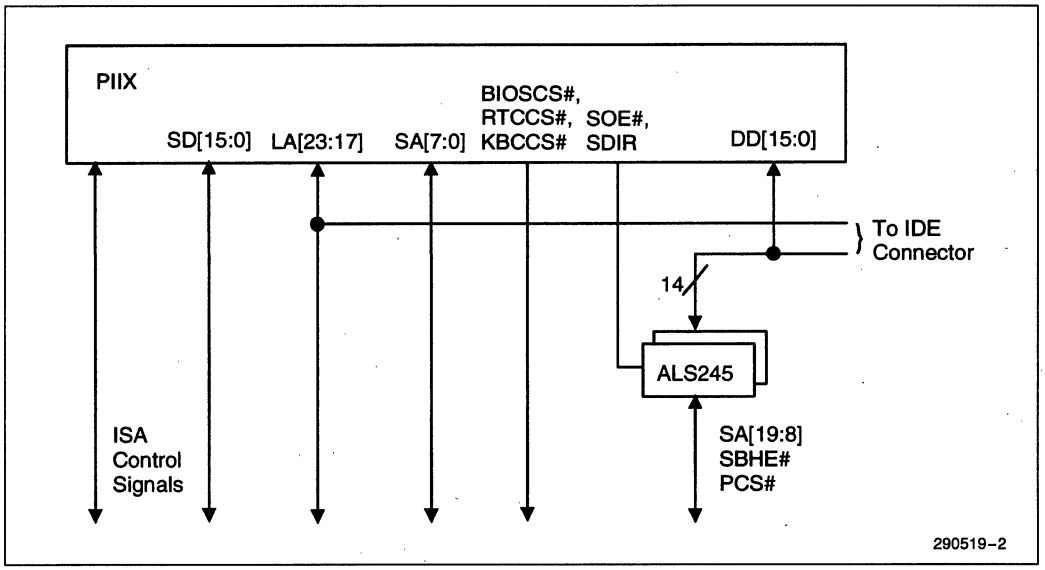


Figure 1. ISA Interface

3.4 DMA Controller

The DMA circuitry incorporates the functionality of two 82C37 DMA controllers with seven independently programmable channels (Channels 0–3 and Channels 5–7). DMA Channel 4 is used to cascade the two controllers and defaults to cascade mode in the DMA Channel Mode (DCM) Register. In addition to accepting requests from DMA slaves, the DMA controller also responds to requests that are initiated by software. Software may initiate a DMA service request by setting any bit in the DMA Channel Request Register to 1. The DMA controller for Channels 0–3 is referred to as "DMA-1" and the controller for Channels 4–7 is referred to as "DMA-2".

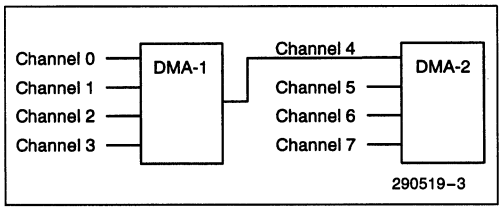


Figure 2. Internal DMA Controller

Each DMA channel is hardwired to the compatible settings for DMA device size; channels [3:0] are hardwired to 8-bit count-by-bytes transfers and channels [7:5] are hardwired to 16-bit count-by-words (address shifted) transfers. The PIIX provides the timing control and data size translation necessary for the DMA transfer between the memory (ISA or main memory) and the ISA Bus device. ISA-Compatible DMA timing is supported. Type F transfers are supported for motherboard devices and ISA add-in cards, when programmed via the MBDMAx Register.

The PIIX provides 24-bit addressing in compliance with the ISA-Compatible specification. Each channel includes a 16-bit ISA-Compatible Current Register that contains the 16 least-significant bits of the 24-bit address, an ISA Compatible Page Register that contains the eight next most significant bits of address. The DMA controller also features refresh address generation, and auto-initialization following a DMA termination.

The DMA controller is either in master or slave mode. In master mode, the DMA controller is either servicing a DMA slave's request for DMA cycles or allowing a 16-bit ISA master to use the bus (via a

cascaded DREQ signal). In slave mode, the PIIX monitors both the ISA Bus and PCI, decoding and responding to I/O read and write commands that address its registers.

Note that a DMA device (I/O device) is always on the ISA Bus, but the memory referenced is located on either an ISA Bus device or on PCI. When the PIIX is running a compatible DMA cycle, it drives the MEMR# or MEMW# strobes if the address is less than 16 Mbytes (000000–FFFFFFh). These memory strobes are generated regardless of whether the cycle is decoded for PCI or ISA memory. The SMEMR# and SMEMW# are generated if the address is less than 1 Mbytes (0000000–00FFFFFh). If the address is greater than 16 Mbytes (1000000–7FFFFFFh), the MEMR# or MEMW# strobe is not generated in order to avoid aliasing problems.

The PIIX drives the AEN signal asserted (high) during DMA cycles to prevent the I/O devices from misinterpreting the DMA cycle as a valid I/O cycle. The BALE signal is also driven high during DMA cycles. Also, during DMA memory read cycles to the PCI Bus, the PIIX return all 1s to the ISA Bus if the PCI cycle is either target aborted or master aborted.

NOTES:

1. For type F timing mode DMA transfers, the channel must be programmed with a memory range that will be forwarded to PCI. This means that if BIOS detects that ISA memory is used in the system (i.e., that the top of memory reported to the operating system is higher than the top of memory programmed in the PIIX Top of Memory Register), the BIOS should not enable type F for any channel.
2. All DMA channels support type F transfers. However, only two channels can be programmed for type F transfers at the same time.
3. In external DMA mode (selected via a strapping option on the TC signal) the PIIX tri-states the AEN, TC, and DACK[7:5, 3:0]# signals, and also forwards PCI master I/O accesses to location 0000h to ISA.

3.4.1 TYPE F TIMING

The type F DMA cycles can be used with motherboard devices and ISA add-in cards, through the use of the MDRQ[1:0] and MDAK[1:0]# signals. The type F cycles occur back to back at a minimum repetition rate of 3 SYSCLKs (360 ns minimum). The type F cycles are always performed using the 4-byte DMA buffer.

DMA Buffer for Type F Transfers

The PIIX has a 4-byte buffer that is used to reduce the PCI utilization resulting from DMA transfers by motherboard devices. The DMA buffer is always used in conjunction with the type F transfers. The type F transfers and the use of the DMA buffer are invoked in the MBDMAx Register. The 4-byte buffer and the type F timings may be used only when the DMA channel is programmed to increment mode (not decrement), and cannot be used when the channel is programmed to operate in block mode (single transfer mode and demand mode are legal). In addition, verify transfers are not supported with type F DMA.

3.4.2 ISA REFRESH CYCLES

Refresh cycle requests are generated by two sources—the refresh controller inside the PIIX component or ISA Bus Masters other than the PIIX. In both cases, the PIIX generates the ISA memory refresh. The PIIX enables address lines SA[7:0]. Thus, when MEMR# goes active, the entire ISA system memory is refreshed at one time. Memory slaves on the ISA Bus must not drive any data onto the data bus during the refresh cycle. The PIIX maintains a four deep buffer to record internally generated refresh requests that have not been serviced. Counter 1 in the timer register set should be programmed to provide a request for refresh about every 15 μs.

PIIX Initiated Refresh Cycle

The PIIX asserts REFRESH# to indicate a refresh cycle and then drives the address lines SA[7:0] onto the ISA Bus and generates MEMR# and SMEMR#. The PIIX drives AEN and BALE high for the entire refresh cycle. The memory device may extend this refresh cycle by pulling IOCHRDY low.

ISA Bus refresh cycles are completely decoupled from DRAM Refresh. Transactions driven by PCI masters that target ISA or IDE resources while refresh is active are held off with wait states until the refresh is complete.

ISA Master Initiated Refresh Cycle

If an ISA Bus Master holds the ISA Bus longer than 15 μs, the ISA master must initiate memory refresh cycles. If the ISA Bus Master initiates a refresh cycle before it relinquishes the bus, it floats the address lines and control signals and asserts the REFRESH# to the PIIX. The PIIX drives address lines SA[7:0] and MEMR# onto the ISA Bus. BALE is driven high and AEN is driven low for the entire refresh cycle.

If the ISA Bus Master holding the bus does not a generate a refresh request and the PIIX's internal refresh request is not serviced within the normal 15 μs, a refresh queue counter is incremented. The counter records up to four incomplete refresh cycles, which are all executed as soon as PIIX gets the ISA Bus.

3.4.3 ISA DATA RETURNED ON A MASTER ABORT CYCLE

The PIIX will return random data to an ISA device on a Master Abort Cycle (i.e., ISA DMA access to main memory).

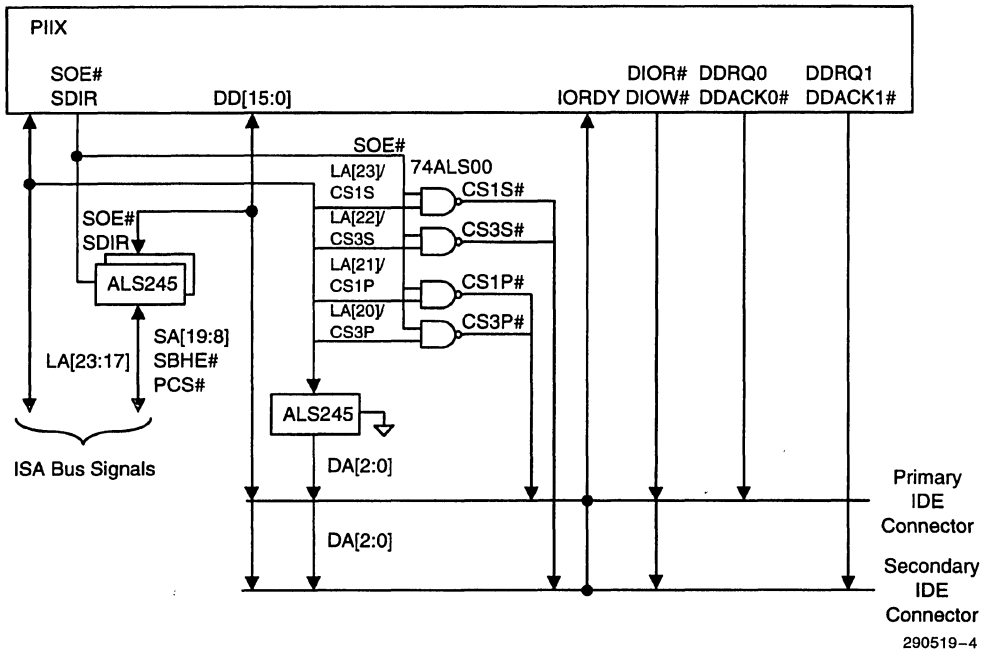
3.5 PCI Local Bus IDE

The PIIX integrates a high-performance interface from PCI to IDE. This interface is capable of accelerated PIO data transfers as well as acting as a PCI Bus Master on behalf of an IDE DMA slave device. The PIIX provides an interface for both primary and secondary IDE connectors (Figure 3).

The IDE data transfer command strobes, DMA request and grant signals, and IORDY signal interface directly to the PIIX. The IDE data lines interface directly to the PIIX, and are buffered to provide part of the ISA address bus as well as the X-Bus chip select signals. The IDE address and chip select signals are multiplexed onto the LA[23:17] lines. The IDE connector signals are driven from the LA[23:17] lines by an ALS244 buffer.

NOTE:
The IRQ14 signal (pin 83) must be used to signal interrupts for the primary channel in Bus Master mode, MIRQ[0] must be used for the secondary channel.

Only PCI masters have access to the IDE port. ISA Bus Masters cannot access the IDE I/O port addresses. Memory targeted by the IDE interface acting as a PCI Bus Master on behalf of IDE DMA slaves must reside on PCI, usually main memory implemented by the host-to-PCI bridge.



NOTES:
Support for Older Drives: There are cases where the PIIX asserts both IDE chip selects (CS1x and CS3x). Some older drives may not operate properly when both chip selects are asserted. Because the IDE chip selects are muxed with the ISA LA lines, the 74ALS00 in the figure is used to ensure proper operation of older drives by gating the LA signals with SOE#.

Figure 3. PIIX IDE Interface

3.5.1 ATA REGISTER BLOCK DECODE

The IDE ATA I/O ports are decoded by the PIIX when enabled in the PCICMD and IDETIM Registers for function 1 (ATA stands for "AT Attachment"—the specification for AT compatible drive interfaces). The actual ATA Registers are implemented in the drive itself. An access to the IDE Registers results in the assertion of the appropriate chip select for the register. The transaction is then run using compatible timing and using the IDE command strobes (DIOR#, DIOW#).

For each cable (primary and secondary), there are two I/O ranges; the command block that corresponds to the CS1x# chip select, and the control block that corresponds to the CS3x# chip select. The command block is an 8-byte range while the control block is a 4-byte range. The upper 16 bits of the I/O address are decoded as all 0s. Note that the IDE chip select signals on the PIIX must be inverted externally to provide active low signals.

Primary Command Block Offset:	01F0h
Primary Control Block Offset:	03F4h
Secondary Command Block Offset:	0170h
Secondary Control Block Offset:	0374h

Table 9 specifies the registers as they affect the PIIX hardware definition.

Table 9. IDE Legacy I/O Port Definition: COMMAND BLOCK (CS1x# Chip Select)

IO Offset	Register Function (Read/Write)	Access
00h	Data	R/W
01h	Error/Features	R/W
02h	Sector Count	R/W
03h	Sector Number	R/W
04h	Cylinder Low	R/W
05h	Cylinder High	R/W
06h	Drive/Head	R/W
07h	Status/Command	R/W

The Data Register is accessed as a 16-bit register for PIO transfers (except for ECC bytes). All other registers are accessed as 8-bit quantities.

Table 10. IDE Legacy I/O Port Definition: COMMAND BLOCK (CS3x# Chip Select)

IO Offset	Register Function (Read/Write)	Access
00h	Reserved	reserved
01h	Reserved	reserved
02h	Alt Status/Device Control	R/W
03h	Forward to ISA (Floppy)	R/W

The PIIX claims all accesses to these ranges. The byte enables do not have to be externally decoded to assert DEVSEL#. Accesses to byte 3 of the Control Block are forwarded to ISA where the floppy disk controller responds.

Each of the two drives (drive 0 or 1) on a cable implement separate ATA Register files. To determine the targeted drive, the PIIX shadows the value of bit 4 (drive bit) of byte 6 (drive/head register) of the ATA command block (CS1x#) for each of the two IDE connectors (primary and secondary).

3.5.2 ENHANCED TIMING MODES

The PIIX includes fast timing modes. The fast timing modes can be enabled only for the IDE data ports. All other transactions to the IDE Registers are run in single transaction mode with compatible timings.

Up to two IDE devices may be attached per IDE connector (drive 0 and drive 1). For each connector, only one fast timing mode may be specified (via the IDETIM Register). This mode can be applied to drive 0, drive 1, or both. Transactions targeting the other drive will use compatible timing.

3.5.2.1 Back-to-Back PIO IDE Transactions

IDE data port transaction latency consists of startup latency, cycle latency, and shutdown latency. Cycle latency consists of the I/O strobe assertion length and recovery time. Recovery time is provided so that transactions may occur back-to-back on the IDE interface (without incurring startup and shutdown latency) without violating minimum cycle periods for the IDE interface.

Startup latency is incurred when a PCI master cycle targeting the IDE data port is decoded and the DA[2:0] and CSxx# lines are not set up. Startup latency provides the setup time for the DA[2:0] and CSxx# lines with respect to the read and write strobes (DIOR# and DIOW#).

Shutdown latency is incurred after outstanding scheduled IDE data port transactions (either a non-empty write post buffer or an outstanding read prefetch cycles) have completed and before other transactions can proceed. It provides hold time on the DA[2:0] and CSxx# lines with respect to the read and write strobes (DIOR# and DIOW#).

Cycle latency is the latency incurred by each individual 16 bit IDE data port transfer, and consists of command strobe width and recovery time. The command strobe assertion width is selected by the IDETIM Register and may be set to 2, 3, 4, or 5 PCI clocks. The recovery time is selected by the IDETIM Register and may be set to 1, 2, 3, or 4 PCI clocks.

If IORDY is asserted when the initial sample point is reached, no wait states are added. If IORDY is negated when the initial sample point is reached, additional wait states are added. Since the rising edge of IORDY must be synchronized, at least two additional PCI clocks are added.

NOTE:
Bit 2 (16 bit I/O recovery enable) of the ISA Controller Recovery Timer Register does not add wait states to IDE data port read accesses when any of the fast timing modes are enabled.

3.5.2.2 IORDY Masking

The IORDY signal can be forced asserted on a drive-by-drive basis via the IDETIM Register.

3.5.2.3 PIO 32 Bit IDE Data Port Mode

If the 32-bit IDE data port mode is enabled (via bit 4 and 0 of the IDETIM Register), 32-bit accesses to the IDE data port address (default 01F0h primary, etc.) result in two back-to-back 16-bit transactions to IDE. The 32-bit data port feature is enabled for all timings, not just enhanced timing.

3.5.3 BUS MASTER FUNCTION

The PIIX can act as a PCI Bus Master on behalf of an IDE slave device. Two PCI Bus Master channels are providedone channel for each IDE connector (primary and secondary). By performing the IDE data transfer as a PCI Bus Master, the PIIX off-loads the CPU and improves system performance in multitasking environments.

NOTE:
The PCICMD Command Register (Function 1) bit 2 must be programmed to 1 for Bus Master operation.

Physical Region Descriptor Format

The physical memory region to be transferred is described by a Physical Region Descriptor (PRD). The PRDs are stored in a table in memory. The data transfer proceeds until all regions described by the PRDs in the table have been transferred. Note that the Bus Master IDE does not support memory for regions or PRDs on ISA.

Each PRD entry is 8 bytes in length. The first 4 bytes specify the byte address of a physical memory region. The next 2 bytes specify the count of the region in bytes (64-Kbyte limit per region). A value of 0 in these 2 bytes indicates 64 Kbytes. Bit 7 of the last byte indicates the end of the table (EOT). Bus Master operation terminates when the last descriptor has been completed.

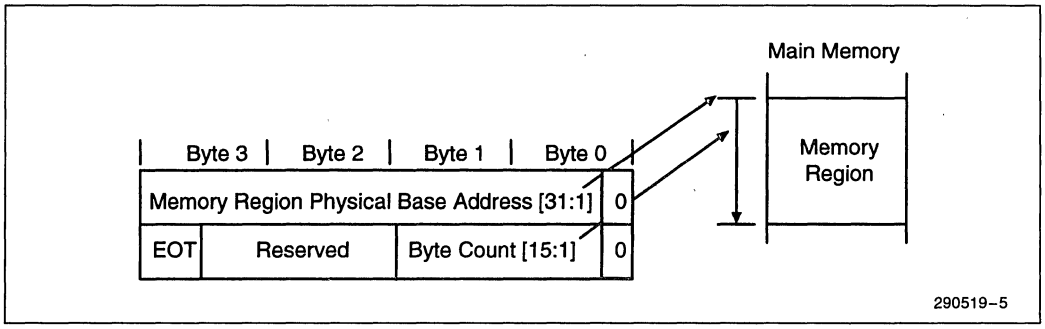


Figure 4. Physical Region Descriptor Table Entry

NOTE:
The memory region specified by the descriptor cannot straddle a 64-Kbyte boundary. This means that the byte count can be limited to 64 Kbytes and the incrementer for the current address register need only extend from bit 1 to bit 15. Also, the total sum of the descriptor byte counts must be equal to or greater than the size of the disk transfer request. If greater than the disk transfer request, the driver must terminate the Bus Master transaction (by setting bit 0 in the Bus Master IDE Command Register to 0) when the drive issues an interrupt to signal transfer completion.

Operation

To initiate a Bus Master transfer between memory and an IDE DMA slave device, the following steps are required:

1. Software prepares a PRD Table in main memory. Each PRD is 8 bytes long and consists of an address pointer to the starting address and the transfer count of the memory buffer to be transferred. In any given PRD Table, two consecutive PRDs are offset by 8-bytes and are aligned on a 4-byte boundary.
2. Software provides the starting address of the PRD Table by loading the PRD Table Pointer Register . The direction of the data transfer is specified by setting the Read/Write Control bit. Clear the Interrupt bit and Error bit in the Status Register.
3. Software issues the appropriate DMA transfer command to the disk device.
4. Engage the Bus Master function by writing 1 to the Start bit in the Bus Master IDE Command Register for the appropriate channel. The first entry in the PRD table is fetched by the PIIX. The channel remains masked until the first descriptor is loaded.
5. The controller transfers data to/from memory responding to DMA requests from the IDE device.
6. At the end of the transfer, the IDE device signals an interrupt.
7. In response to the interrupt, software resets the Start/Stop bit in the command register. It then reads the controller status and then the drive status to determine if the transfer completed successfully.

When the last data transfer for a region has been completed on the IDE interface, the next descriptor is fetched from the table. The descriptor contents are loaded into the Current Base and Current Count Registers.

The last PRD in a table has the End of List (EOL) bit set. The PCI Bus Master data transfers terminates when the physical region described by the last PRD in the table has been completely transferred. The active bit in the BMISx Register is set to 0 and the DDRQx signal is masked.

NOTE:
The IRQ14 signal (pin 83) must be used to signal interrupts for the primary channel in Bus Master mode, MIRQ[0] must be used for the secondary channel.

Line Buffer

A single line buffer exists for the PCI Bus Master IDE interface. This buffer is not shared with any other function. The buffer is maintained in either the read state or the write state. The size of the buffer is 32 bytes, and is aligned on the cache line boundary. The line buffer allows burst data transfers to proceed at peak transfer rates.

Arbitration

The two Bus Master IDE channels are fairly arbitrated (round robin between the two). The ISA DMA channels and the IDE Bus Master channels are arbitrated fairly as a group (fairness between the two groups). This arbitration is not programmable.

3.6 Interval Timer

The PIIX contains three counters that are equivalent to those found in the 82C54 programmable interval timer. The three counters are contained in one PIIX timer unit, referred to as Timer-1. Each counter output provides a key system function. Counter 0 is connected to interrupt controller IRQ0 and provides a system timer interrupt for a time-of-day, diskette time-out, or other system timing functions. Counter 1 generates a refresh request signal and Counter 2 generates the tone for the speaker. The 14.31818 MHz counters normally use OSC as a clock source.

Counter 0, System Timer

This counter functions as the system timer by controlling the state of IRQ0 and is typically programmed for mode 3 operation. The counter produces a square wave with a period equal to the product of the counter period (838 ns) and the initial count value. The counter loads the initial count value one counter period after software writes the count value to the counter I/O address. The counter initially asserts IRQ0 and decrements the count value by two each counter period. The counter negates IRQ0 when the count value reaches 0. It then reloads the initial count value and again decrements the initial count value by two each counter period. The counter then asserts IRQ0 when the count value reaches 0, reloads the initial count value, and repeats the cycle, alternately asserting and negating IRQ0.

Counter 1, Refresh Request Signal

This counter provides the refresh request signal and is typically programmed for mode 2 operation. The counter negates refresh request for one counter period (838 ns) during each count cycle. The initial count value is loaded one counter period after being written to the counter I/O address. The counter initially asserts refresh request, and negates it for 1 counter period when the count value reaches 1. The counter then asserts refresh request and continues counting from the initial count value.

Counter 2, Speaker Tone

This counter provides the speaker tone and is typically programmed for mode 3 operation. The counter provides a speaker frequency equal to the counter clock frequency (1.193 MHz) divided by the initial count value. The speaker must be enabled by a write to port 061h.

3.7 Interrupt Controller

The PIIX provides an ISA compatible interrupt controller that incorporates the functionality of two 82C59 interrupt controllers. The two controllers are cascaded so that 13 external and three internal interrupts are possible. The master interrupt controller provides IRQ [7:0] and the slave interrupt controller provides IRQ[15:8] (Figure 5). The three internal interrupts are used for internal functions only and are not available to the user. IRQ2 is used to cascade the two controllers together. IRQ0 is used as a system timer interrupt and is tied to Interval Timer 1, Counter 0. IRQ13 is connected internally to FERR#. The remaining 13 interrupt lines (IRQ[15,14,12:3,1]) are available for external system interrupts. Edge or level sense selection is programmable on an individual channel by channel basis.

The Interrupt unit also supports interrupt steering. The PIIX can be programmed to allow the four PCI active low interrupts (PIRQ[3:0]#) to be internally routed to one of 11 interrupts (IRQ[15,14,12:9,7:3]). In addition, up to six interrupt signals dedicated to motherboard devices (MIRQ[5:0]) may be routed to any of the 11 interrupts.

The Interrupt Controller consists of two separate 82C59 cores. Interrupt Controller 1 (CNTRL-1) and Interrupt Controller 2 (CNTRL-2) are initialized separately and can be programmed to operate in different modes. The default settings are: 80x86 Mode, Edge Sensitive (IRQ[15:0]) Detection, Normal EOI, Non-Buffered Mode, Special Fully Nested Mode disabled, and Cascade Mode. CNTRL-1 is connected as the Master Interrupt Controller and CNTRL-2 is connected as the Slave Interrupt Controller.

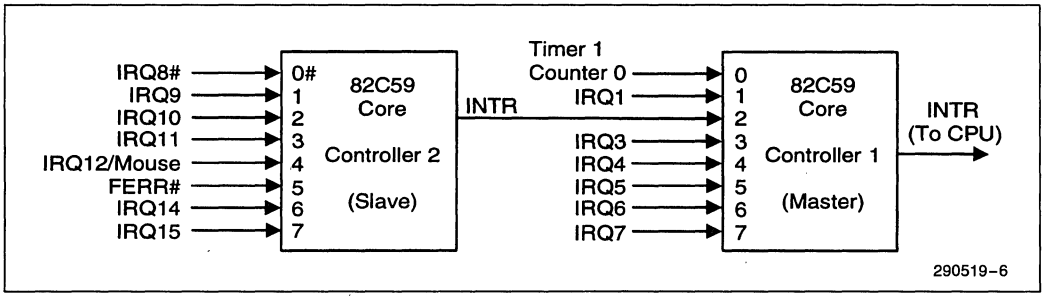


Figure 5. Block Diagram of the Interrupt Controller

Note that IRQ13 is generated internally (as part of the coprocessor error support) by the PIIX. IRQ12/M is generated internally (as part of the mouse support) when bit-4 in the XBCS Register is set to 1. When this bit is set to 0, the standard IRQ12 function is provided and IRQ12 appears externally.

3.7.1 PROGRAMMING THE ICWs/OCWs

The Interrupt Controller accepts two types of command words generated by the CPU or Bus Master:

1. **Initialization Command Words (ICWs)**: Before normal operation can begin, each Interrupt Controller in the system must be initialized. In the 82C59, this is a two to four byte sequence. However, for the PIIX, each controller must be initialized with a four byte sequence. This four byte sequence is required to configure the interrupt controller correctly for the PIIX implementation. This implementation is ISA-compatible. The four initialization command words are referred to by their acronyms: ICW1, ICW2, ICW3, and ICW4. The base address for each interrupt controller is a fixed location in the I/O memory space, at 0020h for CNTRL-1 and at 00A0h for CNTRL-2.

   An I/O write to the CNTRL-1 or CNTRL-2 base address with data bit 4 equal to 1 is interpreted as ICW1. For PIIX-based ISA systems, three I/O writes to "base address + 1" (021h for CNTRL-1 and 0A0h for CNTRL-2) must follow the ICW1. The first write to "base address + 1" (021h/0A0h) performs ICW2, the second write performs ICW3, and the third write performs ICW4.
2. **Operation Command Words (OCWs)**: These are the command words that dynamically reprogram the interrupt controller to operate in various interrupt modes. Any interrupt lines can be masked by writing an OCW1. A 1 written in any bit of this command word masks incoming interrupt requests on the corresponding IRQx line. OCW2 is used to control the rotation of interrupt priorities when operating in the rotating priority mode and to control the End of Interrupt (EOI) function of the controller. OCW3 set up reads of the ISR and IRR, enable/disables the Special Mask Mode (SMM), and sets up the interrupt controller in polled interrupt mode. The OCWs can be written to the Interrupt Controller any time after initialization.

3.7.2 EDGE AND LEVEL TRIGGERED MODE

In ISA systems this mode is programmed using bit 3 in ICW1. With PIIX this bit is disabled and a new register for edge- and level-triggered mode selection (per interrupt input) is included. This is the Edge/Level control Registers ELCR1 and ELCR2. The default programming is equivalent to programming the LTIM bit (ICW1 bit 3) to 0 (all interrupts selected for edge triggered mode). Note that IRQ0, 1, 2, 8#, and 13 can not be programmed for level-sensitive mode and can not be modified by software.

If an ELCR bit = 0, an interrupt request is recognized by a low-to-high transition on the corresponding IRQx input. The IRQ input can remain high without generating another interrupt.

If an ELCR bit = 1, an interrupt request is recognized by a low level on the corresponding IRQ input and there is no need for an edge detection. The interrupt request must be removed before the EOI command is issued to prevent a second interrupt from occurring.

In both the edge and level triggered modes, the IRQ inputs must remain active until after the falling edge of the first INTERRUPT ACKNOWLEDGE CYCLE. If the IRQ input goes inactive before this time, a default IRQ7 occurs when the CPU acknowledges the interrupt. This can be a useful safeguard for detecting interrupts caused by spurious noise glitches on the IRQ inputs. To implement this feature, the IRQ7 routine is used for "clean up" simply executing a return instruction, thus ignoring the interrupt. If IRQ7 is needed for other purposes, a default IRQ7 can still be detected by reading the ISR. A normal IRQ7 interrupt sets the corresponding ISR bit. A default IRQ7 does not set this bit. However, if a default IRQ7 routine occurs during a normal IRQ7 routine, the ISR remains set. In this case, it is necessary to keep track of whether or not the IRQ7 routine was previously entered. If another IRQ7 occurs, it is a default.

3.7.3 INTERRUPT STEERING

The PIIX can be programmed to allow four PCI programmable interrupts (PIRQ[3:0]#) to be internally routed to one of 11 interrupts (IRQ[15,14,12:9,7:3]) using the PIRQx Route Control Register. PCLK is used to synchronize the PIRQx# inputs. The assignment is programmable through the PIRQx Route Control Registers. One or more PIRQx# lines can be routed to the same IRQx input. If interrupt steering is not required, the Route Registers can be programmed to disable steering.

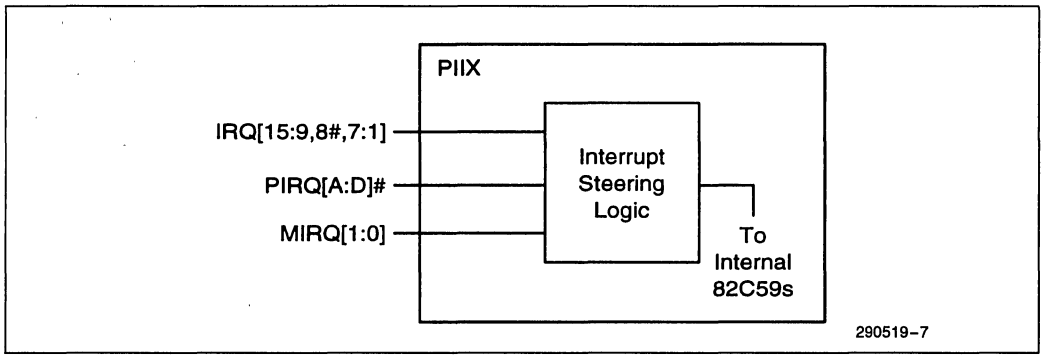


Figure 6. Interrupt Steering

The PIRQx# lines are defined as active low, level-sensitive to allow multiple interrupts on a PCI Board to share a single line across the connector. When a PIRQx# is routed to a specified IRQ line, the software must change the IRQ's corresponding ELCR bit to level-sensitive mode. Note that this means that the selected IRQ can no longer be used by an ISA device, unless that ISA device can respond as an active low level sensitive interrupt.

The PIIX also supports up to two programmable interrupts (MIRQ[1:0]; intended for use with motherboard devices) to be routed to one of the 11 interrupts (IRQ[15,14,12:9,7:3]) using the MBIRQx Route Control Register. The routing is accomplished in the same manner as for the PIRQx# inputs, except that the interrupts are active high. Two MIRQx lines may be routed to the same IRQx input. If interrupt steering is not required, the MBIRQx Registers can be programmed to disable routing.

When more than one MIRQ line is routed to an IRQ input, the software must change the IRQ's corresponding ELCR bit to level sensitive mode. Interrupt sharing for motherboard devices must be evaluated for the particular combination of devices under consideration. The IRQ selected bit MBIRQx[3:0] can no longer be used by an ISA device, unless that ISA device can respond as an active high level sensitive interrupt.

3.8 X-Bus Peripheral Support

The PIIX provides positive decode (chip selects) and X-Bus buffer control (XDIR# and XOE#) for a real time clock, keyboard controller and BIOS for PCI and ISA initiated cycles. The PIIX also generates RTCALE (address latch enable) for the RTC. The chip selects are generated combinatorially from the ISA SA(16:0) and LA (23:17) address lines (it is assumed that ISA masters drive SA(19:16) and LA(23:17) low when accessing I/O devices). The PIIX also provides PS/2 mouse support via the IRQ12/M signal and coprocessor functions (FEER# and IGNNE#). The chip selects and X-Bus buffer control lines can be enabled/disabled via the XBCS Register.

Coprocessor Error Function

This function provides coprocessor error support for the CPU and is enabled via the XBCS Register. FERR# is tied directly to the coprocessor error signal of the CPU. If FERR# is driven active to the PIIX, an internal IRQ13 is generated and the INTR output from the PIIX is driven active. When a write to I/O location F0h is detected, the PIIX negates IRQ13 (internal to the PIIX) and drives IGNNE# active. IGNNE# remains active until FERR# is driven inactive. IGNNE# is not driven active unless FERR# is active.

Mouse Function

When the mouse interrupt function is enabled (via the XBCS Register), the mouse interrupt function is provided on the IRQ12/M input signal. In this mode, a mouse interrupt generates an interrupt through IRQ12 to the Host CPU. The PIIX informs the CPU of this interrupt via a INTR. A read of 60h releases IRQ12. If the mouse interrupt function is disabled, a read of address 60h has no effect on IRQ12/M. Reads and writes to this register flow through to the ISA Bus. For additional information, see the IRQ12/M description in the Signal Description.

3.9 Power Management

The PIIX has extensive power management capability permitting a system to operate in a low power state without being powered down. In a typical desktop personal computer there are two statesPower-On and Power-Off. Leaving a system powered on when not in use wastes power. The PIIX provides a Fast-On/Off feature that creates a third state called Fast-Off (Figure 7). When in the Fast-Off state, the system consumes less power than the Power-On state.

The PIIX's power management architecture is based on three functions—System Management Mode (SMM), Clock Control, and Advanced Power Management (APM). Software (called SMM code) controls the transitions between the Power-On state and the Fast-Off state. The PIIX invokes this software by generating an SMI to the CPU (asserting the SMI# signal). A variety of programmable events are provided that can generate an SMI. The SMM code places the system in either the Power-On state or the Fast-Off state.

A Fast-On event is an event that instructs the computer (via an SMI to the CPU) to enter the Power-On state in anticipation of system activity by the user. Fast-On events are programmable and include moving the mouse, pressing a key on the keyboard, an external hardware event, an incoming call to a system FAX/Modem, a RTC alarm, or the operating system.

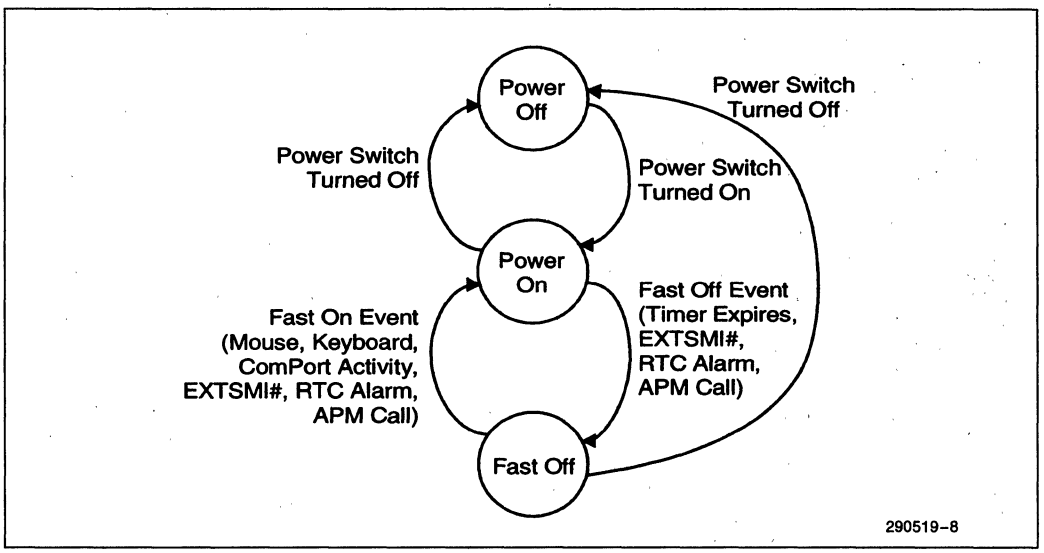


Figure 7. Fast-On/Off Flow

3.9.1 SMM MODE

SMM mode is invoked by asserting the SMI# signal to the CPU. The PIIX provides a variety of programmable events that can generate an SMI. When the CPU receives an SMI, it enters SMM mode and executes SMM code out of SMRAM. Depending on the current state, the SMM code places the system in either the Power-On state or the Fast-Off state. In the Power-On state, the computer system operates normally. In this state, one of the four programmable events listed below can trigger an SMI.

1. A global idle timer called the Fast-Off Timer expires (an indication that the end user has not used the computer for a programmed period of time).
2. The EXTSMI# pin is asserted.
3. The operating system issues an APM call.

3.9.2 SMI SOURCES

The SMI# signal can be asserted by hardware interrupt events, the Fast-Off Timer, an external SMI event (EXTSMI#), and software events (via the APMC and APMS Registers). Enable/disable bits (in the SMIEN Register) permit each event to be individually masked from generating an SMI. In addition, the SMI# signal can be globally enabled/disabled in the SMICNTL Register. Status of the individual events causing an SMI is provided in the SMIREQ Register. For detailed information on the SMI control/status registers, refer to the Register Description section.

Hardware Interrupt Events

Hardware events (IRQ[12,8#,4,3,1] and the Fast-Off Timer) are enabled/disabled from generating an SMI in the SMIEN Register. When enabled, the occurrence of the corresponding hardware event generates an SMI (asserts the SMI# signal), regardless of the current power state of the system.

Fast-Off Timer

The Fast-Off Timer is used to indicate (through an SMI) that the system has been idle for a programmed period of time. The timer counts down from a programmed start value and when the count reaches 00h, can generate an SMI. The timer decrement rate is programmable (via the SMICNTL Register) and is re-loaded each time a system event occurs. This counter should not be programmed to 00h. System and break events are described in the SEE Register.

EXTSMI#

The EXTSMI# input pin provides the system designer the capability to invoke SMM with external hardware. For example, the EXTSMI# input could be connected to a "green button" permitting the user to enter the Fast-Off state by depressing a button. The EXTSMI# generation of an SMI is enabled/disabled in the SMIEN Register.

Software Events

Software events (accessing the APMx Registers) indicate that the OS is passing power management information to the SMI handler. There are two Advanced Power Management (APM) Registers—APM Control (APMC) and APM Status (APMS) Registers. These registers permit software to generate an SMI; by writing to the APMC Register. For example, the APMC Register can be used to pass an APM command between APM OS and BIOS and the APMS Register could be used to pass data between the OS and the SMI handler.

The two APM Registers are located in normal I/O space. The PIIX subtractively decodes PCI accesses to these registers and forwards the accesses to the ISA Bus. The APM Registers are not accessible by ISA masters. Note that the remaining power management registers are located in PCI Configuration space.

3.9.3 CLOCK CONTROL

The CPU can be put in a low power state by asserting the STPCLK# signal. STPCLK# is an interrupt to the CPU. However, for this type of interrupt, the CPU does not generate an interrupt acknowledge cycle. Once the STPCLK# interrupt is executed, the CPU enters the stop-grant state. In this state, the CPU's internal clocks are disabled and instruction execution is stopped. The stop-grant state is exited when the STPCLK# signal is negated.

Software can assert STPCLK# (if enabled via the SMICNTL Register) by a read of the APMC Register. Note that STPCLK# can also be periodically asserted by using clock scaling as described below.

The PIIX automatically negates STPCLK# when a break event occurs (if enabled in the SEE Register). Software can negate STPCLK# by disabling STPCLK# in the SMICNTL Register or by a write to the APMC Register.

Clock Scaling (Emulating Clock Division)

Clock scaling permits the PIIX to periodically place the CPU in a low power state. This emulates clock division. When clock scaling is enabled, the CPU runs at full frequency for a pre-defined time period and then is stopped for a pre-defined time period. The run/stop time interval ratio emulates the clock division effect from a power/performance point of view. However, clock scaling is more effective than dividing the CPU frequency. For example, if the CPU is in the stop grant state and a break event occurs, the CPU clock returns to full frequency. In addition, there is no recovery time latency to start the clock.

Two programmable 8-bit clock scale timer control registers set the STPCLK# high (negate) and low (assert) times—the CTLTMRH and CTLTMRL Registers. The timer is clocked by a 32-μsec internal clock. This allows a programmable timer interval for both the STPCLK# high and low times of 0–8 msec.

3.10 Reset Support

The PIIX integrates the system reset logic for the system. The PIIX generates CPURST, PCIRST#, and RSTDRV during power up (PWROK) and when a hard reset is initiated through the RC Register. The following PIIX signals interface directly to the processor: CPURST, INTR, NMI, IGNNE#, SMI#, and STPCLK#.

These signals are open drain. Thus, external logic is not required for interfacing with the processors based on 3.3V technology which do no support 5V tolerant input buffers. During power-up these signals are driven low to prevent problems associated with 5V/3.3V power sequencing.

Some PCI devices may drive 3.3V friendly signals directly to 3.3V devices that are not 5V tolerant. If such signals are powered from the 5V supply, they must be driven low when PCIRST# is asserted. Some of these signals may need to be driven high before CPURST is negated. PCIRST# is negated 1 ms to 2 ms before CPURST to allow time for this to occur.

3.10.1 HARDWARE STRAPPING OPTIONS

There are two hardware strapping options on the PIIX. The SYSCLK signal is used during a hard reset to select the ISA clock divisor (sampled high for divisor of 3 MHz–25 MHz PCI operation; sampled low for a divisor of 4 MHz–33 MHz or 30 MHz PCI operation.

4.0 PINOUT AND PACKAGE INFORMATION

4.1 82371FB (PIIX) Pinout

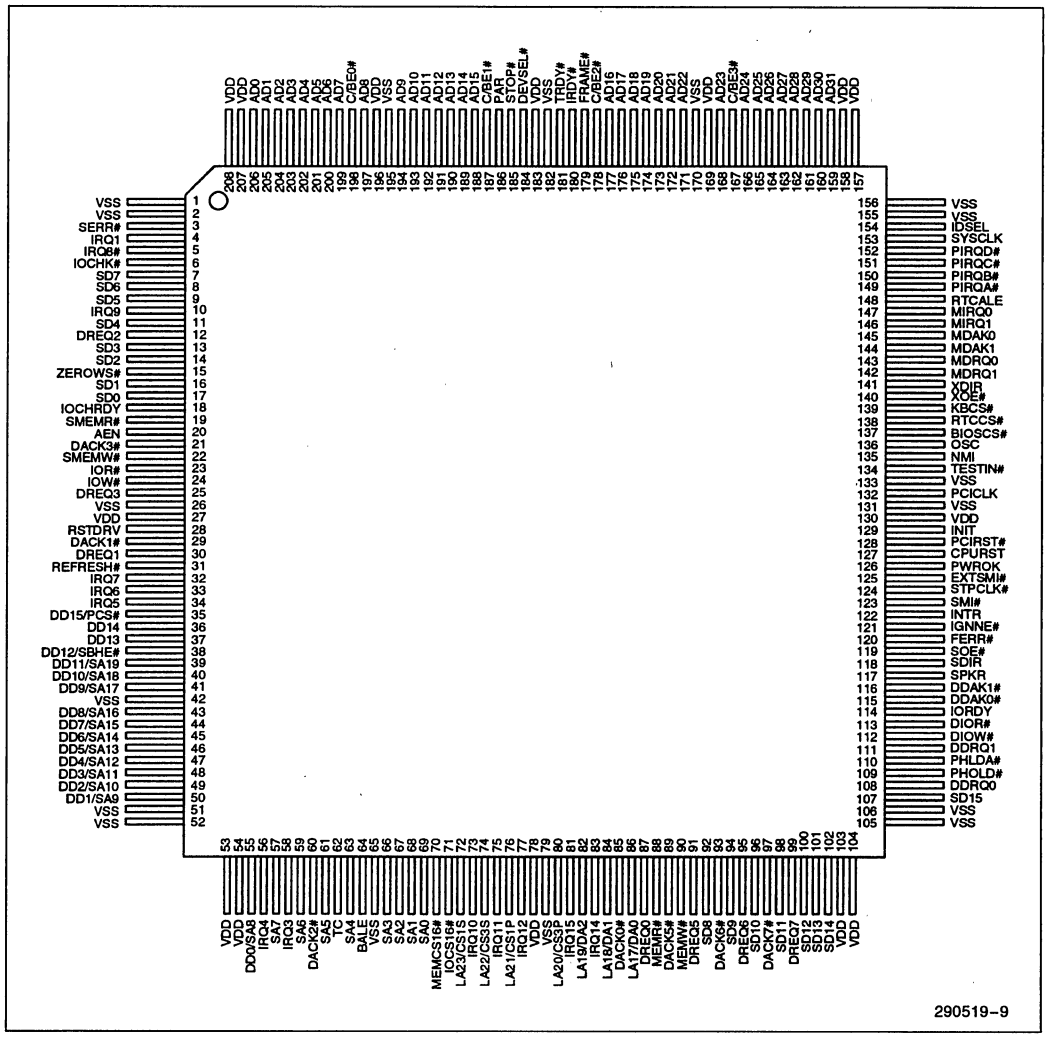


Figure 8. 82371FB (PIIX) Pinout

Table 11. Alphabetical Pin Assignment

Name	Pin#	Type
AD0	206	I/O
AD1	205	I/O
AD2	204	I/O
AD4	202	I/O
AD5	201	I/O
AD6	200	I/O
AD7	199	I/O
AD8	197	I/O
AD9	194	I/O
AD10	193	I/O
AD11	192	I/O
AD12	191	I/O
AD13	190	I/O
AD14	189	I/O
AD15	188	I/O
AD16	177	I/O
AD17	176	I/O
AD18	175	I/O
AD19	174	I/O
AD20	173	I/O
AD21	172	I/O
AD22	171	I/O
AD23	168	I/O
AD24	166	I/O
AD25	165	I/O
AD26	164	I/O
AD27	163	I/O
AD28	162	I/O
AD29	161	I/O
AD3	203	I/O

Name	Pin#	Type
AD30	160	I/O
AD31	159	I/O
AEN	20	O
BALE	64	O
BIOSCS#	137	O
C/BE0#	198	I/O
C/BE1#	187	I/O
C/BE2#	178	I/O
C/BE3#	167	I/O
CPURST	127	O
DACK0#	85	O
DACK1#	29	O
DACK2#	60	O
DACK3#	21	O
DACK5#	89	O
DACK6#	93	O
DACK7#	97	O
DD0/SA8	55	I/O
DD1/SA9	50	I/O
DD2/SA10	49	I/O
DD3/SA11	48	I/O
DD4/SA12	47	I/O
DD5/SA13	46	I/O
DD6/SA14	45	I/O
DD7/SA15	44	I/O
DD8/SA16	43	I/O
DD9/SA17	41	I/O
DD10/SA18	40	I/O
DD11/SA19	39	I/O
DD12/SBHE#	38	I/O

Name	Pin#	Type
DD13	37	I/O
DD14	36	I/O
DD15/PCS#	35	I/O
DDAK0#	115	O
DDAK1#	116	O
DDRQ0	108	I
DDRQ1	111	I
DEVSEL#	184	I/O
DIOR#	113	O
DIOW#	112	O
DREQ0	87	I
DREQ1	30	I
DREQ2	12	I
DREQ3	25	I
DREQ5	91	I
DREQ6	95	I
DREQ7	99	I
EXTSMI#	125	I
FERR#	120	I
FRAME#	179	I/O
IDSEL	154	I
IGNNE#	121	O
INIT	129	O
INTR	122	O
IOCHK#	6	I
IOCHRDY	18	I/O
IOCS16#	71	I
IOR#	23	I/O
IORDY	114	I
IOW#	24	I/O

Table 11. Alphabetical Pin Assignment (Continued)

Name	Pin#	Type
IRDY#	180	I/O
IRQ1	4	I
IRQ3	58	I
IRQ4	56	I
IRQ5	34	I
IRQ6	33	I
IRQ7	32	I
IRQ8#	5	I
IRQ9	10	I
IRQ10	73	I
IRQ11	75	I
IRQ12	77	I
IRQ14	83	I
IRQ15	81	I
KBCS#	139	O
LA17/DA0	86	I/O
LA18/DA1	84	I/O
LA19/DA2	82	I/O
LA20/CS3P	80	I/O
LA21/CS1P	76	I/O
LA22/CS3S	74	I/O
LA23/CS1S	72	I/O
MDAK1	144	O
MDAK0	145	O
MDRQ0	143	I
MDRQ1	142	I
MEMCS16#	70	I/O
MEMR#	88	I/O
MEMW#	90	I/O
MIRQ0	147	O
MIRQ1	146	I

Name	Pin#	Type
NMI	135	O
OSC	136	I
PAR	186	O
PCICLK	132	I
PCIRST#	128	O
PHLDA#	110	I
PHOLD#	109	O
PIRQA#	149	I
PIRQB#	150	I
PIRQC#	151	I
PIRQD#	152	I
PWROK	126	I
REFRESH#	31	I/O
RSTDRV	28	O
RTCALE	148	O
RTCCS#	138	O
SA0	69	I/O
SA1	68	I/O
SA2	67	I/O
SA3	66	I/O
SA4	63	I/O
SA5	61	I/O
SA6	59	I/O
SA7	57	I/O
SD0	17	I/O
SD1	16	I/O
SD2	14	I/O
SD3	13	I/O
SD4	11	I/O
SD5	9	I/O
SD6	8	I/O

Name	Pin#	Type
SD7	7	I/O
SD8	92	I/O
SD9	94	I/O
SD10	96	I/O
SD11	98	I/O
SD12	100	I/O
SD13	101	I/O
SD14	102	I/O
SD15	107	I/O
SDIR	118	O
SERR#	3	I
SMEMR#	19	O
SMEMW#	22	O
SMI#	123	O
SOE#	119	O
SPKR	117	O
STOP#	185	I/O
STPCLK#	124	O
SYSCLK	153	O
TC	62	O
TESTIN#	134	I
TRDY#	181	I/O
VDD	27	V
VDD	53	V
VDD	54	V
VDD	78	V
VDD	103	V
VDD	104	V
VDD	130	V
VDD	157	V
VDD	158	V

Table 11. Alphabetical Pin Assignment (Continued)

Name	Pin#	Type
VDD	169	V
VDD	183	V
VDD	196	V
VDD	207	V
VDD	208	V
VSS	1	V
VSS	2	V
VSS	26	V
VSS	42	V

Name	Pin#	Type
VSS	51	V
VSS	52	V
VSS	65	V
VSS	79	V
VSS	105	V
VSS	106	V
VSS	131	V
VSS	133	V
VSS	155	V

Name	Pin#	Type
VSS	156	V
VSS	170	V
VSS	182	V
VSS	195	V
XDIR	141	I
XOE#	140	O
ZEROWS#	15	I

5.0 PACKAGE DIMENSIONS

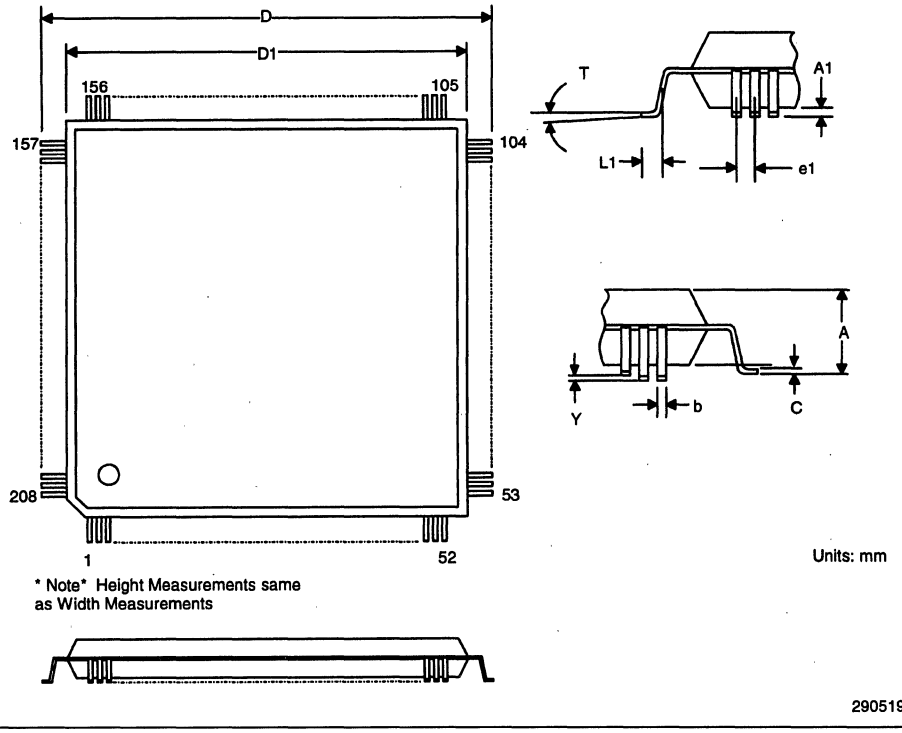


Figure 9. 208-Pin Quad Flat Pack (QFP) Dimensions

Table 12. 208-Pin Quad Flat Pack (QFP) Dimensions

Symbol	Description	Value (mm)
A	Seating Height	4.25 (max)
A1	Stand-off	0.05 (min); 0.40 (max)
b	Lead Width	0.2 ± 0.10
C	Lead Thickness	0.15 +0.1/-0.05
D	Package Length and Width, including pins	30.6 ± 0.4
D1	Package Length and Width, excluding pins	28 ± 0.2
e1	Linear Lead Pitch	0.5 ± 0.1
Y	Lead Coplanarity	0.08 (max)
L1	Foot Length	0.5 ± 0.2
T	Lead Angle	0° - 10°

6.0 82371FB TESTABILITY

6.1 Test Mode Description

The test modes are decoded from the IRQ inputs (IRQ 7, 6, 5) and qualified with the TESTIN# pin. Test mode selection is asynchronous. These signals need to remain in their respective state for the duration of the test modes. The test modes are defined as follows.

Test Mode	IRQ7	IRQ6	IRQ5	TESTIN#
NAND Tree	0	x	x	0
NAND Tree	x	x	0	0
Tri-state All Outputs	1	1	1	0

6.2 NAND Tree Mode

Tri-states all outputs and bi-directional buffers except for XDIR and DACK1#. Every output buffer except for XDIR and DACK1# is configured as an input in NAND tree mode and included in the NAND chain. The first input of the NAND chain is MDRQ1, and the NAND chain is routed counter-clockwise around the chip (e.g., MDRQ1, MDRQ0, MDAK1#, . . .). DACK1# is an intermediate output, and XDIR is the final output. PCICLK and TESTIN# are the only input pins not included in the NAND chain. Note in the table above there are two possible ways to select NAND tree test mode.

To perform a NAND tree test, all pins included in the NAND tree should be driven to 1 except for the following pins, which use inverting Schmitt trigger inputs and should be driven to 0:

Pin #	Pin Name
4	IRQ1
5	IRQ8#
6	IOCHK#
10	IRQ9
15	ZEROWS#
32	IRQ7
33	IRQ6
34	IRQ5
56	IRQ4
58	IRQ3
73	IRQ10
75	IRQ11
77	IRQ12
81	IRQ15
83	IRQ14
126	PWROK

Beginning with MDRQ and working counter-clockwise around the chip, each pin can be toggled and a resulting toggle observed on DACK1# or XDIR. The DACK1# output is provided so that the NAND tree test can be divided into two sections.

Table 13. NAND Tree

Pin #	Pin Name	Notes
134	TESTIN#	TESTIN# should be driven to 0 for the duration of the NAND tree test.
32	IRQ7	Test mode select signal.
33	IRQ6	Test mode select signal.
34	IRQ5	Test mode select signal.
142	MDRQ1	
143	MDRQ0	
144	MDK1	
145	MDK0	
146	MIRQ1	
147	MIRQ0	
148	RTCALE	
149	PIRQA#	
150	PIRQB#	
151	PIRQC#	
152	PIRQD#	
153	SYSCLK	
154	IDSEL	
159	AD31	
160	AD30	
161	AD29	
162	AD28	
163	AD27	
164	AD26	
165	AD25	
166	AD24	
167	C/BE3#	
168	AD23	
171	AD22	
172	AD21	
173	AD20	
174	AD19	
175	AD18	
176	AD17	
177	AD16	
178	C/BE2#	
179	FRAME#	
180	IRDY#	
181	TRDY#	
184	DEVSEL#	
185	STOP#	
186	PAR	
187	C/BE1#	
188	AD15	
189	AD14	
190	AD13	
191	AD12	
192	AD11	
193	AD10	
194	AD9	
197	AD8	
198	C/BE0#	
199	AD7	
200	AD6	
201	AD5	
202	AD4	
203	AD3	
204	AD2	
205	AD1	

Table 13. NAND Tree (Continued)

Pin #	Pin Name	Notes
206	AD0	
3	SERR#	
4	IRQ1	Inverted input signal
5	IRQ8#	Inverted input signal
6	IOCHK#	Inverted input signal
10	IRQ9	Inverted input signal
11	SD4	
12	DREQ2	
13	SD3	
14	SD2	
15	ZEROWS#	Inverted input signal
16	SD1	
17	SD0	
18	IOCHRDY	
19	SMEMR#	
20	AEN	
21	DACK3#	
22	SMEMW#	
23	IOR#	
24	IOW#	
25	DREQ3	
28	RSTDRV	
29	DACK1#	Intermediate NAND-tree output.
30	DREQ1	
31	REFRESH#	
32	IRQ7	Inverted input signal
33	IRQ6	Inverted input signal
34	IRQ5	Inverted input signal
35	DD15	
36	DD14	

Pin #	Pin Name	Notes
37	DD13	
38	DD12	
39	DD11	
40	DD10	
41	DD9	
43	DD8	
44	DD7	
45	DD6	
46	DD5	
47	DD4	
48	DD3	
49	DD2	
50	DD1	
55	DD0	
56	IRQ4	Inverted input signal
57	SA7	
58	IRQ3	Inverted input signal
59	SA6	
60	DACK2#	
61	SA5	
62	TC	
63	SA4	
64	BALE	
66	SA3	
67	SA2	
68	SA1	
69	SA0	
70	MEMCS16#	
71	IOCS16#	
72	LA23	

Table 13. NAND Tree (Continued)

Pin #	Pin Name	Notes
73	IRQ10	Inverted input signal
74	LA22	
75	IRQ11	Inverted input signal
76	LA21	
77	IRQ12	Inverted input signal
80	LA20	
81	IRQ15	Inverted input signal
82	LA19	
83	IRQ14	Inverted input signal
84	LA18	
85	DACK0#	
86	LA17	
87	DREQ0	
88	MEMR#	
89	DACK5#	
90	MEMW#	
91	DREQ5	
92	SD8	
93	DACK6#	
94	SD9	
95	DREQ6	
96	SD10	
97	DACK7#	
98	SD11	
99	DREQ7	
100	SD12	
101	SD13	
102	SD14	
107	SD15	
108	DDRQ0	

Pin #	Pin Name	Notes
109	PHOLD#	
110	PHLDA#	
111	DDRQ1	
112	DIOW#	
113	DIOR#	
114	IORDY	
115	DDAK0#	
116	DDAK1#	
117	SPKR	
118	SDIR	
119	SOE#	
120	FERR#	
121	IGNNE#	
122	INTR	
123	SMI#	
124	STPCLK#	
125	EXTSMI#	
126	PWROK	Inverted input signal
127	CPURST#	
128	PCIRST#	
129	INIT	
132	PCICLK	Input only, not included in the NAND tree test mode.
135	NMI	
136	OSC	
137	BIOSCS#	
138	RTCCS#	
139	KBCS#	
140	XOE#	
141	XDIR	Final NAND tree output

Figure 10 is a schematic of the NAND tree circuitry.

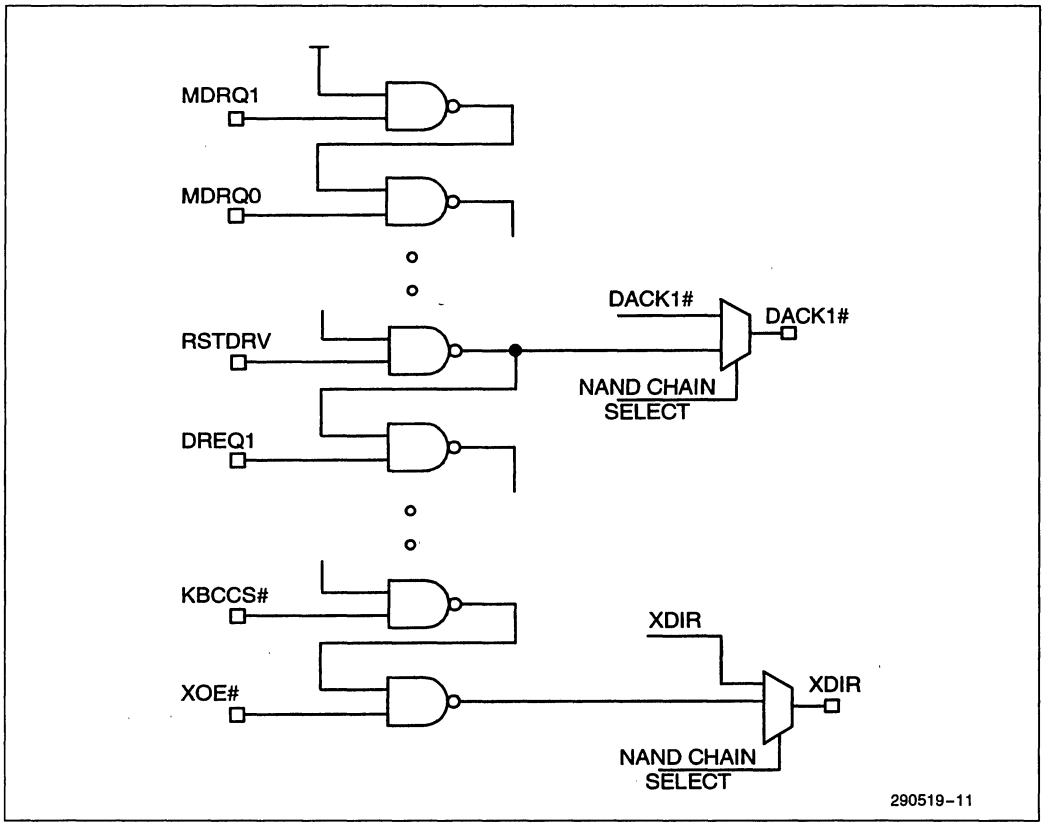


Figure 10. NAND Tree Circuitry

NAND Tree Timing Requirements

Allow 500 ns for the input signals to propagate to the NAND tree outputs (input-to-output propagation delay specification).

6.3 Tri-state Mode

The TESTIN# signal must be 0 and IRQ's 7, 6, and 5 must be 1 to enter the tri-state test mode. When in the tri-state test mode, all outputs and bi-directional pins are tri-stated, including the NAND tree outputs.

82430LX/82430NX PCIset

- Supports the Pentium® Processor at 60 and 66 MHz (82430LX)
- Supports the Pentium Processor at ICOMP™ Index 735\90 MHz, Pentium Processor at ICOMP Index 815\100 MHz, and Pentium Processor at ICOMP Index 610\75 MHz
- Supports Uni-Processor (UP) or Dual-Processor (DP) Configurations
- Interfaces the Host and Standard Buses to the PCI Local Bus
  - Up to 132 MBytes/Sec Transfer Rate
  - Full Concurrency Between CPU Host Bus and PCI Bus Transactions
- Integrated Cache Controller Provided for Optional Second Level Cache
  - 256 KByte or 512 KByte Cache
  - Write-Back or Write-Through Policy (82430LX)
  - Write-Back Policy (82430NX)
  - Standard or Burst SRAM
- Integrated Tag RAM for Cost Savings on Second Level Cache
- Supports the Pipelined Address Mode of the Pentium Processor for Higher Performance
- Provides a 64-Bit Interface to DRAM Memory
  - From 2 MBytes to 512 MBytes of Main Memory
  - 70 ns and 60 ns DRAMs Supported
- Optional ISA or EISA Standard Bus Interface
  - Single Component ISA Controller
  - Two Component EISA Bus Interface
  - Minimal External Logic Required
- Supports Burst Read and Writes of Memory from the CPU and PCI Buses
- Five Integrated Write Posting and Read Prefetch Buffers Increase CPU and PCI Performance
- Host CPU Writes to PCI Converted to Zero Wait-State PCI Bursts with Optional TRDY# Connection
- Integrated Low Skew Host Bus Clock Driver for Cost and Board Space Savings
- PCIset Operates Synchronous to the CPU and PCI Clocks
- Byte Parity Support for the Host and Main Memory Buses
  - Optional Parity on the Second Level Cache

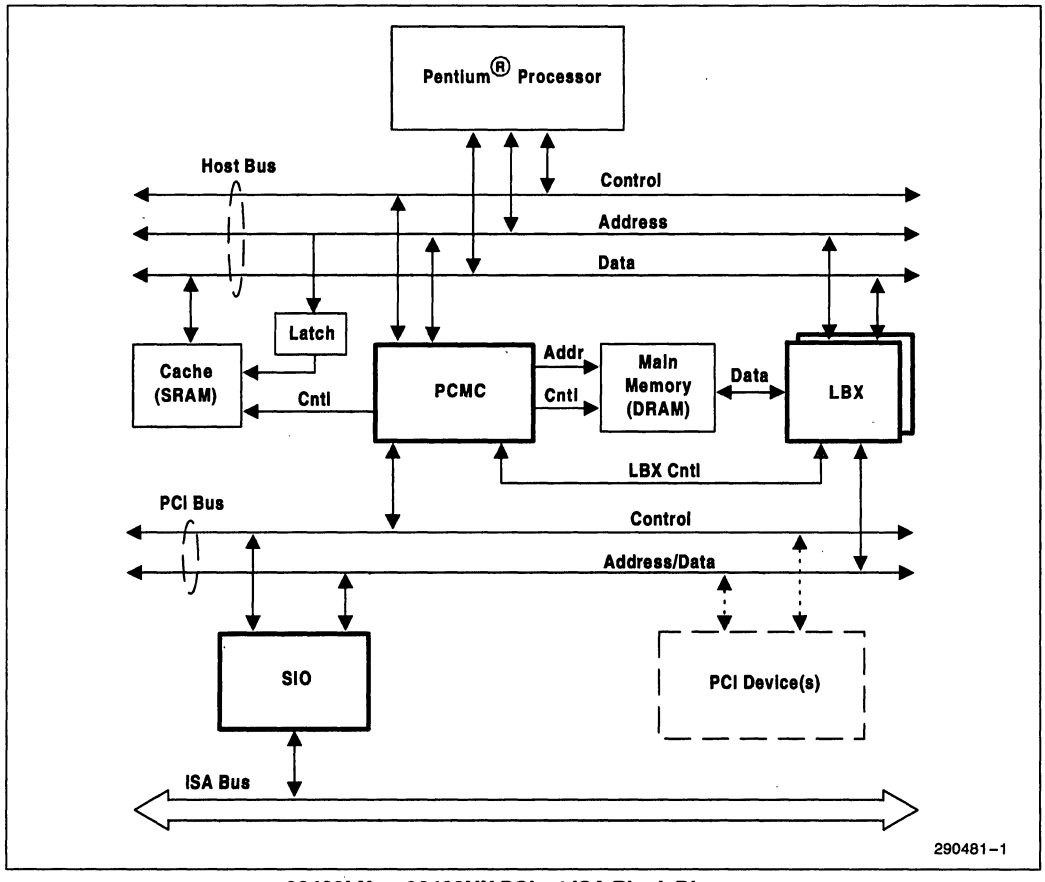
The 82430LX/82430NX PCIsets provide the Host/PCI bridge, cache/main memory controller, and an I/O subsystem core (either PCI/EISA or PCI/ISA bridge) for the next generation of high-performance personal computers based on the Pentium processor. System designers can take advantage of the power of the PCI Local bus for the local I/O while maintaining access to the large base of EISA and ISA expansion cards, and corresponding software applications. Extensive buffering and buffer management within the bridges ensures maximum efficiency in all three bus environments (Host CPU, PCI, and EISA/ISA Buses).

The 82430LX PCIset consists of the 82434LX PCI/Cache Memory Controller (PCMC) and the 82433LX Local Bus Accelerator (LBX) components, plus, either a PCI/ISA bridge or a PCI/EISA bridge. The PCMC and LBX provide the core cache and main memory architecture and serve as the Host/PCI bridge. For an ISA-based system, the 82430LX PCIset includes the 82378ZB System I/O (SIO) component as the PCI/ISA bridge. For an EISA-based system, the 82430LX PCIset includes the 82375EB/SB PCI/EISA Bridge (PCEB) and the 82374EB/SB EISA System Component (ESC). The PCEB and ESC work in tandem to form the complete PCI/EISA bridge. Both the ISA and EISA-based systems are shown on the following pages.

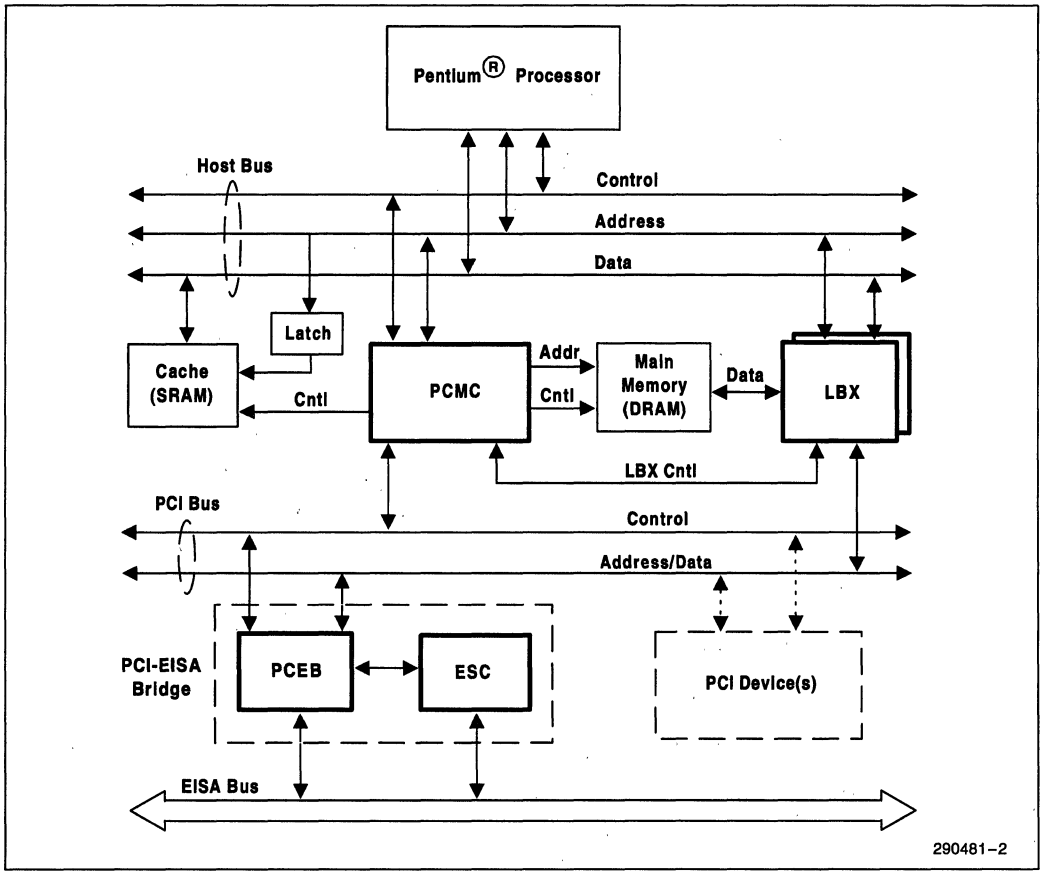
The 82430NX PCIset consists of the 82434NX PCI/Cache Memory Controller (PCMC) and the 82433NX Local Bus Accelerator (LBX) components, plus, either a PCI/ISA bridge or a PCI/EISA bridge. For an ISA-based system, the 82430NX PCIset includes the 82378ZB System I/O (SIO) component as the PCI/ISA bridge. For the DP ISA based system, the 82430NX PCIset includes the 82379AB. For UP or DP EISA-based systems, the 82430NX PCIset includes the 82375EB/SB PCI/EISA Bridge (PCEB) and the 82374EB/SB EISA System Component (ESC).

This document describes both the 82430LX and 82430NX. Unshaded areas describe the 82434LX. Shaded areas, like this one, describe 82430NX operations that differ from the 82434LX.

November 1995
Order Number: 290481-005



82430LX or 82430NX PCIset ISA Block Diagram



82430LX or Uni-Processor 82430NX PCIset EISA Block Diagram

82433LX/82433NX
LOCAL BUS ACCELERATOR (LBX)

- **Supports the Full 64-bit Pentium® Processor Data Bus at Frequencies up to 66 MHz (82433LX and 82433NX)**
- **Drives 3.3V Signal Levels on the CPU Data and Address Buses (82433NX)**
- **Provides a 64-Bit Interface to DRAM and a 32-Bit Interface to PCI**
- **Five Integrated Write Posting and Read Prefetch Buffers Increase CPU and PCI Performance**
  - **— CPU-to-Memory Posted Write Buffer 4 Qwords Deep**
  - **— PCI-to-Memory Posted Write Buffer Two Buffers, 4 Dwords Each**
  - **— PCI-to-Memory Read Prefetch Buffer 4 Qwords Deep**
  - **— CPU-to-PCI Posted Write Buffer 4 Dwords Deep**
  - **— CPU-to-PCI Read Prefetch Buffer 4 Dwords Deep**
- **CPU-to-Memory and CPU-to-PCI Write Posting Buffers Accelerate Write Performance**
- **Dual-Port Architecture Allows Concurrent Operations on the Host and PCI Buses**
- **Operates Synchronously to the CPU and PCI Clocks**
- **Supports Burst Read and Writes of Memory from the Host and PCI Buses**
- **Sequential CPU Writes to PCI Converted to Zero Wait-State PCI Bursts with Optional TRDY# Connection**
- **Byte Parity Support for the Host and Memory Buses**
  - **— Optional Parity Generation for Host to Memory Transfers**
  - **— Optional Parity Checking for the Secondary Cache**
  - **— Parity Checking for Host and PCI Memory Reads**
  - **— Parity Generation for PCI to Memory Writes**
- **160-Pin QFP Package**

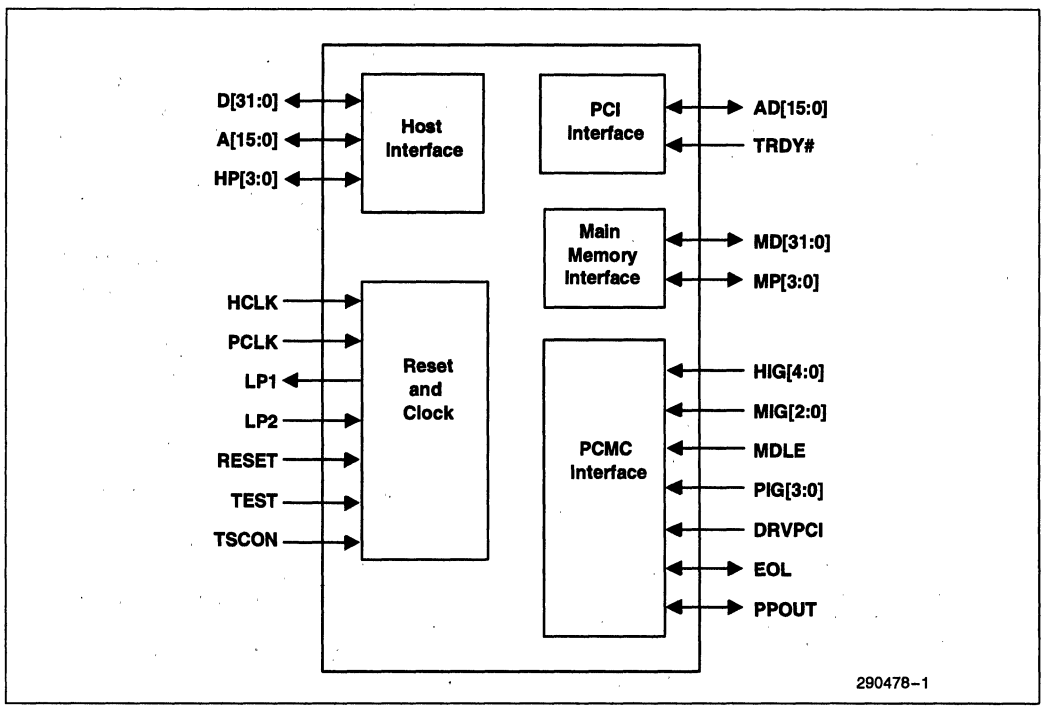
1

Two 82433LX or 82433NX Local Bus Accelerator (LBX) components provide a 64-bit data path between the host CPU/Cache and main memory, a 32-bit data path between the host CPU bus and PCI Local Bus, and a 32-bit data path between the PCI Local Bus and main memory. The dual-port architecture allows concurrent operations on the host and PCI Buses. The LBXs incorporate three write posting buffers and two read prefetch buffers to increase CPU and PCI performance. The LBX supports byte parity for the host and main memory buses. The 82433NX is intended to be used with the 82434NX PCI/Cache/Memory Controller (PCMC). The 82433LX is intended to be used with the 82434LX PCMC. During bus operations between the host, main memory and PCI, the PCMC commands the LBXs to perform functions such as latching address and data, merging data, and enabling output buffers. Together, these three components form a "Host Bridge" that provides a full function dual-port data path interface, linking the host CPU and PCI bus to main memory.

This document describes both the 82433LX and 82433NX. Shaded areas, like this one, describe the 82433NX operations that differ from the 82433LX.

The complete document for this product is available from Intel's Literature Center at 1-800-548-4725.

November 1995
Order Number: 290478-004



LBX Simplified Block Diagram

82434LX/82434NX PCI, CACHE AND MEMORY CONTROLLER (PCMC)

- Supports the Pentium™ Processor at iCOMP™ Index 510\60 MHz and iCOMP Index 567\66 MHz
- Supports the Pentium Processor at iCOMP Index 735\90 MHz, iCOMP Index 815\100 MHz, and iCOMP Index 610\75 MHz
- Supports Pipelined Addressing Capability of the Pentium Processor
- The 82430NX Drives 3.3V Signal Levels on the CPU and Cache Interfaces
- High Performance CPU/PCI/Memory Interfaces via Posted Write and Read Prefetch Buffers
- Fully Synchronous PCI Interface with Full Bus Master Capability
- Supports the Pentium Processor Internal Cache in Either Write-Through or Write-Back Mode
- Programmable Attribute Map of DOS and BIOS Regions for System Flexibility
- Integrated Low Skew Clock Driver for Distributing Host Clock
- Integrated Second Level Cache Controller
  - Integrated Cache Tag RAM
  - Write-Through and Write-Back Cache Modes for the 82434LX
  - Write-Back for the 82434NX
  - 82434NX Supports Low-Power Cache Standby
  - Direct Mapped Organization
  - Supports Standard and Burst SRAMs
  - 256-KByte and 512-KByte Sizes
  - Cache Hit Cycle of 3-1-1-1 on Reads and Writes Using Burst SRAMs
  - Cache Hit Cycle of 3-2-2-2 on Reads and 4-2-2-2 on Writes Using Standard SRAMs
- Integrated DRAM Controller
  - Supports 2 MBytes to 192 MBytes of Cacheable Main Memory for the 82434LX
  - Supports 2 MBytes to 512 MBytes of Cacheable Main Memory for the 82434NX
  - Supports DRAM Access Times of 70 ns and 60 ns
  - CPU Writes Posted to DRAM 4-1-1-1
  - Refresh Cycles Decoupled from ISA Refresh to Reduce the DRAM Access Latency
  - Six RAS# Lines (82434LX)
  - Eight RAS# Lines (82434NX)
  - Refresh by RAS#-Only, or CAS-Before-RAS#, in Single or Burst of Four
- Host/PCI Bridge
  - Translates CPU Cycles into PCI Bus Cycles
  - Translates Back-to-Back Sequential CPU Memory Writes into PCI Burst Cycles
  - Burst Mode Writes to PCI in Zero PCI Wait-States (i.e. Data Transfer Every Cycle)
  - Full Concurrency Between CPU-to-Main Memory and PCI-to-PCI Transactions
  - Full Concurrency Between CPU-to-Second Level Cache and PCI-to-Main Memory Transactions
  - Same Cache and Memory System Logic Design for ISA and EISA Systems
  - Cache Snoop Filter Ensures Data Consistency for PCI-to-Main Memory Transactions
- 208-Pin QFP Package

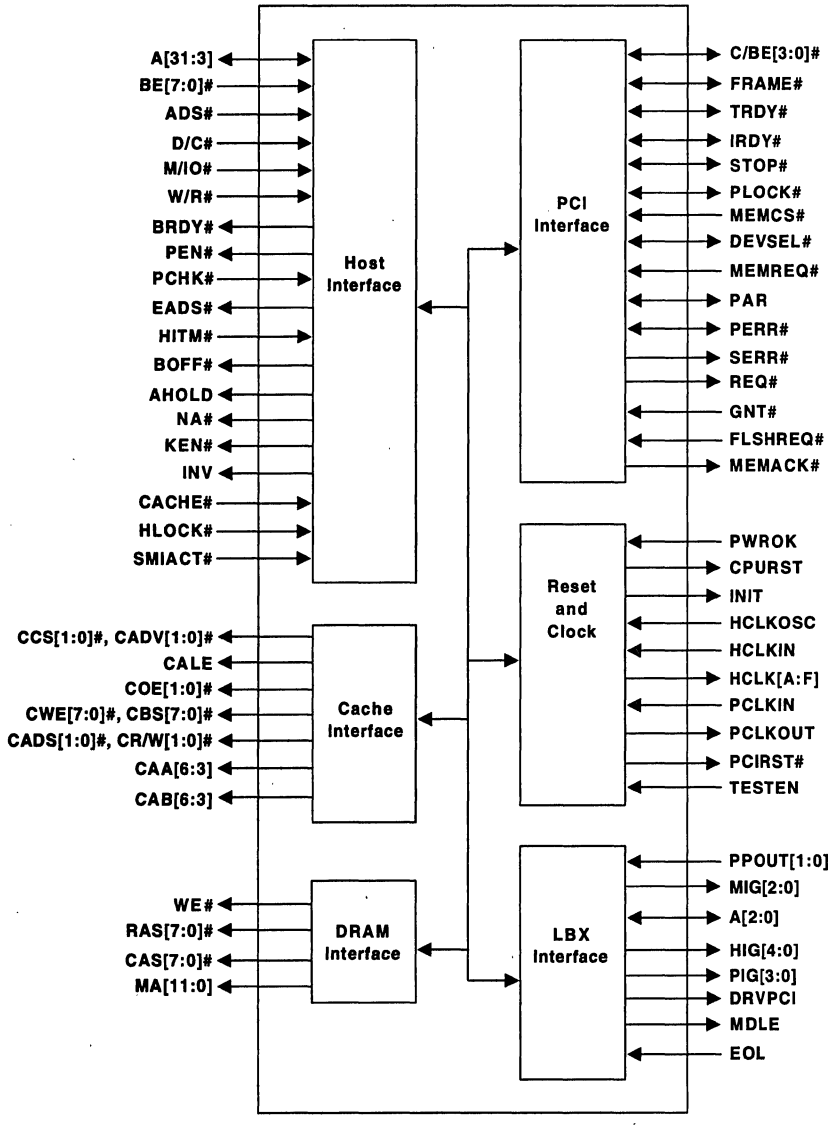
*Other brands and names are the property of their respective owners.

The complete document for this product is available from Intel's Literature Center at 1-800-548-4725.

December 1994
Order Number: 290479-004

This document describes both the 82434LX and 82434NX. Unshaded areas describe the 82434LX. Shaded areas, like this one, describe 82434NX operations that differ from the 82434LX.

The 82434LX/82434NX PCI, Cache, Memory Controllers (PCMC) integrate the cache and main memory DRAM control functions and provide bus control for transfers between the CPU, cache, main memory, and the PCI Local Bus. The cache controller supports write-back (or write-through for 82434LX) cache policy and cache sizes of 256-KBytes and 512-KBytes. The cache memory can be implemented with either standard or burst SRAMs. The PCMC cache controller integrates a high-performance Tag RAM to reduce system cost.



NOTE:
RAS[7:6]# and MA11 are only on the 82434NX. CCS[1:0] functionality is only on the 82434NX.

Simplified Block Diagram of the PCMC

82420EX PCISET DATA SHEET
82425EX PCI SYSTEM CONTROLLER (PSC) AND 82426EX ISA BRIDGE (IB)

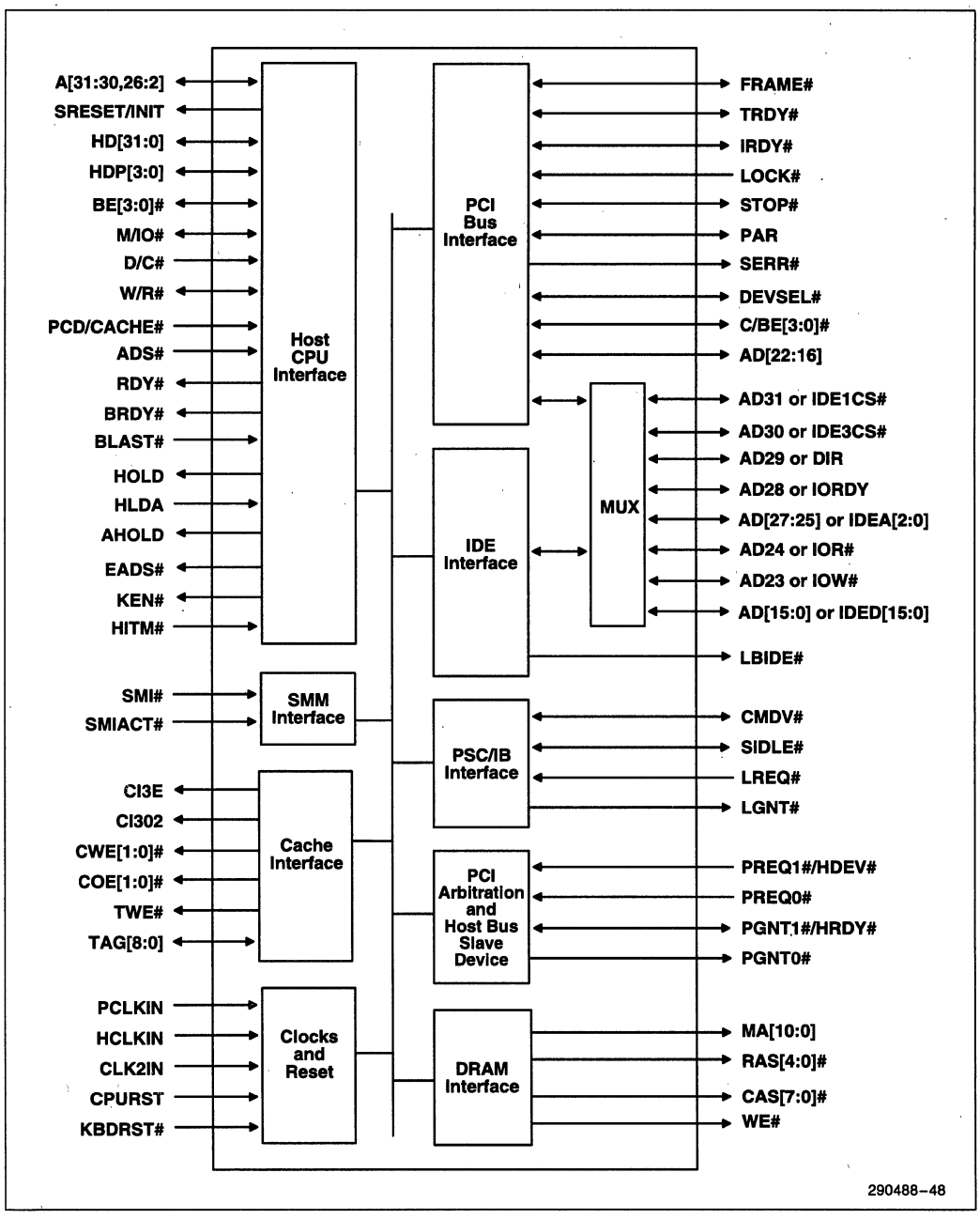
- Host CPU
  - 25–33 MHz Intel486™ and OverDrive™ Processors
  - L1 Write-Back Support
- Integrated DRAM Controller
  - 1 to 128 MByte Main Memory
  - 70 ns Fast Page Mode DRAM SIMMs Supported
  - Supports 256 KByte, 1 MByte, and 4 MByte Double and Single Sided SIMMs
  - Read Page Hit Timing of 3-2-2-2 at 33 MHz
  - Burst Mode PCI Master Accesses
  - Decoupled Refresh Reduces DRAM Latency
  - Five RAS Lines
- Integrated L2 Cache Controller
  - Write-Back and Write-Through Cache Policies
  - Direct Mapped Organization
  - 64, 128, 256 or 512 KByte Cache Sizes
  - Programmable Zero Wait-State L2 Cache Read and Write Accesses
  - Two Banks Interleaved or a Single Bank Non-Interleaved Operation
  - No VALID Bit Required
- 25/33 MHz PCI Bus Interface
  - Two Bus Masters
  - PCI Auto Configuration Support
- Host/PCI Bridge
  - Converts Back-to-Back Sequential Memory Writes to PCI Burst Writes
  - CPU Memory Write Posting to PCI
- PCI Local Bus IDE Interface
  - Supports Mode 3 Timing
- Programmable Attribute Map for First 1 MByte of Main Memory
- 100% ISA Compatible
  - Directly Drives 5 ISA Slots
- Two 8237 DMA Controllers
  - 7 DMA Channels
  - 27-bit Addressability
  - Compatible DMA Transfers
- One 82C54 Timer/Counter
  - System Timer
  - Refresh Request
  - Speaker Tone
- Two 82C59 Interrupt Controllers
  - 14 Interrupts
  - Edge/Level Sense is Programmable per Channel
  - PCI Interrupt Steering for Plug and Play Compatibility
- X-Bus Peripheral Support
  - RTC, KBC, BIOS Chip Selects
  - Control for Lower X-Bus Transceiver
  - Integrates Mouse Interrupt
  - Coprocessor Error Reporting
- Non-Maskable Interrupts (NMI)
  - PCI System Errors
  - Main Memory Parity Errors
  - ISA Parity Errors
- System Power Management (Intel SMM Support)
  - Programmable System Management Interrupt (SMI)—Hardware Events, Software Events, EXTSMI#
  - Programmable CPU Clock Control
  - Fast On/Off Mode
- Generates System Clocks
- 160-Pin QFP Package for IB
- 208-Pin QFP Package for PSC

November 1995
Order Number: 290488-004

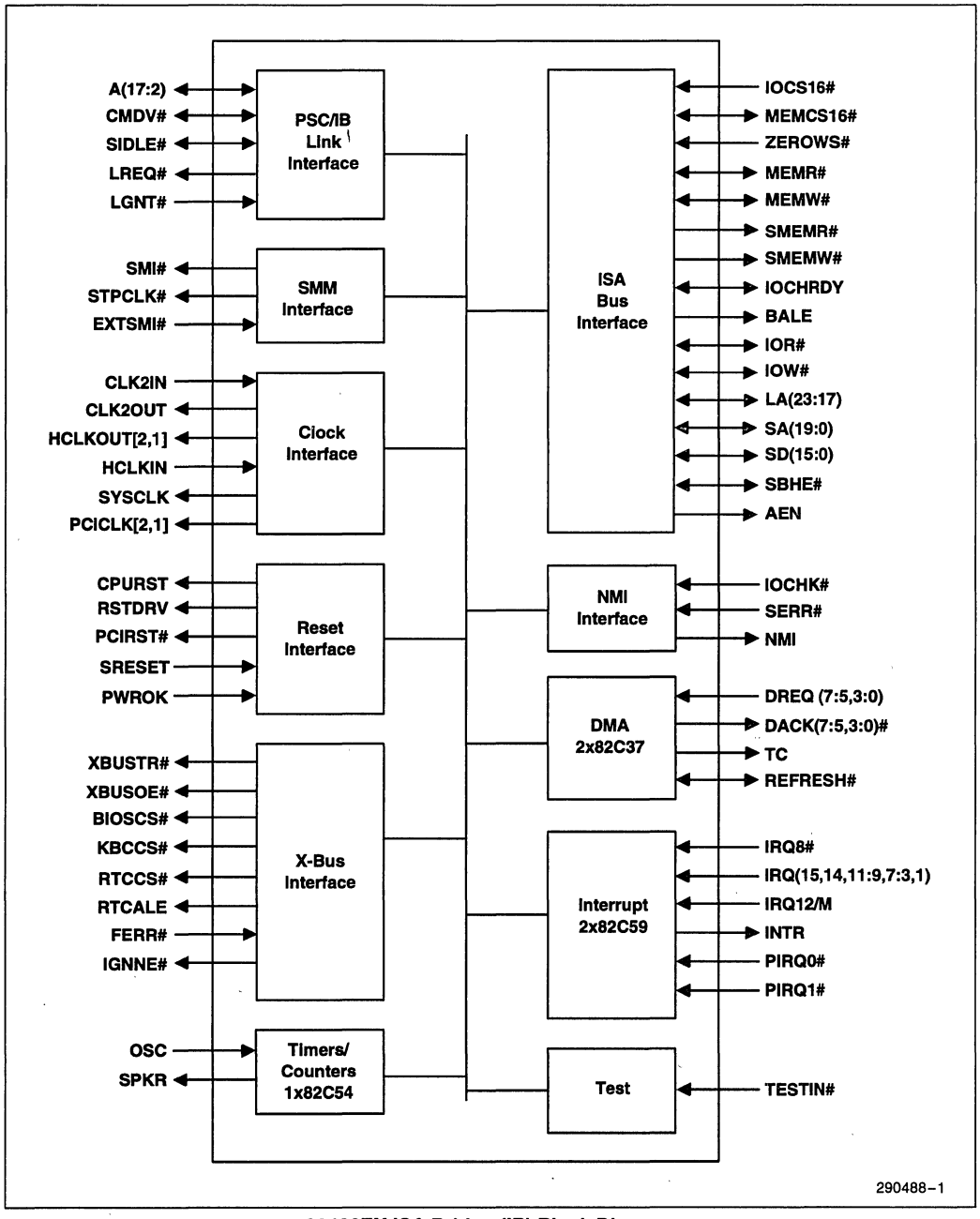
The 82420EX PCIset is the foundation for the **Value Flexible Motherboard** solution for entry-level Intel486™ processor-based PCI systems. The Value Flexible Motherboard solution, including 82420EX, Intel486 processor, 82091AA Advanced Integrated Peripherals, 82C42 Keyboard Controller, Flash BIOS, and Plug & Play software, drives PCI into the mainstream. The 82420EX PCIset is a highly integrated solution enabling low cost, small form factor motherboard designs. All Intel486 processors and upgrades are supported, including L1 write-back and Intel SMM power management. PCI Local Bus IDE is incorporated for higher performance IDE at no additional cost.

The 82420EX was designed from the ground up for PCI performance. It consists of two components—the 82425EX PCI System Controller (PSC) and the 82426EX ISA Bridge (IB). The PSC integrates the L2 cache controller and the DRAM controller. The cache controller supports both write-through and write-back cache policies and cache sizes from 64 KBytes to 512 KBytes in an interleaved or non-interleaved configuration. The DRAM controller interfaces main memory to the Host Bus and the PCI Bus. The PSC supports a two-way interleaved DRAM organization for optimum performance. Up to ten single-sided SIMMs or four double-sided and two single-sided SIMMs provide a maximum of 128 MBytes of main memory. The PSC provides memory write posting to PCI for enhanced CPU-to-PCI memory write performance. In addition, the PSC provides a high performance PCI Local Bus IDE interface.

The IB is the bridge between the ISA Bus and Host Bus, and integrates the common I/O functions found in today's ISA-based PC systems—a seven channel DMA controller, two 82C59 interrupt controllers, an 8254 timer/counter, Intel SMM power management support, and control logic for NMI generation. The IB also provides the decode for external BIOS, real time clock, and keyboard controller. Edge/Level interrupts and interrupt steering are supported for PCI plug and play compatibility. The IB integrates the ISA address and data path, reducing TTL and system cost. In addition, the integration of system clock generation logic eliminates the need for external host and PCI clock drivers.



82425EX PCI System Controller (PSC) Block Diagram



82426EX ISA Bridge (IB) Block Diagram

82420EX PCISET DATA SHEET
82425EX PCI SYSTEM CONTROLLER (PSC)
AND 82426EX ISA BRIDGE (IB)

CONTENTS

CONTENTS

CONTENTS

CONTENTS

PAGE

1

CONTENTS

CONTENTS

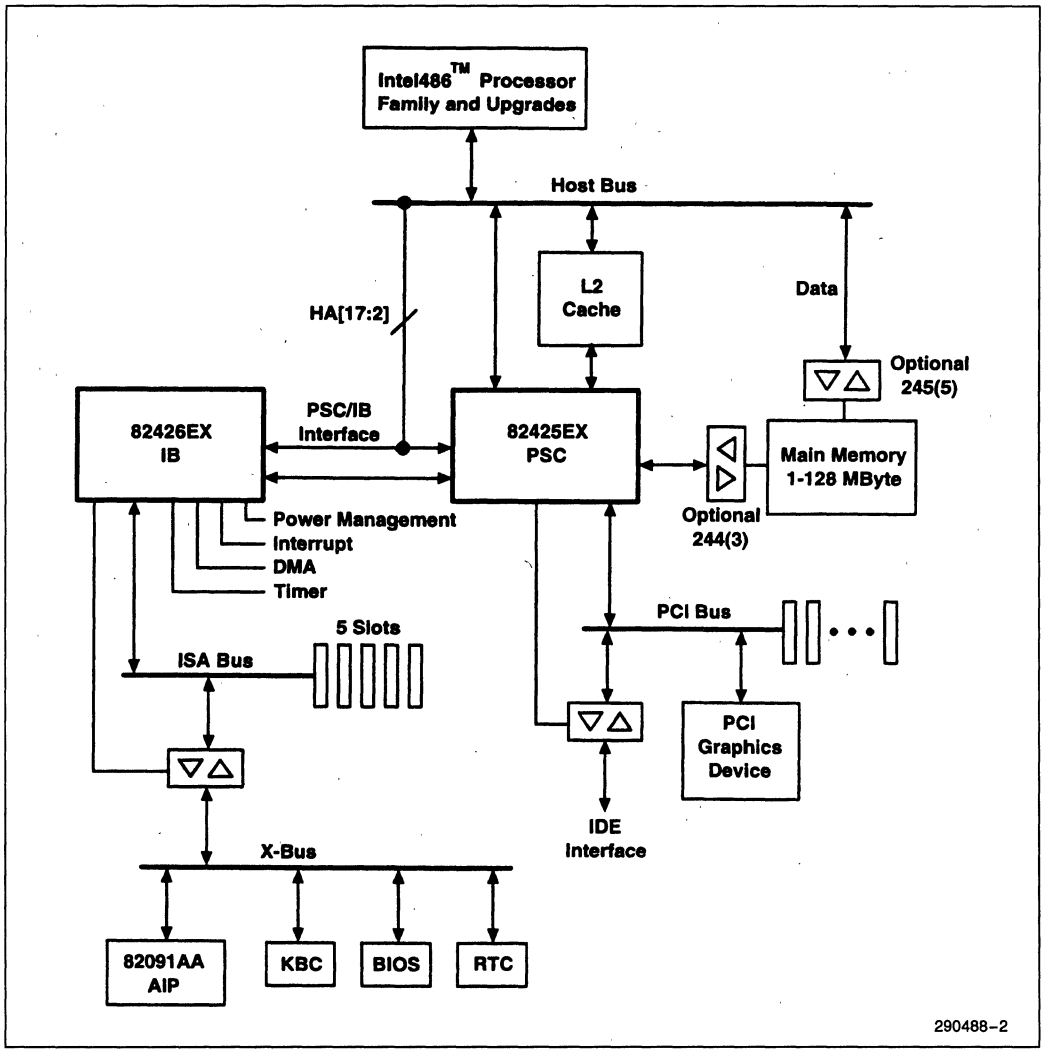


Figure 1. Example System Block Diagram

1.0 PINOUT INFORMATION

This section provides the PSC and IB pin assignment and package information. For each device, the pin assignments are listed in both alphabetical and numerical order.

1.1 PSC Pin Assignment

The PSC package is a 208-pin Quad Flatpack (QFP). Figure 2 shows the pin assignment on the package. Tables 1 and 2 list the pin assignments alphabetically and numerically, respectively.

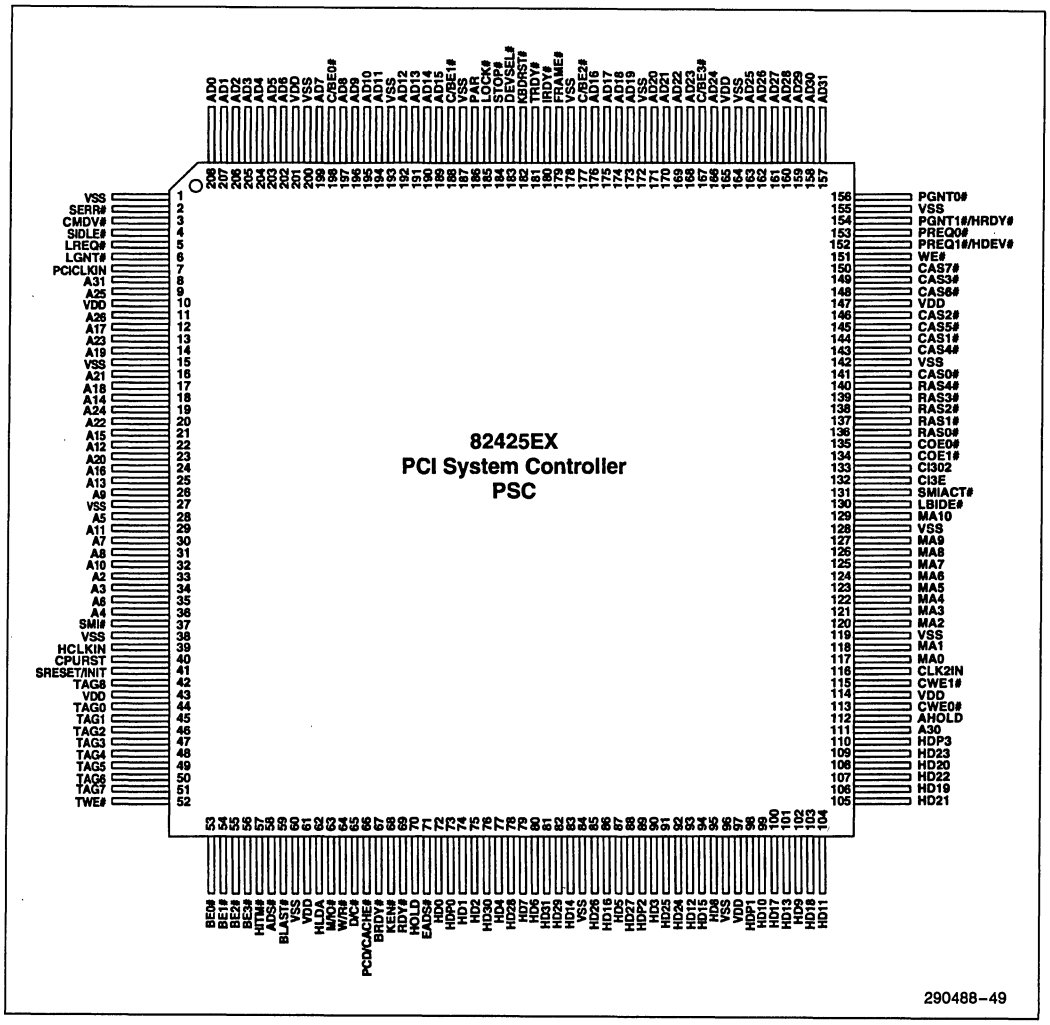


Figure 2. PSC Pin Assignment

Table 1. Alphabetical PSC Pin Assignment List

Pin Name	Pin #	I/O
A2	33	I/O
A3	34	I/O
A4	36	I/O
A5	28	I/O
A6	35	I/O
A7	30	I/O
A8	31	I/O
A9	26	I/O
A10	32	I/O
A11	29	I/O
A12	22	I/O
A13	25	I/O
A14	18	I/O
A15	21	I/O
A16	24	I/O
A17	12	I/O
A18	17	I/O
A19	14	I/O
A20	23	I/O
A21	16	I/O
A22	20	I/O
A23	13	I/O
A24	19	I/O
A25	9	I/O
A26	11	I/O
A30	111	I/O
A31	8	I/O
AD0	208	I/O
AD1	207	I/O
AD2	206	I/O
AD3	205	I/O
AD4	204	I/O

Pin Name	Pin #	I/O
AD5	203	I/O
AD6	202	I/O
AD7	199	I/O
AD8	197	I/O
AD9	196	I/O
AD10	195	I/O
AD11	194	I/O
AD12	192	I/O
AD13	191	I/O
AD14	190	I/O
AD15	189	I/O
AD16	176	I/O
AD17	175	I/O
AD18	174	I/O
AD19	173	I/O
AD20	171	I/O
AD21	170	I/O
AD22	169	I/O
AD23	168	I/O
AD24	166	I/O
AD25	163	I/O
AD26	162	I/O
AD27	161	I/O
AD28	160	I/O
AD29	159	I/O
AD30	158	I/O
AD31	157	I/O
ADS#	58	I
AHOLD	112	O
BE0#	53	I/O
BE1#	54	I/O
BE2#	55	I/O

Pin Name	Pin #	I/O
BE3#	56	I/O
BLAST#	59	I
BRDY#	67	O
C/BE0#	198	I/O
C/BE1#	188	I/O
C/BE2#	177	I/O
C/BE3#	167	I/O
CAS0#	141	O
CAS1#	144	O
CAS2#	146	O
CAS3#	149	O
CAS4#	143	O
CAS5#	145	O
CAS6#	148	O
CAS7#	150	O
CI3E	132	O
CI3O2	133	O
CLK2IN	116	I
CMDV#	3	I/O
COE0#	135	O
COE1#	134	O
CPURST	40	I
CWE0#	113	O
CWE1#	115	O
D/C#	65	I
DEVSEL#	183	s/t/s
EADS#	71	O
FRAME#	179	s/t/s
HCLKIN	39	I
HD0	72	I/O
HD1	74	I/O
HD2	75	I/O

Pin Name	Pin #	I/O
HD3	90	I/O
HD4	77	I/O
HD5	87	I/O
HD6	80	I/O
HD7	79	I/O
HD8	95	I/O
HD9	102	I/O
HD10	99	I/O
HD11	104	I/O
HD12	93	I/O
HD13	101	I/O
HD14	83	I/O
HD15	94	I/O
HD16	86	I/O
HD17	100	I/O
HD18	103	I/O
HD19	106	I/O
HD20	108	I/O
HD21	105	I/O
HD22	107	I/O
HD23	109	I/O
HD24	92	I/O
HD25	91	I/O
HD26	85	I/O
HD27	88	I/O
HD28	78	I/O
HD29	82	I/O
HD30	76	I/O
HD31	81	I/O
HDP0	73	I/O
HDP1	98	I/O
HDP2	89	I/O

Table 1. Alphabetical PSC Pin Assignment List (Continued)

Pin Name	Pin #	I/O
HDP3	110	I/O
HITM#	57	I
HLDA	62	I
HOLD	70	O
IRDY#	180	s/t/s
KBDRST#	182	I
KEN#	68	O
LBIDE#	130	O
LGNT#	6	O
LOCK#	185	I
LREQ#	5	I
M/IO#	63	I/O
MA0	117	O
MA1	118	O
MA2	120	O
MA3	121	O
MA4	122	O
MA5	123	O
MA6	124	O
MA7	125	O
MA8	126	O

Pin Name	Pin #	I/O
MA9	127	O
MA10	129	O
PAR	186	I/O
PCD/ CACHE	66	I
PCICLKIN	7	I
PGNT0#	156	O
PGNT1#/ HRDY#	154	I/O
PREQ0#	153	I
PREQ1#/ HDEV#	152	I
RAS0#	136	O
RAS1#	137	O
RAS2#	138	O
RAS3#	139	O
RAS4#	140	O
RDY#	69	O
SERR#	2	OD
SIDLE#	4	I/O
SMI#	37	I

Pin Name	Pin #	I/O
SMIACT#	131	I
SRESET/ INIT	41	O
STOP#	184	s/t/s
TAG0	44	I/O
TAG1	45	I/O
TAG2	46	I/O
TAG3	47	I/O
TAG4	48	I/O
TAG5	49	I/O
TAG6	50	I/O
TAG7	51	I/O
TAG8	42	I/O
TRDY#	181	s/t/s
TWE#	52	O
V_{DD}	10	V
V_{DD}	43	V
V_{DD}	61	V
V_{DD}	97	V
V_{DD}	147	V
V_{DD}	114	V

Pin Name	Pin #	I/O
V_{DD}	165	V
V_{DD}	201	V
V_{SS}	1	V
V_{SS}	15	V
V_{SS}	27	V
V_{SS}	38	V
V_{SS}	60	V
V_{SS}	84	V
V_{SS}	96	V
V_{SS}	119	V
V_{SS}	128	V
V_{SS}	142	V
V_{SS}	155	V
V_{SS}	164	V
V_{SS}	172	V
V_{SS}	178	V
V_{SS}	187	V
V_{SS}	193	V
V_{SS}	200	V
WE#	151	O
W/R#	64	I/O

1

Table 2. Numerical PSC Pin Assignment List

Pin Name	Pin #	I/O
V_{SS}	1	V
SERR#	2	OD
CMDV#	3	I/O
SIDLE#	4	I/O
LREQ#	5	I
LGNT#	6	O
PCICLKIN	7	I
A31	8	I/O
A25	9	I/O
V_{DD}	10	V
A26	11	I/O
A17	12	I/O
A23	13	I/O
A19	14	I/O
V_{SS}	15	V
A21	16	I/O
A18	17	I/O
A14	18	I/O
A24	19	I/O
A22	20	I/O
A15	21	I/O
A12	22	I/O
A20	23	I/O
A16	24	I/O
A13	25	I/O
A9	26	I/O
V_{SS}	27	V
A5	28	I/O
A11	29	I/O
A7	30	I/O

Pin Name	Pin #	I/O
A8	31	I/O
A10	32	I/O
A2	33	I/O
A3	34	I/O
A6	35	I/O
A4	36	I/O
SMI#	37	I
V_{SS}	38	V
HCLKIN	39	I
CPURST	40	I
SRESET/ INIT	41	O
TAG8	42	I/O
V_{DD}	43	V
TAG0	44	I/O
TAG1	45	I/O
TAG2	46	I/O
TAG3	47	I/O
TAG4	48	I/O
TAG5	49	I/O
TAG6	50	I/O
TAG7	51	I/O
TWE#	52	O
BE0#	53	I/O
BE1#	54	I/O
BE2#	55	I/O
BE3#	56	I/O
HITM#	57	I
ADS#	58	I
BLAST#	59	I
V_{SS}	60	V

Pin Name	Pin #	I/O
V_{DD}	61	V
HLDA	62	I
M/IO#	63	I/O
W/R#	64	I/O
D/C#	65	I
PCD/ CACHE#	66	I
BRDY#	67	O
KEN#	68	O
RDY#	69	O
HOLD	70	O
EADS#	71	O
HD0	72	I/O
HDP0	73	I/O
HD1	74	I/O
HD2	75	I/O
HD30	76	I/O
HD4	77	I/O
HD28	78	I/O
HD7	79	I/O
HD6	80	I/O
HD31	81	I/O
HD29	82	I/O
HD14	83	I/O
V_{SS}	84	V
HD26	85	I/O
HD16	86	I/O
HD5	87	I/O
HD27	88	I/O
HDP2	89	I/O
HD3	90	I/O

Pin Name	Pin #	I/O
HD25	91	I/O
HD24	92	I/O
HD12	93	I/O
HD15	94	I/O
HD8	95	I/O
V_{SS}	96	V
V_{DD}	97	V
HDP1	98	I/O
HD10	99	I/O
HD17	100	I/O
HD13	101	I/O
HD9	102	I/O
HD18	103	I/O
HD11	104	I/O
HD21	105	I/O
HD19	106	I/O
HD22	107	I/O
HD20	108	I/O
HD23	109	I/O
HDP3	110	I/O
A30	111	I/O
AHOLD	112	O
CWE0#	113	O
V_{DD}	114	V
CWE1#	115	O
CLK2IN	116	I
MA0	117	O
MA1	118	O
V_{SS}	119	V
MA2	120	O

Table 2. Numerical PSC Pin Assignment List (Continued)

Pin Name	Pin #	I/O
MA3	121	O
MA4	122	O
MA5	123	O
MA6	124	O
MA7	125	O
MA8	126	O
MA9	127	O
V_{SS}	128	V
MA10	129	O
LBIDE#	130	O
SMIACT#	131	I
C13E	132	O
C13O2	133	O
COE1#	134	O
COE0#	135	O
RAS0#	136	O
RAS1#	137	O
RAS2#	138	O
RAS3#	139	O
RAS4#	140	O
CAS0#	141	O
V_{SS}	142	V

Pin Name	Pin #	I/O
CAS4#	143	O
CAS1#	144	O
CAS5#	145	O
CAS2#	146	O
V_{DD}	147	V
CAS6#	148	O
CAS3#	149	O
CAS7#	150	O
WE#	151	O
PREQ1# / HDEV#	152	I
PREQ0#	153	I
PGNT1# / HRDY#	154	I/O
V_{SS}	155	V
PGNT0#	156	O
AD31	157	I/O
AD30	158	I/O
AD29	159	I/O
AD28	160	I/O
AD27	161	I/O
AD26	162	I/O
AD25	163	I/O
V_{SS}	164	V

Pin Name	Pin #	I/O
V_{DD}	165	V
AD24	166	I/O
C/BE3#	167	I/O
AD23	168	I/O
AD22	169	I/O
AD21	170	I/O
AD20	171	I/O
V_{SS}	172	V
AD19	173	I/O
AD18	174	I/O
AD17	175	I/O
AD16	176	I/O
C/BE2#	177	I/O
V_{SS}	178	V
FRAME#	179	s/t/s
IRDY#	180	s/t/s
TRDY#	181	s/t/s
KBDRST#	182	I
DEVSEL#	183	s/t/s
STOP#	184	s/t/s
LOCK#	185	I
PAR	186	I/O

Pin Name	Pin #	I/O
V_{SS}	187	V
C/BE1#	188	I/O
AD15	189	I/O
AD14	190	I/O
AD13	191	I/O
AD12	192	I/O
V_{SS}	193	V
AD11	194	I/O
AD10	195	I/O
AD9	196	I/O
AD8	197	I/O
C/BE0#	198	I/O
AD7	199	I/O
V_{SS}	200	V
V_{DD}	201	V
AD6	202	I/O
AD5	203	I/O
AD4	204	I/O
AD3	205	I/O
AD2	206	I/O
AD1	207	I/O
AD0	208	I/O

1

1.2 IB Pin Assignment

The IB package is a 160-pin Quad Flatpack (QFP). Figure 3 shows the package pin assignment. Table 3 and Table 4 list the pin assignment alphabetically and numerically, respectively.

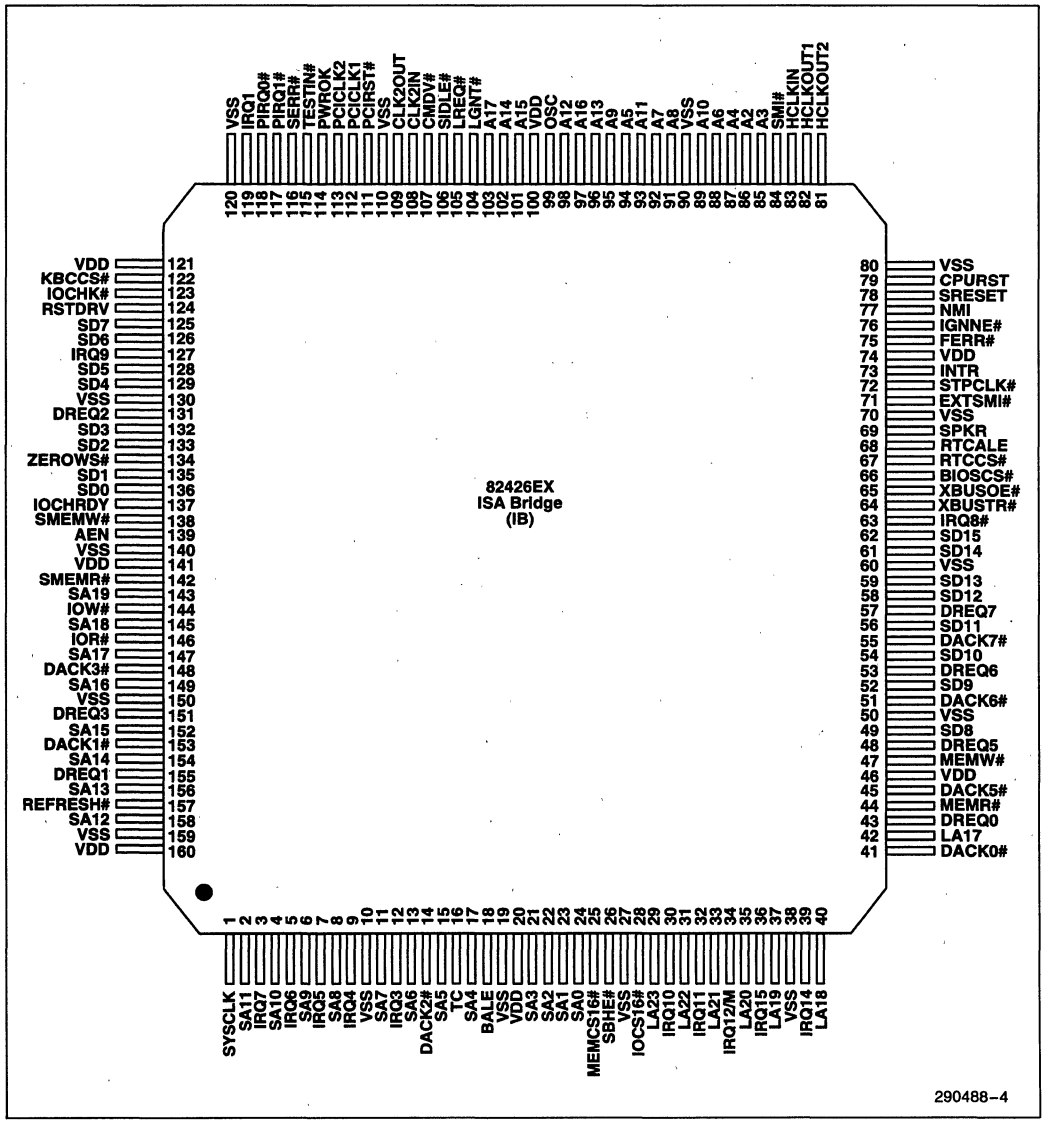


Figure 3. IB Pin Assignment

Table 3. Alphabetical IB Pin Assignment List

Pin Name	Pin #	I/O
A2	86	I/O
A3	85	I/O
A4	87	I/O
A5	94	I/O
A6	88	I/O
A7	92	I/O
A8	91	I/O
A9	95	I/O
A10	89	I/O
A11	93	I/O
A12	98	I/O
A13	96	I/O
A14	102	I/O
A15	101	I/O
A16	97	I/O
A17	103	I/O
AEN	139	O
BALE	18	O
BIOSCS#	66	O
CLK2IN	108	I
CLK2OUT	109	O
CMDV#	107	I/O
CPURST	79	O
DACK0#	41	O
DACK1#	153	O
DACK2#	14	O
DACK3#	148	O
DACK5#	45	O
DACK6#	51	O
DACK7#	55	O

Pin Name	Pin #	I/O
DREQ0	43	I
DREQ1	155	I
DREQ2	131	I
DREQ3	151	I
DREQ5	48	I
DREQ6	53	I
DREQ7	57	I
EXTSMI#	71	IS
FERR#	75	I
HCLKIN	83	I
HCLKOUT1	82	O
HCLKOUT2	81	O
IGNNE#	76	O
INTR	73	O
IOCHK#	123	I
IOCHRDY	137	I/O
IOCS16#	28	I
IOR#	146	I/O
IOW#	144	I/O
IRQ1	119	I
IRQ3	12	I
IRQ4	9	I
IRQ5	7	I
IRQ6	5	I
IRQ7	3	I
IRQ8#	63	I
IRQ9	127	I
IRQ10	30	I
IRQ11	32	I
IRQ12/M	34	I

Pin Name	Pin #	I/O
IRQ14	39	I
IRQ15	36	I
KBCCS#	122	O
LA17	42	I/O
LA18	40	I/O
LA19	37	I/O
LA20	35	I/O
LA21	33	I/O
LA22	31	I/O
LA23	29	I/O
LGNT#	104	I
LREQ#	105	O
MEMCS16#	25	I/O
MEMR#	44	I/O
MEMW#	47	I/O
NMI	77	O
OSC	99	I
PCICLK1	112	O
PCICLK2	113	O
PCIRST#	111	O
PIRQ0#	118	I
PIRQ1#	117	I
PWROK	114	IS
REFRESH#	157	I/O
RSTDRV	124	O
RTCALE	68	O
RTCCS#	67	O
SA0	24	I/O
SA1	23	I/O
SA2	22	I/O

Pin Name	Pin #	I/O
SA3	21	I/O
SA4	17	I/O
SA5	15	I/O
SA6	13	I/O
SA7	11	I/O
SA8	8	I/O
SA9	6	I/O
SA10	4	I/O
SA11	2	I/O
SA12	158	I/O
SA13	156	I/O
SA14	154	I/O
SA15	152	I/O
SA16	149	I/O
SA17	147	I/O
SA18	145	I/O
SA19	143	I/O
SBHE#	26	I/O
SD0	136	I/O
SD1	135	I/O
SD2	133	I/O
SD3	132	I/O
SD4	129	I/O
SD5	128	I/O
SD6	126	I/O
SD7	125	I/O
SD8	49	I/O
SD9	52	I/O
SD10	54	I/O
SD11	56	I/O

Table 3. Alphabetical IB Pin Assignment List (Continued)

Pin Name	Pin #	I/O
SD12	58	I/O
SD13	59	I/O
SD14	61	I/O
SD15	62	I/O
SERR#	116	I
SIDLE#	106	I/O
SMEMR#	142	O
SMEMW#	138	O
SMI#	84	O
SPKR	69	O

Pin Name	Pin #	I/O
SRESET	78	I
STPCLK#	72	O
SYSCLK	1	O
TC	16	O
TESTIN#	115	I
V_{DD}	20	V
V_{DD}	46	V
V_{DD}	74	V
V_{DD}	100	V
V_{DD}	121	V

Pin Name	Pin #	I/O
V_{DD}	141	V
V_{DD}	160	V
V_{SS}	10	V
V_{SS}	19	V
V_{SS}	27	V
V_{SS}	38	V
V_{SS}	50	V
V_{SS}	60	V
V_{SS}	70	V
V_{SS}	80	V

Pin Name	Pin #	I/O
V_{SS}	90	V
V_{SS}	110	V
V_{SS}	120	V
V_{SS}	130	V
V_{SS}	140	V
V_{SS}	150	V
V_{SS}	159	V
XBUSOE#	65	O
XBUSTR#	64	O
ZEROWS#	134	I

Table 4. Numerical IB Pin Assignment List

Pin Name	Pin #	I/O	Pin Name	Pin #	I/O	Pin Name	Pin #	I/O	Pin Name	Pin #	I/O
SYSCLK	1	O	LA22	31	I/O	SD14	61	I/O	A8	91	I/O
SA11	2	I/O	IRQ11	32	I	SD15	62	I/O	A7	92	I/O
IRQ7	3	I	LA21	33	I/O	IRQ8#	63	I	A11	93	I/O
SA10	4	I/O	IRQ12/M	34	I	XBUSTR#	64	O	A5	94	I/O
IRQ6	5	I	LA20	35	I/O	XBUSOE#	65	O	A9	95	I/O
SA9	6	I/O	IRQ15	36	I	BIOSCS#	66	O	A13	96	I/O
IRQ5	7	I	LA19	37	I/O	RTCCS#	67	O	A16	97	I/O
SA8	8	I/O	V_{SS}	38	V	RTCALE	68	O	A12	98	I/O
IRQ4	9	I	IRQ14	39	I	SPKR	69	O	OSC	99	I
V_{SS}	10	V	LA18	40	I/O	V_{SS}	70	V	V_{DD}	100	V
SA7	11	I/O	DACK0#	41	O	EXTSMI#	71	I	A15	101	I/O
IRQ3	12	I	LA17	42	I/O	STPCLK#	72	O	A14	102	I/O
SA6	13	I/O	DREQ0	43	I	INTR	73	O	A17	103	I/O
DACK2#	14	O	MEMR#	44	I/O	V_{DD}	74	V	LGNT#	104	I
SA5	15	I/O	DACK5#	45	O	FERR#	75	I	LREQ#	105	O
TC	16	O	V_{DD}	46	V	IGNNE#	76	O	SIDLE#	106	I/O
SA4	17	I/O	MEMW#	47	I/O	NMI	77	O	CMDV#	107	I/O
BALE	18	O	DREQ5	48	I	SRESET	78	I	CLK2IN	108	I
V_{SS}	19	V	SD8	49	I/O	CPURST	79	O	CLK2OUT	109	O
V_{DD}	20	V	V_{SS}	50	V	V_{SS}	80	V	V_{SS}	110	V
SA3	21	I/O	DACK6#	51	O	HCLKOUT2	81	O	PCIRST#	111	O
SA2	22	I/O	SD9	52	I/O	HCLKOUT1	82	O	PCICLK1	112	O
SA1	23	I/O	DREQ6	53	I	HCLKIN	83	I	PCICLK2	113	O
SA0	24	I/O	SD10	54	I/O	SMI#	84	O	PWROK	114	I
MEMCS16#	25	I/O	DACK7#	55	O	A3	85	I/O	TESTIN#	115	I
SBHE#	26	I/O	SD11	56	I/O	A2	86	I/O	SERR#	116	I
V_{SS}	27	V	DREQ7	57	I	A4	87	I/O	PIRQ1#	117	I
IOCS16#	28	I	SD12	58	I/O	A6	88	I/O	PIRQ0#	118	I
LA23	29	I/O	SD13	59	I/O	A10	89	I/O	IRQ1	119	I
IRQ10	30	I	V_{SS}	60	V	V_{SS}	90	V	V_{SS}	120	V

Table 4. Numerical IB Pin Assignment List (Continued)

Pin Name	Pin #	I/O
V_{DD}	121	V
KBCCS#	122	O
IOCHK#	123	I
RSTDRV	124	O
SD7	125	I/O
SD6	126	I/O
IRQ9	127	I
SD5	128	I/O
SD4	129	I/O
V_{SS}	130	V

Pin Name	Pin #	I/O
DREQ2	131	I
SD3	132	I/O
SD2	133	I/O
ZEROWS#	134	I
SD1	135	I/O
SD0	136	I/O
IOCHRDY	137	I/O
SMEMW#	138	O
AEN	139	O
V_{SS}	140	V

Pin Name	Pin #	I/O
V_{DD}	141	V
SMEMR#	142	O
SA19	143	I/O
IOW#	144	I/O
SA18	145	I/O
IOR#	146	I/O
SA17	147	I/O
DACK3#	148	O
SA16	149	I/O
V_{SS}	150	V

Pin Name	Pin #	I/O
DREQ3	151	I
SA15	152	I/O
DACK1#	153	O
SA14	154	I/O
DREQ1	155	I
SA13	156	I/O
REFRESH#	157	I/O
SA12	158	I/O
V_{SS}	159	V
V_{DD}	160	V

2.0 SIGNAL DESCRIPTION

This section contains a detailed description of each signal. The PSC signals are presented first, followed by the IB signals. The signals are arranged in functional groups according to their interface.

Note that the "#" symbol at the end of a signal name indicates that the active, or asserted state occurs when the signal is at a low voltage level. When "#" is not present after the signal name, the signal is asserted when at the high voltage level.

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

The following notations are used to describe signal types.

Signal Type	Description
I	**Input.** Standard input-only signal.
IS	**Input.** Schmitt Trigger
O	**Totem Pole Output.** Standard active driver.
OD	**Open Drain.** Input/Output
I/O	**Input/Output.** Bi-directional, tri-state pin.
s/t/s	**Sustained Tri-state.** Active low, tri-state signal with a pullup. Must be driven high for a clock before tri-state. Turn-around time must be maintained.

2.1 PSC Signals

2.1.1 HOST CPU INTERFACE SIGNALS (PSC)

Name	Type	Description
SRESET/ INIT	O	**SOFT RESET/INITIALIZE:** This is the soft reset output of the PSC and should be connected to the SRESET or INIT input to the CPU, depending on the CPU type.
A[31:30, 26:2]	I/O	**HOST ADDRESS:** A[31:30,26:2] are used as inputs to the PSC for CPU driven cycles. A[31:30,26:4] are outputs during Snoop cycles. Note that A[29:27] are not driven by the PSC. These signal lines must be externally driven low by either weak pull-down resistors or by driving these lines low when HLDA is active asserted. A[17:2] are also used for PSC/IB Link Interface transfers. These signals are tri-stated after a hard reset.
HD[31:0]	I/O	**HOST DATA:** HD[31:0] are connected to the host CPU data bus. These signals are inputs after a hard reset.
HDP[3:0]	I/O	**HOST DATA PARITY:** HP[3:0] are bi-directional parity signals for the host data bus. These signals provide parity to the PSC during main memory read cycles. The PSC sends parity information to main memory during non-CPU main memory write cycles. These signals are tri-stated after a hard reset.
BE[3:0]#	I/O	**BYTE ENABLE:** The Byte Enable signals indicate active bytes during read and write cycles. These signals are tri-stated after a hard reset.
M/IO# D/C# W/R#	I/O I I/O	**BUS CYCLE DEFINITION (Memory/Input-Output, Data/Code, Write/Read:** These signals define the Host Bus cycle. Note that special cycles are identified by BE[3:0]# and A[4:2]. These signals are tri-stated after a hard reset.

2.1.1 HOST CPU INTERFACE SIGNALS (PSC) (Continued)

Name	Type	Description
PCD/ CACHE#	I	**PAGE CACHE DISABLE/CACHE:** This multiplexed signal pin has two functions, depending on the type of CPU used. The **PCD** input signal, when asserted, indicates the current cycle can not be cached in the L2 cache during a cache line fill operation. When PCD is asserted the line will not be cached in L1 or L2. The **CACHE#** signal is active along with the first ADS# until the first RDY# or BRDY#. For line fills, the functionality of the CACHE# signal is identical to that of the PCD signal. During write-back cycles, CACHE# is always asserted at the beginning of the line write-back. The beginning of a write-back cycle is uniquely identified by active ADS#, W/R# and CACHE#. Beginning of the snoop write-back is identified by the ADS#, W/R#, CACHE# and HITM# being active.
ADS#	I	**ADDRESS STATUS:** The ADS# input indicates that the bus cycle definition signals (M/IO#, D/C#, W/R#), BE[3:0]#, and A[31:30, 26:2] are available on their corresponding pins.
RDY#	O	**READY:** RDY# indicates that the current non-burst bus cycle is complete. This signal is negated after a hard reset.
BRDY#	O	**BURST READY:** BRDY performs the same function during a burst cycle that RDY# performs during a non-burst cycle. This signal is negated after hard reset.
BLAST#	I	**BURST LAST:** BLAST# indicates the end of a burst access for CPU-initiated cycles.
HOLD	O	**HOLD:** The PSC asserts HOLD to the CPU to request ownership of the Host Bus. This signal is negated after a hard reset.
HLDA	I	**HOLDA:** HLDA must be asserted by the CPU for the PSC to grant a new master on the PCI or ISA Buses. When HLDA is negated, the CPU is the Host Bus master and the PSC is the PCI Bus master. When HLDA is negated, the PSC is also the master on the PSC/IB link interface.
AHOLD	O	**ADDRESS HOLD:** The AHOLD output signal forces the CPU to float its address bus in the next clock. The PSC asserts this signal in preparation to perform a PSC/IB Interface transfer, when SRESET needs to be asserted, or upon Deturbo logic requests. This signal is negated after a hard reset.
EADS#	O	**EXTERNAL ADDRESS:** EADS#, when asserted, indicates that an external address has been driven onto the CPU address lines. This address is used to perform an internal cache snoop cycle. This signal is negated after a hard reset.
KEN#	O	**CACHE ENABLE:** KEN#, when asserted, indicates whether the current cycle is cacheable in the CPU internal (L1 or primary) cache. This signal is negated after a hard reset.
HITM#	I	**HIT MODIFIED:** HITM#, when asserted, indicates that a hit to a modified data cache has occurred during the snoop cycle. A pull-up is used to keep HITM# negated, when not used.

2.1.2 SECONDARY CACHE SIGNALS (PSC)

Name	Type	Description
CI3E CI3O2	O	**CACHE INDEX SIGNALS:** The Cache Index signals generate the burst sequence required by the CPU during secondary cache accesses. The PSC latches the starting burst address and internally generates subsequent dword addresses for the entire cache line. The CI3E signal is always used for cache index bit 3. When used in a bank interleaved configuration, CI3O2 is used to drive cache index bit 3 to the odd bank, and CI3E is used to drive bit 3 to the even bank. When used in non-interleaved mode (only one bank), CI3O2 is used to drive cache index bit 2.
CWE[1:0]#	O	**CACHE WRITE ENABLE:** CWE[1:0]# are used to enable single writes and line fills to be written into the L2 cache. CWE0# is driven to all SRAMs in the even bank and CWE1# is driven to all SRAMs in the odd bank. The chip select signals of the SRAMs are asserted based on the byte enable signals and the W/R# signal, which are gated externally. Thus, only the bytes selected by the CPU are written. When programmed for non-interleaved mode, CWE[1:0]# mirror each other and are asserted to both banks. These signals are negated after a hard reset.
COE[1:0]#	O	**CACHE OUTPUT ENABLE:** COE[1:0]# are used to perform read cycles from the cache data SRAMs. COE0# is connected to the output enable pins of the cache data SRAMs of the even bank. COE1# is connected to the output enable pins of the cache data SRAMs of the odd bank. When programmed for non-interleaved mode, COE[1:0]# mirror each other and are asserted to both banks. These signals are negated after a hard reset.
TWE#	O	**TAG WRITE ENABLE:** TWE# is connected to the tag SRAM write enable (WE#) pin. TWE# is asserted during CPU read-miss cycles when a cache line is allocated and during write-hit cycles, when the Dirty (Modified) bit of the tag is updated. This signal is negated after a hard reset.
TAG[8:0]	I/O	**CACHE TAG:** TAG[8:0] are directly connected to the tag SRAM data bus. The L2 cache size determines the relationship between TAG[7:0] and the A[26:16] host address signals (see Section 4.0, Functional Description). TAG8 is always used as the Dirty (Modified) bit for the write-back L2 cache.

2.1.3 PCI SIGNALS (PSC)

Name	Type	Description
AD[31:0]	I/O	**ADDRESS/DATA:** AD[31:0] are connected to the PCI multiplexed address/data bus. These signals are also multiplexed with the IDE interface (refer to the section on PCI Bus IDE Signals). These signals are driven high after a hard reset.
C/BE[3:0]#	I/O	**BUS COMMAND/BYTE ENABLE:** PCI Bus commands (C) and Byte Enables (BE[3:0]#) are multiplexed on the same pins. PCI local bus command encoding and types are listed in Section 4.0, Functional Description. These signals are driven high after a hard reset.
FRAME#	I/O s/t/s	**FRAME:** FRAME# is an output when the PSC is a master on the PCI bus. FRAME indicates that a PCI cycle has started. This signal is tri-stated after a hard reset.
TRDY#	I/O s/t/s	**TARGET READY:** TRDY# is an input when PSC is a master on the PCI Bus. TRDY# is an output when the PSC acts as a PCI slave. TRDY# indicates that the target device is ready. This signal is tri-stated after a hard reset.
IRDY#	I/O s/t/s	**INITIATOR READY:** IRDY# is an output when PSC is a PCI master. IRDY# is an input when the PSC is a PCI slave. IRDY# indicates that the initiator of the cycle is ready. This signal is tri-stated after a hard reset.
LOCK#	I	**LOCK:** LOCK# indicates an exclusive bus operation and may require multiple transactions to complete. The PSC supports a bus type of LOCK only. Thus, when a PCI master locks the PCI Bus, it owns the system for the duration of the locked transactions.
STOP#	I/O s/t/s	**STOP:** STOP# indicates that the current bus target is requesting the master to stop the current transaction. STOP# is used for disconnect, retry, and abort sequences on the PCI Bus. This signal is tri-stated after a hard reset.
PAR	I/O	**PARITY:** PAR is driven by the PSC, as a PCI master, during the address and data phases for a write cycle and during the address phase for a read cycle. When the PSC is a PCI slave, parity is driven by the PSC for the data phase of a PCI read cycle. Parity is even parity across AD[31:0] and C/BE[3:0]#. PAR lags the corresponding address or data phase by 1 PCICLK. This signal is asserted after a hard reset.
SERR#	OC	**SYSTEM ERROR:** SERR#, when driven by the PSC, indicates that either a main memory parity error occurred or the PSC, as a master, received a target abort.
DEVSEL#	I/O s/t/s	**DEVICE SELECT:** DEVSEL#, when asserted, indicates that a PCI slave device has decoded the bus cycle address as the target of the current access. The PSC drives DEVSEL# based on the main memory address range being accessed by a PCI master. As an input, DEVSEL# indicates whether any device on the bus has been selected. This signal is tri-stated after a hard reset.

2.1.4 SYSTEM POWER MANAGEMENT (SMM) SIGNALS (PSC)

Name	Type	Description
SMI#	I	**SYSTEM MANAGEMENT INTERRUPT:** SMI#, when asserted, indicates that there is an active SMI#. It is used, along with SMIACT#, to block SRESET.
SMIACT#	I	**SYSTEM MANAGEMENT INTERRUPT ACTIVE:** SMIACT# indicates that the system is running in system management mode. SMIACT# is used by the PSC to access the SMRAM for CPU-initiated cycles. While SMIACT# is active, SRESET and A20M# must be blocked.

2.1.5 DRAM CONTROL SIGNALS (PSC)

Name	Type	Description
MA[10:0]	O	**MULTIPLEXED DRAM ADDRESS:** The MA[10:0] bus provides row and column address information to the main memory DRAMs.
RAS[4:0]#	O	**ROW ADDRESS STROBE:** Each of the RAS[4:0]# output signals corresponds to one DRAM row of four or eight bytes. These signals are used to latch the row addresses on the MA[10:0] bus into the DRAMs. RAS[4:0]# drive the DRAMs directly, without any external buffers. These signals are negated after a hard reset.
CAS[7:0]#	O	**COLUMN ADDRESS STROBE:** CAS[7:0]# are used to latch the column addresses on the MA[10:0] bus into the DRAMs. CAS[7:0]# drive the DRAMs directly, without any external buffers. These signals are negated after a hard reset.
WE#	O	**DRAM WRITE ENABLE:** WE# is externally buffered when MA[10:0] are externally buffered. This signal is asserted after a hard reset.

2.1.6 PSC/IB LINK INTERFACE (PSC)

See Section 2.2.7, PSC/IB Link Interface Signals (IB).

2.1.7 PCI BUS ARBITRATION/HOST BUS SLAVE DEVICE (PSC)

Name	Type	Description
PREQ1#/ HDEV#	I	**REQUEST1/HOST DEVICE:** This multiplexed signal has two functions. **PREQ1#** is used by the PCI master to gain control of the PCI Bus. This signal can be externally cascaded to support multiple PCI masters. The **HDEV#** function is used when the PSC is programmed to support a Host Bus slave device.
PREQ0#	I	**REQUEST0:** PREQ0# is used by PCI master to gain control of the PCI Bus. This signal can be externally cascaded to support multiple PCI masters.
PGNT1# HRDY#	I/O	**GRANT1/HOST READY: PGNT1#** is driven by the PSC to grant control of the PCI Bus to a PCI master. PGNT1# can be externally cascaded to support multiple PCI masters. The **HRDY#** function is used when the PSC is programmed to support a Host Bus slave device. This signal is driven high during and after a hard reset.
PGNT0#	O	**GRANT0:** PGNT0# is driven by the PSC to grant control of the PCI Bus to a PCI master. PGNT0# can be externally cascaded to support multiple PCI masters. This signal is driven high during and after a hard reset.

2.1.8 PCI BUS IDE (PSC)

LBIDE# is the only signal dedicated to PCI Bus IDE support. This pin is used to control the output enable of the 245 data transceivers and 244 control signal buffer. The other signals that support the IDE are shared with the PCI AD lines.

PCI Signal ADName	PCI IDE Signal Name	Type	Description
NA	LBIDE#	O	**LOCAL BUS IDE:** LBIDE# controls the output enables of the data transceivers and control signal buffers during accesses to the PCI local bus IDE path.
AD31	IDE1CS#	O	**IDE 1XX CHIP SELECT:** When the primary PCI local bus IDE is enabled, this signal is asserted for accesses to I/O addresses 1F0–1F7h. When the secondary PCI local bus IDE is enabled, this signal is asserted for an I/O cycle to addresses 170–177h.
AD30	IDE3CS#	O	**IDE 3XX CHIP SELECT:** When the primary PCI local bus IDE is enabled, this signal is asserted for accesses to I/O addresses 3F6,3F7h. When the secondary PCI local bus IDE is enabled this signal is asserted for accesses to I/O addresses 376,377h.
AD29	DIR	O	**DIRECTION:** DIR controls the direction of the data transceivers connected to the IDE connector. This signal is driven high for IDE reads and low for IDE writes.
AD28	IORDY	I	**IO READY:** IORDY allows the IDE drive to extend the cycle. The IDE cycle is held in wait-states as long as IORDY is sampled low. This input is synchronous to the first sample point, but asynchronous to subsequent sample points.
AD(27:25)	IDEA(2:0)	O	**IDE ADDRESS [2:0]:** These outputs are the A[2:0] signals that select individual ports on the IDE drive.
AD24	IOR#	O	**I/O READ STROBE:** IOR# is asserted for PCI local bus IDE I/O read cycles.
AD23	IOW#	O	**I/O WRITE STROBE:** IOW# is asserted for PCI local bus IDE I/O write cycles.
AD(15:0)	IDED(15:0)	I/O	**IDE DATA:** These bi-directional signals output data during IDE write cycles and input data during PCI local bus IDE read cycles.

2.1.9 CLOCKS AND RESET (PSC)

Name	Type	Description
PCICLKIN	I	**PCI CLOCK INPUT:** PCICLKIN is the clock input used by the PCI interface. PCI clock frequency can be configured to be the same or half the Host Bus frequency.
HCLKIN	I	**HOST BUS CLOCK INPUT:** HCLKIN is used for the Host Bus, L2 cache, PSC/IB Link and DRAM interfaces. Host Bus frequency can be configured to be the same or twice the PCI clock frequency.
CLK2IN	I	**CLOCK 2 INPUT:** CLK2IN is a 2X clock that is used during L2 cache writes.
CPURST	I	**CPU RESET:** CPURST is used as an input to place the PSC in a known state. CPURST is driven by the IB when PWROK is negated or when hard reset is driven by software through the TRC Register.
KBDRST#	I	**KEYBOARD RESET:** This signal is an input from the keyboard controller and is used to generate a soft reset to the CPU.

2.2 IB Signals

2.2.1 ISA INTERFACE SIGNALS (IB)

Name	Type	Description
BALE	O	**BUS ADDRESS LATCH ENABLE:** BALE is asserted by the IB to indicate that the address (SA[19:0], LA[23:17]) and SBHE# signal lines are valid. This signal is negated after a hard reset.
AEN	O	**ADDRESS ENABLE:** AEN is asserted during DMA cycles to prevent I/O slaves from mis-interpreting DMA cycles as valid I/O cycles. This signal is also asserted during IB-initiated refresh cycles. This signal is negated after a hard reset.
SYSCLK	O	**SYSTEM CLOCK:** Refer to the Clock signal descriptions.
IOCHRDY	I/O	**I/O CHANNEL READY:** Resources on the ISA Bus negate IOCHRDY to indicate that additional time (wait-states) is required to complete the cycle. This signal is normally high on the ISA Bus. IOCHRDY is an input when the IB owns the ISA Bus and the CPU or a PCI agent is accessing an ISA slave or during DMA transfers. IOCHRDY is output when an external ISA Bus master owns the ISA Bus and is accessing main memory or an IB register. As an IB output, IOCHRDY is negated from the falling edge of the ISA commands. After data is available for an ISA master read or the IB latches the data for a write cycle, IOCHRDY is asserted for 70 ns. After 70 ns, the IB tri-states IOCHRDY. The IB does not drive this signal when an ISA Bus master is accessing an ISA Bus slave. IOCHRDY is tri-stated upon CPURST.
IOCS16#	I	**16-BIT I/O CHIP SELECT:** This signal is driven by I/O devices on the ISA Bus to indicate that they support 16-bit I/O bus cycles.
IOCHK#	I	**I/O CHANNEL CHECK:** IOCHK# can be driven by any resource on the ISA Bus. When asserted, it indicates that a parity or an uncorrectable error has occurred for a device or memory on the ISA Bus. If IOCHK# is asserted and NMIs are enabled (via the NMISC and NMIERTC Registers), an NMI is generated to the CPU.

2.2.1 ISA INTERFACE SIGNALS (IB) (Continued)

Name	Type	Description
IOR#	I/O	**I/O READ:** IOR# asserted indicates to an ISA I/O slave device that the slave may drive data on to the ISA data bus (SD[15:0]). The I/O slave device must hold the data valid until after IOR# is negated. IOR# is an output when the IB owns the ISA Bus. IOR# is an input when an external ISA master owns the ISA Bus. This signal is negated after a hard reset.
IOW#	I/O	**I/O WRITE:** IOW# asserted indicates to an ISA I/O slave device that the slave may latch data from the ISA data bus (SD[15:0]). IOW# is an output when the IB owns the ISA Bus. IOW# is an input when an external ISA master owns the ISA Bus. This signal is negated after a hard reset.
LA[23:17]	I/O	**UNLATCHED ADDRESS:** These bi-directional address lines allow accesses to physical memory on the ISA Bus up to 16 MBytes. LA[23:17] are outputs when the IB owns the ISA Bus. The LA[23:17] lines become inputs when an ISA master owns the ISA Bus. The LA[23:17] signals are driven to an unknown state after a hard reset.
SA[19:0]	I/O	**SYSTEM ADDRESS BUS:** SA[19:0] are outputs when the IB owns the ISA Bus. SA[19:0] are inputs when an external ISA master owns the ISA Bus. Note that SA[19:17] have the same values as LA[19:17] for all memory cycles. For I/O accesses, only SA[15:0] are used. SA[19:0] are driven to an unknown state after a hard reset.
SBHE#	I/O	**SYSTEM BYTE HIGH ENABLE:** SBHE# indicates, when asserted, that a byte is being transferred on the upper byte (SD[15:8]) of the data bus. SBHE# is negated during refresh cycles. SBHE# is an output when the IB owns the ISA Bus and an input when an external ISA master owns the ISA Bus. This signal is at an unknown state after a hard reset .
MEMCS16#	OD	**MEMORY CHIP SELECT 16:** ISA slaves that are 16-bit memory devices drive this signal low. MEMCS16# is an input when the IB owns the ISA Bus. MEMCS16# is an output when an ISA Bus master owns the ISA Bus. The IB drives this signal low during ISA master to main memory cycles.
MEMR#	I/O	**MEMORY READ:** MEMR# is the command to a memory slave that it may drive data onto the ISA data bus. MEMR# is an output when the IB is a master on the ISA Bus and an input when an ISA master, other than the IB, owns the ISA Bus. This signal is also driven by the IB during refresh cycles. For DMA cycles, the IB, as a master, asserts MEMR#. This signal is tri-stated after a hard reset.
MEMW#	I/O	**MEMORY WRITE:** MEMW# is the command to a memory slave that it may latch data from the ISA data bus. MEMW# is an output when the IB owns the ISA Bus and an input when an ISA master, other than the IB, owns the ISA Bus. For DMA cycles, the IB, as a master, asserts MEMW#. This signal is tri-stated after a hard reset.
SMEMR#	O	**STANDARD MEMORY READ:** The IB asserts SMEMR# to request an ISA memory slave to drive data onto the data lines. If the access is below the 1 MByte range (00000000–000FFFFFh) during DMA compatible, IB master, or ISA master cycles, the IB asserts SMEMR#. SMEMR# is a delayed version of MEMR#. This signal is negated after a hard reset.
SMEMW#	O	**STANDARD MEMORY WRITE:** The IB asserts SMEMW# to request an ISA memory slave to accept data from the data lines. If the access is below the 1 MByte range (00000000–000FFFFFh) during DMA compatible, IB master, or ISA master cycles, the IB asserts SMEMW#. SMEMW# is a delayed version of MEMW#. This signal is negated after a hard reset.

2.2.1 ISA INTERFACE SIGNALS (IB) (Continued)

Name	Type	Description
ZEROWS#	I	**ZERO WAIT-STATES:** An ISA slave asserts ZEROWS# after its address and command signals have been decoded to indicate that the current cycle can be shortened. If IOCHRDY is negated and ZEROWS# is asserted during the same clock, then ZEROWS# is ignored and wait-states are added as a function of IOCHRDY (i.e., IOCHRDY has precedence over ZEROWS#).
SD[15:0]	I/O	**SYSTEM DATA:** SD[15:0] provide the 16-bit data path for devices residing on the ISA Bus. SD[15:8] correspond to the high order byte and SD[7:0] correspond to the low order byte. SD[15:0] are undefined during refresh. These signals are tri-stated after a hard reset.

2.2.2 NMI SIGNALS (IB)

Name	Type	Description
NMI	O	**NON-MASKABLE INTERRUPT:** NMI is used to force a non-maskable interrupt to the CPU. The IB generates an NMI when either SERR# or IOCHK# is asserted, depending on how the NMI Status and Control Register is programmed. NMI generation can be globally disabled. This signal is negated after a hard reset.
SERR#	I	**SYSTEM ERROR:** SERR# can be pulsed active by any PCI device that detects a system error condition. Upon sampling SERR# active, the IB generates a non-maskable interrupt (NMI) to the CPU.

2.2.3 DMA SIGNALS (IB)

Name	Type	Description
DREQ [3:0,7:5]	I	**DMA REQUEST:** The DREQ lines are used to request DMA service from the IB's DMA controller or for a 16-bit master to gain control of the ISA expansion bus. The active level (high or low) is programmed via the DMA Command Register. All inactive to active edges of DREQ are assumed to be asynchronous. The request must remain active until the appropriate DACKx# signal is asserted.
DACK# [3:0, 7:5]	O	**DMA ACKNOWLEDGE:** The DACK output lines indicate that a request for DMA service has been granted by the IB or that a 16-bit master has been granted the bus. The active level (high or low) is programmed via the DMA Command Register. These signals are negated after a hard reset.
TC	O	**TERMINAL COUNT:** The IB asserts TC to DMA slaves as a terminal count indicator. This signal is negated after a hard reset.
REFRESH#	I/O	**REFRESH:** As an output, REFRESH# asserted indicates when a refresh cycle is in progress. As an output, this signal is driven directly onto the ISA Bus. This signal is an output only when the IB DMA refresh controller is a master on the bus responding to an internally generated request for refresh. As an input, REFRESH# is driven by 16-bit ISA Bus masters to initiate refresh cycles. This signal is tri-stated after a hard reset.

2.2.4 TIMER/COUNTER SIGNALS (IB)

Name	Type	Description
SPKR	O	**SPEAKER DRIVE:** This signal drives an external speaker driver device, which in turn drives the ISA system speaker. This signal can be enabled/disabled via the NMI Status and Control Register. When enabled, the SPKR signal is the output of counter 2. This signal is negated after a hard reset.
OSC	I	**OSCILLATOR:** The oscillator is the 14.31818 MHz ISA clock signal. It is used by the internal 82C54 Timer, counters 0, 1, and 2.

2.2.5 INTERRUPT CONTROLLER SIGNALS (IB)

Name	Type	Description
IRQ[15,14, 11:9,7:3,1]	IS	**INTERRUPT REQUEST:** The IRQ signals provide both system board components and ISA Bus I/O devices with a mechanism for asynchronously interrupting the CPU. The assertion mode (edge or level triggered) is selected via the Edge/Level Triggered 1 Register and Edge/Level Triggered 2 Register. For edge triggered mode, a low-to-high transition on the IRQ line is recognized as an interrupt request. For level triggered mode, a low level on the IRQ line is recognized as an interrupt request. IRQ[8#,2:0] and the internal IRQ13 (FERR#) are not programmable through the ELCR registers. These IRQ's, with the exception of IRQ8#, are always active high edge triggered and can not be modified by software. IRQ8# is always active low edge sensitive. A low-to-high transition on IRQ1 is latched by the IB. The IB continues to generate an internal IRQ1 to the 8259 core until a CPURST or an I/O read access to port 60h is detected. These signals are placed in edge triggered mode after a hard reset.
IRQ8#	IS	**INTERRUPT REQUEST 8:** IRQ8# is always an active low edge triggered interrupt input (i.e. this interrupt can not be modified by software).
IRQ12/M	IS	**INTERRUPT REQUEST 12/MOUSE INTERRUPT:** Refer to the X-Bus Signal Description.
PIRQ[0,1]#	IS	**PROGRAMMABLE INTERRUPT REQUEST:** The PIRQ0 and PIRQ1 signals can be shared with interrupts IRQ[15,14,12:9,7:3]. The routing is controlled by the PIRQ Route Control Registers. Each PIRQx# line has a separate Route Control Register. These signals require external pull-up resistors.
INTR	O	**CPU INTERRUPT:** The IB asserts INTR to signal the CPU that an interrupt request is pending and needs to be serviced. It is asynchronous with respect to SYSCLK or HCLKIN. The interrupt controller must be programmed following reset to ensure that INTR is at a known state. This signal is negated after a hard reset.

2.2.6 X-BUS SIGNALS (IB)

Name	Type	Description
XBUSTR#	O	**X-BUS DATA TRANSMIT/RECEIVE:** XBUSTR# is tied directly to the direction control of a 74F245 that buffers the X-Bus data (XD[7:0]). XBUSTR# is asserted for all I/O read cycles, regardless if the access is to an IB supported device. XBUSTR# is asserted for memory cycles only if BIOS space has been decoded. This signal is negated after a hard reset.
XBUSOE#	O	**X-BUS OUTPUT ENABLE:** XBUSOE# is tied directly to the output enable of a 74F245 that buffers the X-Bus data (XD[7:0]), from the system data bus, SD[7:0]. XBUSOE# is asserted anytime an IB supported X-Bus device is decoded, and the devices decode is enabled in the X-Bus Chip Select Enable Register (XBCSA Register). This signal is negated after a hard reset.
KBCCS#	O	**KEYBOARD CONTROLLER CHIP SELECT:** This signal is asserted during read or write accesses to KBC locations 60h, 62h, 64h, or 66h. This signal is negated after a hard reset.
BIOSCS#	O	**BIOS CHIP SELECT:** This signal is asserted during read or write accesses to BIOS. During DMA cycles, BIOSCS# is not generated. This signal is negated after a hard reset.
RTCCS#	O	**REAL TIME CLOCK CHIP SELECT:** This signal is asserted during read or write accesses to RTC location 71h. RTCALE can be tied to a pair of external OR gates to generate the real time clock read and write command signals. This signal is negated after a hard reset.
RTCALE	O	**REAL TIME CLOCK ADDRESS LATCH:** RTCALE latches the appropriate memory address into the RTC. A write to port 70h with the appropriate RTC memory address that will be written to or read from, causes RTCALE to be asserted. RTCALE is asserted based on IOW# falling, and remains asserted for two SYSCLKs. This signal is negated after a hard reset.
IGNNE#	O	**IGNORE ERROR:** This signal is connected to the ignore error pin on the CPU. IGNNE# is only used if the IB coprocessor error reporting function is enabled in the XBCSA Register (bit 5 = 1). This signal is negated after a hard reset.
FERR#	IS	**NUMERIC COPROCESSOR ERROR:** This pin functions as a FERR# signal supporting coprocessor errors. This signal is tied to the coprocessor error signal on the CPU. FERR# is only used if the IB coprocessor error reporting function is enabled in the XBCSA Register (bit 5 = 1). FERR# has a weak internal pull-up resistor to ensure a high level when the coprocessor error function is disabled.
IRQ12/M	IS	**INTERRUPT REQUEST/MOUSE INTERRUPT:** In addition to providing the standard interrupt function as described in the signal description for IRQ[15,14, 11:9, 7:3, 1], the IRQ12/M signal also provides a mouse interrupt function. The X-Bus Chip Select Register determines the functionality of IRQ12/M. An internal IRQ12 interrupt continues to be generated until a reset or an I/O read access to address 60h (falling edge of IOR#) is detected. After a reset, this pin provides the standard IRQ12 function.

2.2.7 PSC/IB LINK INTERFACE SIGNALS (IB)

Name	Type	Description
CMDV#	I/O	**COMMAND VALID:** The link master (PSC or IB) asserts CMDV# to indicate the beginning of a link transfer. The PSC negates this signal after a hard reset. CMDV# is used along with SIDLE# to set the PSC/IB system clock configuration during a PWROK hard reset. These inputs are strapped to the appropriate levels, sampled while PWROK is inactive, and latched when PWROK goes active (see Section 4.0, Functional Description).
SIDLE#	I/O	**SLAVE IDLE:** The link slave (PSC or IB) asserts SIDLE# to indicate that it is available for data transfers. The IB asserts this signal after a hard reset. SIDLE# is used along with CMDV# to set the PSC/IB system clock configuration during PWROK hard reset. These inputs are strapped to the appropriate levels, sampled while PWROK is inactive, and latched when PWROK goes active (see Section 4.0, Functional Description).
LREQ#	O	**LINK REQUEST:** The IB asserts LREQ# to request a link transfer. This signal is negated after a hard reset.
LGNT#	I	**LINK GRANT:** The PSC asserts LGNT# to grant the IB a link transfer. This signal is negated after a hard reset.
A[17:2]	I/O	**HOST ADDRESS/LINK:** For PSC/IB Link transfers, A[17:2] transfer data/commands between the IB and PSC. These signals are tri-stated after a hard reset.

2.2.8 SYSTEM POWER MANAGEMENT (SMM) SIGNALS (IB)

Name	Type	Description
SMI#	O	**SYSTEM MANAGEMENT INTERRUPT:** SMI# is an active low synchronous output that is asserted by the IB in response to one of many enabled hardware or software events. SMI# is ORed externally with SRESET and driven to the CPU. This signal is negated after a hard reset.
STPCLK#	O	**STOP CLOCK:** STPCLK# is an active low synchronous output that is asserted by the IB in response to one of many enableable hardware or software events. STPCLK# connects directly to the CPU. This signal is negated after a hard reset.
EXTSMI#	I	**EXTERNAL SYSTEM MANAGEMENT INTERRUPT:** EXTSMI# is a falling edge triggered input to the IB indicating that an external device is requesting the system to enter SMM mode. When enabled, a falling edge on EXTSMI# results in the assertion of the SMI# signal to the CPU. EXTSMI# is an asynchronous input to the IB. However, when the setup and hold times are met it is only required to be asserted for one HCLKIN. Once negated it must remain negated for at least four HCLKINs in order to allow the edge detect logic to reset.

2.2.9 SYSTEM CLOCK SIGNALS (IB)

Name	Type	Description
CLK2IN	I	**2X CLOCK IN:** CLK2IN is a 2X clock input. CLK2IN is divided as shown in Section 4.0, Functional Description, to generate HCLKIN, PCICLK, and SYSCLK.
CLK2OUT	O	**2X Clock Out:** CLK2OUT is a delayed version of CLK2IN. The PSC uses this clock.
HCLKOUT[2,1]	O	**HOST CLOCK OUT:** HCLKOUT[2,1] provide the reference clock for the IB, PSC, and CPU devices. The IB divides the CLK2IN input to generate HCLKOUT[2,1]. Either HCLKOUT2 or HCLKOUT1 is routed back to the IB's HCLKIN input providing the IB with its HCLKIN reference.
HCLKIN	I	**HOST CLOCK IN:** This 1X clock input provides the fundamental timing and internal operating frequency for the IB. This signal is connected as a feedback from the HCLKOUT outputs. The IB operates at 25 MHz or 33 MHz, depending on the frequency of the CLK2IN input.
PCICLK[2,1]	O	**PCI CLOCK OUT:** PCICLK[2,1] provide the reference clock for the PSC and PCI devices. The IB divides the CLK2IN input to generate PCICLK[2,1]. The PCI Bus operates at 25 MHz or 33 MHz, depending on the frequency of the CLK2IN input.
SYSCLK	O	**ISA SYSTEM CLOCK:** SYSCLK is the reference clock for the ISA Bus. It drives the ISA Bus directly. The SYSCLK frequencies supported are 8 MHz and 8.33 MHz.

2.2.10 SYSTEM RESET SIGNALS (IB)

Name	Type	Description
PWROK	I	**POWER OK:** When asserted, PWROK is an indication to the IB that power and CLK2IN have been stable for at least 1 ms. PWROK can be driven asynchronously. When PWROK is negated, the IB asserts CPURST, PCIRST# and RSTDRV. The IB also uses the rising edge of PWROK to sample the CMDV# and SIDLE# signals to determine the HCLKIN, PCICLK, and SYSCLK divisor values.
CPURST	O	**CPU RESET:** The IB asserts CPURST to reset the CPU, IB, and PSC. The IB asserts CPURST during power-up and when a hard reset sequence is initiated through the TRC Register. If a hard reset is initiated through the TRC register, the IB resets it's internal registers and signals to their default state, but maintains its clock divisor values.
SRESET	I	**Soft Reset:** SRESET is used internally by the IB.
PCIRST#	O	**PCI RESET:** The IB asserts PCIRST# to reset devices that reside on the PCI bus. The IB asserts PCIRST# during power-up and when a hard reset sequence is initiated through the TRC register. PCIRST# is driven asynchronously relative to PCICLK.
RSTDRV	O	**RESET DRIVE:** The IB asserts RSTDRV to reset devices that reside on the ISA Bus. The IB asserts this signal during a hard reset and during power-up.

2.2.11 TEST SIGNALS (IB)

Name	Type	Description
TESTIN#	I	**TEST:** The TESTIN# signal is used to tri-state all of the IB outputs. During normal operation, this input should be pulled up.

3.0 REGISTER DESCRIPTION

The 82420EX PCIset contains I/O control registers, PCI configuration registers, and ISA Compatible registers. These registers are discussed in this section.

The PCIset, upon receiving a hard reset, sets its internal registers to pre-determined **default** states. The default values are indicated in the individual register descriptions. Note that the default state of some ISA-Compatible register bits is indeterminate after a hard reset.

The following notation is used to describe register access attributes:

RO **Read Only.** If a register is read only, writes have no effect.

WO **Write Only.** If a register is write only, reads have no effect.

R/W **Read/Write.** A register with this attribute can be read and written. Note that individual bits in some read/write registers may be read only.

3.1 Register Access

Table 5, Table 6, and Table 7 show the I/O assignments for the I/O Control Registers, PCI Configuration Registers, and the ISA-Compatible Registers. Little-endian ordering is used for all multi-byte accesses (i.e., lower addresses contain the least significant parts of the fields).

NOTE:
Aliasing of the 90–9Fh address range to 80–8Fh is enabled/disabled in the ISA Controller Recovery Timer Register. When aliasing is enabled, the IB aliases I/O accesses in the 90–9Fh range to the 80–8Fh range. In this case, the IB only forwards write accesses to these locations to the ISA Bus. When aliasing is disabled, the IB allows both read and write accesses to the 90–9Fh range to be forwarded to the ISA Bus (i.e. they are no longer considered IB internal registers). Note that port 92h is always a distinct ISA register in the 90–9Fh range and is always forwarded to the ISA Bus. In addition, when aliasing is disabled, ISA master accesses to the 90h–9Fh range are ignored by the IB.

I/O Control Registers

The I/O control registers (Table 5) are located in the CPU I/O space and can only be accessed by the CPU. The TRC and CONFDATA Registers can be accessed as byte, word, or dword quantities. The CONFADD Register can only be accessed as a dword quantity.

The CONFADD and CONFDATA Registers are used to access PCI configuration space. This is accomplished in two steps. First, the PCI configuration address is written to the CONFADD Register using the PCI configuration space access mechanism 1 address field definitions. Second, configuration register data is read/written from/to the CONFDATA Register address location.

The address written to the CONFADD Register contains five programmable fields (Bus Number, Device Number, Function Number, Configuration Register Offset, and the configuration enable bit—bit 31). If the Device Number = 05, the Bus Number = 00, and bit 31 = 1, subsequent CONFDATA Register accesses, read/write the PCI configuration register pointed to by the Register Offset field. If the Register Offset field points to a reserved register location, reads return all 0's to the CPU and writes are ignored by the PSC. If bit 31 = 1, but the Device Number\05, a PCI configuration cycle is run on the PCI Bus. If bit 31 = 0 (regardless of the Bus Number or Device Number values), the access to the CONFDATA Register location is forwarded to the PCI Bus and, if unclaimed on PCI, forwarded to ISA as a normal access to I/O address 0CFCh.

Table 5. I/O Control Registers

I/O Address	Mnemonic	Register	Register Access	Bus Master
0CF8h	CONFADD	Configuration Address	R/W	CPU Only
0CFCh	CONFDATA	Configuration Data	R/W	CPU Only
0CF9h	TRC	Turbo/Reset Control	R/W	CPU Only

Mechanism 1 PCI Configuration Address Fields

31	30 24	23 16	15 11	10 8	7 2	1 0
	Reserved	Bus Number	Device Number	Function Number	Register Offset	0 0

NOTE:
Device number=05 is equivalent to IDSEL 16. Thus, other PCI devices cannot use IDSEL 16.

PCI Configuration Registers

The PCI configuration registers are located in the 82420EX PCIset's PCI configuration space and can only be accessed by the CPU. These registers (Table 6) can be accessed as byte, word, or dword quantities. The addresses for the configuration registers in the table are PCI configuration space offset values. The CPU accesses PCI configuration space (all PCI devices including the PSC) using mechanism 1 configuration access. For a detailed description of the PCI mechanism 1 configuration access, refer to the PCI Local Bus Specification document.

Some of the PCI configuration registers contain reserved bits. When reserved bits are read, a value of 0 is returned. In addition, the PCI configuration space includes reserved I/O locations. When reserved I/O locations are read, a value of 00h is returned. Writes to reserved bits or reserved I/O locations have no affect.

ISA-Compatible Registers

The ISA Compatible registers (Table 7) include DMA registers, timer registers, interrupt control registers, non-maskable interrupt, X-Bus support, and advanced power management control. These registers can be accessed by the CPU, a PCI master, or an ISA master as shown in Table 7. CPU or PCI masters can access the ISA-Compatible registers as 8-bit, 16-bit, 24-bit, or 32-bit quantities. However, only the first active BE[3:0]# is processed by the PSC. The remaining active byte enables in the same cycle are ignored. ISA Bus masters access the registers as 8-bit quantities. Unless otherwise stated in the individual register description, reserved bits must be written with a 0 and these bits return a 0 when read.

Table 6. PCI Configuration Register

Configuration Offset	Mnemonic	Register	Register Access	Bus Master
00–01h	VID	Vendor Identification	RO	CPU Only
02–03h	DID	Device Identification	RO	CPU Only
04–05h	PCICOM	PCI Command	R/W	CPU Only
06–07h	DS	Device Status	R/WC	CPU Only
08h	RID	Revision Identification	RO	CPU Only
09–3Fh	—	Reserved	—	—
40h	PCICON	PCI Control	R/W	CPU Only
41–43h	—	Reserved	—	—
44h	HDEVCON	Host Device Control	R/W	CPU Only
45–47h	—	Reserved	—	—
48–49h	LBIDE	PCI Local Bus IDE Control	R/W	CPU Only
4A–4Bh	—	Reserved	—	—
4Ch	IORT	ISA I/O Recovery Timer	R/W	CPU Only
4Dh	PREV	Part Revision Identification	R/W	CPU Only
4Eh	XBCSA	X-Bus Chip Select Enable A	R/W	CPU Only
4Fh	—	Reserved	—	—
50h	HOSTSEL	Host Bus Select	R/W	CPU Only
51h	DFC	Deturbo Frequency Control Register	R/W	CPU Only
52–53h	SCC	Secondary Cache Control	R/W	CPU Only
54–55h	—	Reserved	—	—
56–57h	DRAMC	DRAM Control	R/W	CPU Only
58h	—	Reserved	—	—

Table 6. PCI Configuration Register (Continued)

Configuration Offset	Mnemonic	Register	Register Access	Bus Master
59–5Fh	PAM	Programmable Attribute Map Registers	R/W	CPU Only
60–64h	DRB	DRAM Row Boundary Registers	R/W	CPU Only
65h	—	Reserved	—	—
66h	PIRQ0RC	PIRQ0 Route Control	R/W	CPU Only
67h	PIRQ1RC	PIRQ1 Route Control	R/W	CPU Only
68h	DMH	DRAM Memory Hole	R/W	CPU Only
69h	TOM	Top of Memory	R/W	CPU Only
6A–6Fh	—	Reserved	—	—
70h	SMRAMCON	SMRAM Control	R/W	CPU Only
71–9Fh	—	Reserved	—	—
A0h	SMICNTL	SMI Control	R/W	CPU Only
A1h	—	Reserved	R/W	CPU Only
A2–A3h	SMIEN	SMI Enable	R/W	CPU Only
A4–A7h	SEE	System Event Enable	R/W	CPU Only
A8h	FTMR	Fast Off Timer	R/W	CPU Only
A9	—	Reserved	—	—
AA–ABh	SMIREQ	SMI Request	R/W	CPU Only
ACh	CTLTMRL	Clock Throttle STPCLK# Low Timer	R/W	CPU Only
ADh	—	Reserved	—	—
AEh	CTLTMRH	Clock Throttle STPCLK# High Timer	R/W	CPU Only
AF–FFh	—	Reserved	—	—

Table 7. ISA-Compatible Registers

Address	FEDC	BA98	7654	3210	Register Name	Access Type	Bus Access
0000h	0000	0000	000x	0000	DMA1 CH0 Base and Current Address	r/w	CPU/PCI
0001h	0000	0000	000x	0001	DMA1 CH0 Base and Current Count	r/w	CPU/PCI
0002h	0000	0000	000x	0010	DMA1 CH1 Base and Current Address	r/w	CPU/PCI
0003h	0000	0000	000x	0011	DMA1 CH1 Base and Current Count	r/w	CPU/PCI
0004h	0000	0000	000x	0100	DMA1 CH2 Base and Current Address	r/w	CPU/PCI
0005h	0000	0000	000x	0101	DMA1 CH2 Base and Current Count	r/w	CPU/PCI
0006h	0000	0000	000x	0110	DMA1 CH3 Base and Current Address	r/w	CPU/PCI
0007h	0000	0000	000x	0111	DMA1 CH3 Base and Current Count	r/w	CPU/PCI
0008h	0000	0000	000x	1000	DMA1 Status (r), Command (w)	r/w	CPU/PCI
0009h	0000	0000	000x	1001	DMA1 Request	wo	CPU/PCI
000Ah	0000	0000	000x	1010	DMA1 Write Single Mask Bit	wo	CPU/PCI
000Bh	0000	0000	000x	1011	DMA1 Channel Mode	wo	CPU/PCI
000Ch	0000	0000	000x	1100	DMA1 Clear Byte Pointer	wo	CPU/PCI
000Dh	0000	0000	000x	1101	DMA1 Master Clear	wo	CPU/PCI
000Eh	0000	0000	000x	1110	DMA1 Clear Mask	wo	CPU/PCI
000Fh	0000	0000	000x	1111	DMA1 Write All Mask Bits	r/w	CPU/PCI
0020h	0000	0000	001x	xx00	INT 1 Control	r/w	CPU/PCI/ISA
0021h	0000	0000	001x	xx01	INT 1 Mask	r/w	CPU/PCI/ISA
0040h	0000	0000	010x	0000	Timer Counter 1 - Counter 0 Count	r/w	CPU/PCI/ISA
0041h	0000	0000	010x	0001	Timer Counter 1 - Counter 1 Count	r/w	CPU/PCI/ISA
0042h	0000	0000	010x	0010	Timer Counter 1 - Counter 2 Count	r/w	CPU/PCI/ISA
0043h	0000	0000	010x	0011	Timer Counter 1 Command Mode	wo	CPU/PCI/ISA
0060h[1]	0000	0000	0110	00x0	Reset X-Bus IRQ12/M and IRQ1	r	CPU/PCI/ISA
0061h	0000	0000	0110	0xx1	NMI Status and Control	r/w	CPU/PCI/ISA

Table 7. ISA-Compatible Registers (Continued)

Address	FEDC	BA98	7654	3210	Register Name	Access Type	Bus Access
0070h[1]	0000	0000	0111	0xx0	CMOS RAM Address and NMI Mask	wo	PCI/ISA
0080h[2]	0000	0000	100x	0000	DMA Page Register (Reserved)	r/w	CPU/PCI/ISA
0081h	0000	0000	100x	0001	DMA Channel 2 Page	r/w	CPU/PCI/ISA
0082h	0000	0000	1000	0010	DMA Channel 3 Page	r/w	CPU/PCI/ISA
0083h	0000	0000	100x	0011	DMA Channel 1 Page	r/w	CPU/PCI/ISA
0084h[2]	0000	0000	100x	0100	DMA Page Register (Reserved)	r/w	CPU/PCI/ISA
0085h[2]	0000	0000	100x	0101	DMA Page Register (Reserved)	r/w	CPU/PCI/ISA
0086h[2]	0000	0000	100x	0110	DMA Page Register (Reserved)	r/w	CPU/PCI/ISA
0087h	0000	0000	100x	0111	DMA Channel 0 Page	r/w	CPU/PCI/ISA
0088h[2]	0000	0000	100x	0100	DMA Page Register (Reserved)	r/w	CPU/PCI/ISA
0089h	0000	0000	100x	1001	DMA Channel 6 Page	r/w	CPU/PCI/ISA
008Ah	0000	0000	100x	1010	DMA Channel 7 Page	r/w	CPU/PCI/ISA
008Bh	0000	0000	100x	1011	DMA Channel 5 Page	r/w	CPU/PCI/ISA
008Ch[2]	0000	0000	100x	1100	DMA Page Register (Reserved)	r/w	CPU/PCI/ISA
008Dh[2]	0000	0000	100x	1101	DMA Page Register (Reserved)	r/w	CPU/PCI/ISA
008Eh[2]	0000	0000	100x	1110	DMA Page Register (Reserved)	r/w	CPU/PCI/ISA
008Fh	0000	0000	100x	1111	DMA Low Page Register Refresh	r/w	CPU/PCI/ISA
00A0h	0000	0000	101x	xx00	INT 2 Control	r/w	CPU/PCI/ISA
00A1h	0000	0000	101x	xx01	INT 2 Mask	r/w	CPU/PCI/ISA
00B2h	0000	0000	1011	0010	Advanced Power Management Control	r/w	CPU Only
00B3h	0000	0000	1011	0011	Advanced Power Management Status	r/w	CPU/PCI
00C0h	0000	0000	1100	000x	DMA2 CH0 Base and Current Address	r/w	CPU/PCI
00C2h	0000	0000	1100	001x	DMA2 CH0 Base and Current Count	r/w	CPU/PCI
00C4h	0000	0000	1100	010x	DMA2 CH1 Base and Current Address	r/w	CPU/PCI

Table 7. ISA-Compatible Registers (Continued)

Address	FEDC	BA98	7654	3210	Register Name	Access Type	Bus Access
00C6h	0000	0000	1100	011x	DMA2 CH1 Base and Current Count	r/w	CPU/PCI
00C8h	0000	0000	1100	100x	DMA2 CH2 Base and Current Address	r/w	CPU/PCI
00CAh	0000	0000	1100	101x	DMA2 CH2 Base and Current Count	r/w	CPU/PCI
00CCh	0000	0000	1100	110x	DMA2 CH3 Base and Current Address	r/w	CPU/PCI
00CEh	0000	0000	1100	111x	DMA2 CH3 Base and Current Count	r/w	CPU/PCI
00D0h	0000	0000	1101	000x	DMA2 Status(r) Command(w)	r/w	CPU/PCI
00D2h	0000	0000	1101	001x	DMA2 Request	wo	CPU/PCI
00D4h	0000	0000	1101	010x	DMA2 Write Single Mask Bit	wo	CPU/PCI
00D6h	0000	0000	1101	011x	DMA2 Write Mode Register	wo	CPU/PCI
00D8h	0000	0000	1101	100x	DMA2 Clear Byte Pointer	wo	CPU/PCI
00DAh	0000	0000	1101	101x	DMA2 Master Clear	wo	CPU/PCI
00DCh	0000	0000	1101	110x	DMA2 Clear Mask	wo	CPU/PCI
00DEh	0000	0000	1101	111x	DMA2 Write All Mask Bits	r/w	CPU/PCI
00F0h[1]	0000	0000	1111	0000	Coprocessor Error	wo	CPU/PCI/ISA
0481h	0000	0100	1000	0001	DMA CH2 High Page Register	r/w	CPU/PCI/ISA
0482h	0000	0100	1000	0010	DMA CH3 High Page Register	r/w	CPU/PCI/ISA
0483h	0000	0100	1000	0011	DMA CH1 High Page Register	r/w	CPU/PCI/ISA
0487h	0000	0100	1000	0111	DMA CH0 High Page Register	r/w	CPU/PCI/ISA
0489h	0000	0100	1000	1001	DMA CH6 High Page Register	r/w	CPU/PCI/ISA
048Ah	0000	0100	1000	1010	DMA CH7 High Page Register	r/w	CPU/PCI/ISA
048Bh	0000	0100	1000	1011	DMA CH5 High Page Register	r/w	CPU/PCI/ISA
04D0h	0000	0100	1101	0000	INT-1 Edge/Level Control	r/w	CPU/PCI/ISA
04D1h	0000	0100	1101	0001	INT-2 Edge/Level Control	r/w	CPU/PCI/ISA

NOTES:
1. Read and write accesses to these locations are always forwarded to the ISA Bus.
2. Write accesses to these locations are forwarded to the ISA Bus. Read Accesses are not forwarded to the ISA Bus. If programmed in the ISA I/O Recovery Timer Register, the IB will not alias the 90–9Fh address range with the 80–8Fh address range. In this case, accesses to the 90–9Fh address range are forwarded to the ISA Bus for both reads and writes (i.e. they are no longer considered IB registers).

3.2 I/O Control Registers

There are three I/O control registers (CONFADD, CONFDATA, and TRC) and these registers are all located in the CPU I/O space.

3.2.1 CONFADD—CONFIGURATION ADDRESS REGISTER

IO Address: 0CF8h
Default Value: 00000000h
Attribute: Read/Write
Size: 32 Bits

The CONFADD Register contains the address information for the next PCI configuration space access. Once the address is programmed into this register, the CPU can access the selected device register by a read/write to the CONFDATA Register. Only dword accesses are permitted to this register.

Bit	Description
31	**PCI Configuration Space Enable (CONFE):** This bit enables/disables access to the PCI configuration space. When CONFE = 1, PCI configuration space access is enabled. When CONFE = 0 (default), PCI configuration space access is disabled. When disabled, accesses to the CONFDATA Register, if not claimed on the PCI Bus, are forwarded to the ISA Bus.
30:24	**Reserved**
23:16	**Bus Number (BUSNUM):** This field selects the PCI Bus to be accessed. The PCI Bus behind the PSC is bus number 0 (00h). Thus, this field must be 00h for access to the PCIset's configuration registers.
15:11	**Device Number (DEVNUM):** This field selects the PCI Bus device to be accessed. The PSC uses this field to drive the IDSEL lines that select a specific PCI device during initialization. The IDSEL lines are only driven when BUSNUM = 00h. Otherwise, the PSC sends the configuration to a PCI-to-PCI bridge device. This field must be 05h for access to the PCIset's configuration registers (which is equivalent to IDSEL 16). Note that other PCI devices cannot use IDSEL 16.
10:8	**Function Number (FUNCNUM):** A device connected to the PCI Bus can have up to eight functions. This field selects a particular function within a device and must be 000 for access to the PCIset's configuration registers.
7:2	**Register Number (REGNUM):** This field is the configuration register offset address and indexes a dword in configuration space. REGNUM is used during initialization to select a specific device configuration registers.
1:0	**Reserved:** Fixed at 00.

3.2.2 CONFDATA—CONFIGURATION DATA REGISTER

IO Address: 0CFCh
Default Value: 00000000h
Attribute: Read/Write
Size: 32 Bits

The CONFDATA Register contains the data that is sent or received during a PCI configuration space access. Note that a read or write to this register accesses the PCI configuration space location specified by the contents of the CONFADD Register. CONFDATA supports CPU byte, word, and dword accesses.

3.2.3 TRC—TURBO/RESET CONTROL REGISTER

IO Address: 0CF9h
Default Value: 00h
Attribute: Read/Write
Size: 8 Bits

The TRC Register provides a means of generating soft or hard resets. During a hard reset, CPURST, PCIRST#, and RSTDRV are asserted for approximately 1 ms. A hard reset is initiated when this register is programmed for a hard reset or PWROK is asserted. During a soft reset, SRESET is asserted for a minimum of 16 Host Bus clocks. This register also selects the CPU De-Turbo mode. The TRC Register can only be accessed by the CPU with 8 bit IN or OUT instructions. Note that it is illegal for a PCI master or an ISA master to access the TRC Register.

Bit	Description
7:3	**Reserved:** Must be 0 when programming this register.
2	**Reset CPU (RCPU):** RCPU is used to initiate a hard reset or soft reset to the CPU, depending on the state of bit 1 of this register. Bit 1 must be set up prior to writing a 1 to bit 2. Thus, two write operations are required to initiate a reset. The first write programs bit 1 to the appropriate state while setting this bit to 0. The second write operation keeps bit 1 at its programmed state while setting this bit to a 1. When RCPU transitions from a 0 to a 1, a hard reset is initiated if bit 1 = 1 and a soft reset is initiated if bit 1 = 0.
1	**Reset CPU Mode (RCPUM):** This bit is used in conjunction with bit 2 of this register to initiate either a hard or soft reset. When RCPUM = 1, the PSC initiates a hard reset to the CPU when bit 2 transitions from 0 to 1. When RCPUM = 0, the PSC initiates a soft reset when bit 2 transitions from a 0 to a 1.
0	**Deturbo Mode (DM):** This bit enables/disables deturbo mode. When DM = 1, the 82420EX PCIset is in deturbo mode. In this mode, the PSC periodically asserts HOLD. The frequency of the HOLD assertion is fixed at once in 1024 Host Clocks. The duty cycle of the HOLD active period is controlled by the Deturbo Frequency Control (DFC) Register. When DM = 0, the Deturbo mode is disabled. Note that the deturbo counter does not start until HLDA is returned by the CPU. Deturbo mode can be used to maintain backward compatibility with older software packages that rely on the operating speed of the older processors. For accurate speed emulation, L1 caching should be disabled. If L1 is disabled during runtime, the following steps should be performed to make sure that all dirty data has been flushed from the cache to main memory before entering deturbo mode. Disable the L1 cache via the L1EN bit in the HOSTSEL Register. This prevents the KEN# signal from being asserted, which effectively disables the L1 cache. Thus, no new L1 cache line fills will occur. At this point, software executes the **WBINVD** of **INVD** instruction to flush the L1 cache, and then sets DM to 1. When exiting the deturbo mode, the system software must first set DM to 0, then enable L1 caching by writing to the HOSTSEL Register.

3.3 PCI Configuration Registers

This section describes the PCI configuration registers of the 82420 PCIset. The registers are listed in the order that they appear in Table 6.

3.3.1 VID—VENDOR IDENTIFICATION REGISTER

Address Offset: 00–01h
Default Value: 8086h
Attribute: Read Only
Size: 16 Bits

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

3.3.2 DID—DEVICE IDENTIFICATION REGISTER

Address Offset: 02–03h
Default Value: 0486h
Attribute: Read Only
Size: 16 Bits

This 16-bit register, combined with the Vendor Identification register, uniquely identifies any PCI device. The 16-bit value in this register is the device number assigned to the 82425EX PSC. Writes to this register have no effect.

3.3.3 PCICOM—PCI COMMAND REGISTER

Address Offset: 04–05h
Default Value: 0007h
Attribute: Read/Write
Size: 16 Bits

This 16-bit register enables/disables the SERR# signal.

Bit	Description
15:9	**Reserved**
8	**SERR# Enable (SERRE):** SERRE enables/disables the SERR# signal. When SERRE = 1, SERR# is asserted if the PCIset detects a parity error during a main memory read cycle or a target abort is received on a PSC-initiated PCI cycle. When SERRE = 0, SERR# is never asserted.
7:0	**Not Used:** Defaults to 07h for compatibility reasons.

3.3.4 DS—DEVICE STATUS REGISTER

Address Offset: 06–07h
Default Value: 0200h
Attribute: Read Only and Read/Write Clear
Size: 16 Bits

DS is a 16-bit status register that reports the occurrence of a PCI master abort, PCI target abort, and main memory or cache parity errors. PCISTS also indicates the DEVSEL# timing that has been set by the PSC hardware.

Bit	Attribute	Description
15	R/WC	**Main Memory Parity Error (MMPERR):** When the PSC detects a parity error, this bit is set to 1. Software sets this bit to 0 by writing a 1 to it.
14	R/WC	**SERR# Status (SERRS):** When the PSC asserts the SERR# signal, this bit is set to a 1. Note that the SERR# signal is enabled/disabled in the PCICOM Register. When SERR# is enabled (via the PCICOM Register) and the PSC detects a parity error on a main memory read cycle or receives a target abort during a PSC-initiated PCI cycle, the PSC sets this bit to a 1.
13	R/WC	**Master Abort Status (MAS):** When a PSC-initiated PCI configuration cycle is not claimed, the PSC master aborts the cycle and sets this bit to a 1. For a CPU read, the PSC returns all 1s. When a memory cycle above 16 MBytes and not in an enabled BIOS region is not claimed, the PSC master-aborts the cycle and sets this bit to a 1. Software sets this bit to 0 by writing a 1 to it. **NOTE:** When a PSC-initiated PCI memory access under 16 MBytes or in an enabled BIOS range above 16 MBytes is not claimed, the PSC master aborts the cycle and forwards the cycle to ISA. When a PSC-initiated PCI I/O access is not claimed, the PSC master aborts the cycle and forwards the cycle to ISA. For these master aborts, the MAS bit is not set to 1.
12	R/WC	**Received Target Abort Status (RTAS):** When a PSC-initiated PCI transaction is terminated with a target abort, RTAS is set to 1. In addition, when the SERRE bit in the PCICOM Register is set to a 1 (enabling SERR#), the PSC asserts the SERR# signal. Software sets this bit to 0 by writing a 1 to it. Note that if the target aborted cycle is an I/O or memory read, the PSC completes the CPU cycle by returning RDY#.
11		**Reserved**
10:9	RO	**DEVSEL# Timing (DEVT):** This 2-bit field indicates the timing of the DEVSEL# signal when the PSC responds as a target. The PCI specification defines three allowable timings for assertion of DEVSEL#: 00 = fast, 01 = medium, and 10 = slow (DEVT = 11 is reserved). DEVT indicates the slowest time that a device asserts DEVSEL# for any bus command, except configuration read and write cycles. The 82420EX PCIset implements medium speed DEVSEL# timing and, therefore, bits[10:9] = 01 when read.
8:0		**Reserved**

3.3.5 RID—REVISION IDENTIFICATION REGISTER

Address Offset: 08h
Default Value: 00h
Attribute: Read Only
Size: 8 Bits

This register contains the revision number of the PSC. The Fabrication House ID Number and Revision Number correspond to bits 7-5 and the lower nibble respectively of the Revision Identification Register as follows:

bits 7-5 (upper 3 bits) Fabrication House ID Number
bits 3-0 (lower nibble) Revision Number

3.3.6 PCICON—PCI CONTROL REGISTER

Address Offset: 40h
Default Value: 00h
Attribute: Read /Write
Size: 8 Bits

The PCICON register enables/disables target abort and main memory DRAM parity error reporting. This register also selects the subtractive decode sample point, enables/disables PCI write buffers, and controls PCI bursting of consecutive CPU-to-PCI write cycles and byte merging.

Bit	Description
7	**Reserved**
6	**Target Abort Error Enable (TAEE):** When ATEE = 1 and a PSC-initiated cycle is target aborted, the PSC asserts SERR# for a PCI Clock. When TAEE = 0 (default) and a PSC-initiated cycle is target aborted, the PSC does not assert SERR#.
5	**DRAM Parity Error Enable (DPEE):** When DPEE = 1 and a main memory parity error is detected, the PSC asserts SERR# for a PCI Clock. When DPEE = 0 (default) and a main memory parity error is detected, the PSC will not assert SERR#.
4:3	**Subtractive Decode Sample Point (SDSP):** The SDSP field determines the DEVSEL# sample point, after which an inactive DEVSEL# results in the PSC forwarding the unclaimed PCI cycle to the ISA Bus (subtractive decoding). This setting should match the slowest device in the system. The values for this field and associated sampling point meaning are shown below. **Bits[4:3] — Sampling Point** 00 — Slow (default) 01 — Typical 10 — Fast 11 — Reserved
2	**PCI Posted Write Buffer Enable (PPBE):** When PPBE = 1, the PCI posted write buffers are enabled. When PPBE = 0 (default), the PCI posted write buffers are disabled.

Bit	Description
1	**CPU-to-PCI Bursting Enable (CPBE):** This bit enables/disables PCI burst cycles for CPU-to-PCI write cycles. When CPBE = 1, back-to-back sequential CPU memory writes are sent out on the PCI Bus as a burst cycle. When CPBE = 0 (default), CPU write cycles are always sent out on the PCI Bus as separate PCI memory write cycles.
0	**CPU-to-PCI Byte Merging (CPME):** Byte merging permits the PSC to merge the data of consecutive CPU-to-PCI byte/word writes within the same dword address, into the same posted write buffer location. The merged collection of bytes is then sent over the PCI Bus as a single dword. Byte merging is performed in the compatible VGA range only (0A0000–0BFFFFh). When CPME = 1, back-to-back CPU memory byte/word write cycles within the same dword address (in the 0A0000–0BFFFFh range) are merged into a single posted write buffer location. When CPME = 0 (default), or when the address location is outside of the VGA range, each memory write is stored in a separate posted write buffer location and sent separately over the PCI Bus. Some PCI graphics cards memory map their I/O location in the A0000h to B0000h memory region. If consecutive, multiple 8 or 16 bit write cycles are made to the add-in card at a memory mapped I/O location between A0000h–BFFFFh, the PSC will merge the data if the PSC is programmed for byte merging. The first write cycle will be written to the add-in card. However, subsequent write cycles will be overwritten in the PSC and never reach the add-in card. Because the consecutive, multiple write cycles are to the same address, the PSC will "merge" (overwrite the previous data) as long as the PCI bus is unavailable, causing the add-in card to not receive all the intended write cycles. Byte merging should be disabled when used with graphics cards that memory map I/O locations in the compatibility Video buffer area (A0000h–BFFFFh). Byte merging enhances graphics performances when used in operating systems that write to the video memory area in 8- or 16-bit writes (e.g. DOS). For operating systems that write to the video memory area in 32-bit writes, byte merging is not necessary.

3.3.7 HOSTDEV—HOST DEVICE CONTROL REGISTER

Address Offset: 44h
Default Value: 00h
Attribute: Read /Write
Size: 8 Bits

The HOSTDEV Register indicates to the PSC if there is a slave device, other than the PSC, that resides on the Host Bus. If there is another slave device present, the PSC sampling points for HDEV# and HRDY# are set in this register.

Bit	Description
7:3	**Reserved**
2	**Host Device Present (HDEVP):** When HDEVP = 1, there is a Host Bus slave device present. This device can claim any I/O or memory range that is not positively decoded by the PSC by asserting HDEV#. When HDEVP = 0 (default), there is no host bus slave device in the system.
1	**HDEV# Signal Sampling Point (HDEVSP):** HDEVSP specifies the maximum delay for HDEV# response and this bit only has meaning when HDEVP = 1. When HDEVSP = 1 (and HDEVP = 1), the PSC assumes that the Host Bus slave device asserts HDEV# with Host Bus fast timing (i.e., HDEV# can be asserted as late as one host clock after ADS# is asserted). In this case, the PSC samples HDEV# in the host cycle after ADS# and, if not asserted, forwards the cycle to the PCI Bus. When HDEVSP = 0 (and HDEVSP = 1), the PSC assumes that the Host Bus slave device asserts HDEV# with Host Bus slow timing (i.e., HDEV# can be asserted as late as two host clocks after ADS# is asserted). In this case, the PSC samples HDEV# in the host cycle after ADS# and one host cycle later and if not asserted, forwards the cycle to the PCI Bus.
0	**HRDY# Maximum Signal Sampling Point (HRDYSP):** HDEVSP specifies the delay from HRDY# to RDY# and this bit only has meaning when HDEVP = 1. When HRDYSP = 1 (and HDEVP = 1), the PSC assumes that the Host Bus slave device asserts HRDY# with Host Bus fast timing (i.e., Host RDY# is asserted in the same host clock as HRDY# is asserted). When HRDYSP = 0 (and HDEVP = 1), the PSC assumes that the Host Bus slave device asserts HRDY# with host bus slow timing (i.e., Host RDY# is asserted one host clock after HRDY# is asserted). Note that, when HDEVP = 0, HRDYSP has no meaning.

3.3.8 LBIDE—PCI LOCAL BUS IDE CONTROL REGISTER

Address Offset: 48–49h
Default Value: 0000h
Attribute: Read /Write
Size: 16 Bits

The LBIDE Register controls the PSC's IDE interface. The register determines when the PCI Local Bus IDE path will be used and the timing characteristics of the PCI Local Bus IDE cycle.

Bit	Description
15:13	**Reserved**
12:10	**Recover Time (RCT[2:0]):** This field controls the minimum time from the time IORDY is sampled asserted on the first cycle to the IOx# assertion of the next cycle. This recovery time mechanism applies to all cycles using the fast timing bank, even if the next cycle is a compatible cycle. For example, if a first cycle is a data port access using the fast timing bank and the next cycle is to a control or status port, LBIDE# is negated and the full setup protocol occurs prior to the second cycle. Normally, this setup protocol is longer than the programmed recovery time. However, if the setup protocol is shorter, the proper recovery time must still be met. The value of this field determines the minimum number of PCI clocks between the last IORDY sample point and the IOx# strobe of the next cycle. **RCT[2:0]** / **Bits[12:10]** / **Recovery Time** 000 — 8 001 — 7 010 — 6 011 — 5 100 — 4 101 — 3 110 — Note 111 — Note **NOTE:** The recovery time is 3 PCI clocks.
9:8	**IORDY Sample Point (ISP[1:0]):** This field determines the number of clocks between IOx# assertion and the first IORDY sample point (see the following table). IORDY is sampled for the first time on the programmed number of clocks (number of low-to-high clock transitions) following the assertion of IOx#. If IORDY is negated when sampled, wait-states are inserted until IORDY is sampled asserted. **ISP[1:0] Bits[9:8]** / **IORDY Sampling Point** 00 — 6 Clocks (default) 01 — 5 Clocks 10 — 4 Clocks 11 — 3 Clocks
7:6	**Reserved**
5:4	**IORDY Sample Point Enable Drive Select (ISPEDS[1:0]):** ISPEDS[1:0] enable/disable the sampling of IORDY for drive 0 and 1. When this feature is enabled for a drive (via this field), and the drive is selected (via a copy of bit 4 of 1x6h), all accesses to the enabled I/O address range sample IORDY. The IORDY sample point is specified by the ISP[1:0] field. When the drive is disabled (0), IORDY sampling is disabled. The internal IORDY signal is forced asserted guaran-teeing that IORDY is sampled asserted at the first sample point as specified by the ISP[1:0] field. **ISPEDS[1:0] Bits[5:4]** / **IORDY Sampling Point Enable Drive Select** 00 — Neither Drive Enabled (default) 01 — Drive 0 Enabled 10 — Drive 1 Enabled 11 — Both Drives Enabled

Bit	Description
3:2	**Fast Timing Bank Drive Select 1 (FTBDS1):** FTBDS[1:0] enable/disable the fast timing PCI local bus IDE path for drive 0 and 1. When this feature is enabled for a drive (via this field), and the drive is selected (via a copy of bit 4 of 1x6h), all accesses to the enabled I/O address range will use the fast timing bank PCI local bus IDE path. Note that accesses to all non-data ports of the enabled I/O address range use the 8-bit compatible timing PCI local bus path. When the drive is disabled (0), accesses to the data port of the selected I/O address range use the 16-bit compatible timing PCI local bus path. **Bits[3:2] — Fast Timing Bank Select** 00 — Neither Drive Enabled (default) 01 — Drive 0 Enabled 10 — Drive 1 Enabled 11 — Both Drives Enabled
1:0	**Primary/Secondary PCI IDE Enable (IDEE):** This field enables/disables the PCI IDE, and, if enabled, selects the primary/secondary IDE address that is used as shown below. Accesses to the unselected address range (primary/secondary) are forwarded to the ISA Bus. **Bits[1:0] — Primary/Secondary IDE Address Select** 00 — IDE Disabled (default) 01 — Primary: 1F0–1F7h, 3F6h,3F7h 10 — Secondary: 170–177H, 376h, 377h 11 — Reserved

1

LBIDE Programming Information

The BIOS code will assess the CPU frequency and drive capabilities, and then program the timing fields appropriately. Table 8 shows the typical settings of the various cycle timing bits for the supported CPU frequencies and IDE modes. The table assumes that the drives are utilizing IORDY. If IORDY is not utilized, additional wait-states may be deleted via the ISP bits.

Table 8. Typical Register Settings for Different CPU Frequencies

PCI Frequency	IDE Mode	RCT(2:0)		ISP(1:0)		Cycle Length (ns) 1x Clock Mode(1)
		Bits[12:10]	Clocks	Bits[9:8]	Clocks	
20	1	100	4	10	4	400(1)
25	1	101	6	10	4	400
33	1	001	7	00	6	390
20	2	110	2	11	3	300(2)
25	2	110	2	10	4	280(2)
33	2	101	3	01	5	240
20	3	110	2	11	3	300(2)
25	3	110	2	11	3	240(2)
33	3	101	3	10	4	210

NOTES:
1. The clock modes are determined by strapping options at powerup and the mode is reflected in the HOSTSEL Register.
2. Cycle times are governed by the inherent RDY#, ADS#, and address decoding delay between back-to-back cycles rather than the programmed value in the RCT field.

3.3.9 IORT—ISA I/O RECOVERY TIMER REGISTER

Address Offset: 4Ch
Default Value: 4Dh
Attribute: Read/Write
Size: 8 bits

The IORT Register provides ISA I/O recovery time control and enables/disables the aliasing of addresses 80–8Fh and 90–9Fh. The I/O recovery mechanism in the IB adds recovery delay between CPU or PCI master originated 8-bit and 16-bit I/O cycles to the ISA Bus. The IB automatically forces a minimum delay of 3.5 SYSCLKs between back-to-back 8 and 16 bit I/O cycles to the ISA Bus. This delay is measured from the rising edge of the I/O command (IOR# or IOW#) to the falling edge of the next I/O command. If a delay of greater than 3.5 SYSCLKs is required, the ISA I/O Recovery Time Register can be programmed to increase the delay in increments of SYSCLKs. No additional delay is inserted for back-to-back I/O "sub-cycles" generated as a result of byte assembly or disassembly. This register defaults to 8- and 16-bit recovery enabled with one SYSCLK clock added to the standard I/O recovery.

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>7</td><td>DMA Reserved Page Register Aliasing Disable (DMAAD): When DMAAD = 0 (default), the IB aliases I/O accesses in the 90–9Fh range to the 80–8Fh range. In this case, the IB only forwards CPU/PCI write accesses to the 90–9Fh range to the ISA Bus. When DMAAD = 1, the IB forwards both CPU/PCI read and write accesses to these address locations to the ISA Bus (i.e. the I/O address locations are no longer considered IB internal registers). Note, that port 92h is always a distinct ISA register in the 90–9Fh range and is always forwarded to the ISA Bus. When DMAAD = 1, ISA master accesses to the 90–9Fh range are ignored by the IB.

When this bit is set to 1, the IB does not re-load the power management Fast Off-Timer with its original value when accesses to the 90–9Fh address range are decoded.</td></tr>
<tr><td>6</td><td>8-Bit I/O Recovery Select (IOR8E): When IOR8E = 1, bits[5:3] select the I/O recovery time. When IOR8E = 0, programmable recovery times are disabled and the standard recovery time of 3.5 SYSCLKS is inserted.</td></tr>
<tr><td>5:3</td><td>8-Bit I/O Recovery Times (IOR8): This 3-bit field defines the recovery time for 8-bit I/O as shown Below. Programmable delays between back-to-back 8-bit PCI or CPU cycles to ISA I/O slaves are shown in terms of additional ISA clock recovery cycles (SYSCLK). The selected delay programmed into this field is enabled/disabled via bit 6 of this register.

<table>
<tr><th>Bits[5:3]</th><th>8-Bit I/O Recover Time</th><th>Bits[5:3]</th><th>8-Bit I/O Recover Time</th></tr>
<tr><td>000</td><td>8 SYSCLKs</td><td>100</td><td>4 SYSCLKs</td></tr>
<tr><td>001</td><td>1 SYSCLKs (default)</td><td>101</td><td>5 SYSCLKs</td></tr>
<tr><td>010</td><td>2 SYSCLKs</td><td>110</td><td>6 SYSCLKs</td></tr>
<tr><td>011</td><td>3 SYSCLKs</td><td>111</td><td>7 SYSCLKs</td></tr>
</table></td></tr>
<tr><td>2</td><td>16-Bit I/O Recovery Enable (IOR16E): When IOR16E = 1, bits[1:0] select the I/O recovery time. When IOR16E = 0, programmable recovery times are disabled and the standard recovery time of 3.5 SYSCLKS is inserted.</td></tr>
</table>

Bit	Description
1:0	**16-Bit I/O Recovery Times (IOR16):** This 2-bit field defines the recovery time for 16-bit I/O as shown below. Programmable delays between back-to-back 16-bit PCI or CPU cycles to ISA I/O slaves are shown in terms of additional ISA clock recovery cycles (SYSCLK). The selected delay programmed into this field is enabled/disabled via bit 2 of this register. **Bits[1:0] — 16-Bit I/O Recover Time** 00 — 4 SYSCLKs 01 — 1 SYSCLKs (default) 10 — 2 SYSCLKs 11 — 3 SYSCLKs

3.3.10 PREV—PART REVISION REGISTER

Address Offset: 4Dh
Default Value: 00h
Attribute: Read/Write. Read Only
Size: 8 bits

This register provides the device stepping information for the IB and enables/disables DMA and ISA master accesses to DRAM BIOS locations E0000–EFFFFh. Bits 0 and 1 in this register are hardwired and write accesses have no effect.

Bit	Attribute	Description
7:5		**IB Fabrication House ID**
4	R/W	**E0000–EFFFFh ISA Forwarding Enable:** When bit 4 = 1 (and bit 6 in the XBCSA Register is set to 0), ISA master and DMA accesses to memory locations E0000–EFFFFh are forwarded to main memory. When bit 4 = 0 (default), ISA master and DMA accesses to this memory region are confined to the ISA Bus. Note that if bit 6 = 1 in the XBCSA Register, memory accesses to memory locations E0000–EFFFFh are always confined to the ISA Bus, regardless of the setting of bit 4 of this register.
3:0	RO	**Revision ID:** This field contains the device stepping information for the 82426EX IB.

3.3.11 XBCSA—X-BUS CHIP SELECT A REGISTER

Address Offset: 4Eh
Default Value: 03h
Attribute: Read/Write
Size: 8 bits

This register enables/disables accesses to the real time clock (RTC), keyboard controller (KBC), and BIOS. Disabling any of these bits prevents the chip select and X-Bus output enable control signal (XBUSOE#) for that device from being generated. The XBCSA Register also provides coprocessor error and mouse functions.

Bit	Description
7	**Extended BIOS Enable (EXBIOSE):** When EXBIOSE = 1 (enabled), CPU or PCI master accesses to locations FFF80000–FFFDFFFFh, that are not claimed by a PCI device are forwarded to the ISA Bus and results in the generation of BIOSCS# and XBUSOE#. Note that forwarding this region at the top of 4 GBytes to the ISA Bus (24-bit addressing) results in aliasing this 384 KByte region to the top of the 16 MByte ISA memory space. To avoid contention, ISA add-in memory must not be present in this space. When EXBIOSE = 0 (disabled: default), BIOSCS# or XBUSOE# are not generated. CPU accesses to locations FFF80000–FFFDFFFFh, that are not claimed by PCI devices are master-aborted by the PSC. Note that the Master Abort Status bit is set to 1 (DS Register). For reads, data of all 1's is returned to the CPU. PCI master accesses to locations FFF80000–FFFDFFFFh that are not claimed by PCI devices are ignored by the PSC.
6	**Lower BIOS Enable (LBIOSE):** When LBIOSE = 1 (enabled: default), CPU, PCI master, or ISA master accesses to the lower 64 KByte BIOS block at the top of 1 MByte (E0000–EFFFFh), or the aliases at the top of 4 GByte, that are not claimed by PCI devices, result in the generation of BIOSCS# and XBUSOE#. Note that forwarding this region at the top of 4 GBytes to the ISA Bus (24-bit addressing) results in aliasing this region to the top of the 16 MByte ISA memory space. To avoid contention, ISA add-in memory must not be present in this space. When LBIOSE = 0 (disabled: default), BIOSCS# or XBUSOE# are not generated during these accesses. Also, when LBIOSE = 0 (and bit 4 is 1 in the IB's PREV Register), ISA master or DMA accesses to this region are forwarded to main memory. CPU accesses to the lower 64 KByte BIOS region (0FFFE0000–0FFFEFFFFh) that are not claimed by PCI devices are master-aborted by the PSC. The Master Abort Status bit is set to 1 (DS Register). For reads, data of all 1's is returned to the CPU. PCI master accesses to the lower 64 KByte BIOS region (0FFFE0000–0FFFEFFFFh), that are not claimed by PCI devices, are ignored by the PSC.
5	**Coprocessor Error Function Enable (CPEE):** This bit enables/disables the coprocessor error support. When CPEE = 1 (enabled), the FERR# input, when asserted, triggers IRQ13 (internal). FERR# also gates the IGNNE# output. This bit defaults to disabled (0).
4	**IRQ12/M Mouse Function Enable (IRQ12/ME):** When bit 4 = 1, IRQ12/M provides the mouse function. When bit 4 = 0 (default), IRQ12/M provides the standard IRQ12 interrupt function.
3	**Reserved:** Must be set to 0.
2	**BIOS Memory Write Protect (BIOSWP):** When BIOSWP = 1 (read/write access), BIOSCS# is asserted for BIOS memory read and write cycles in the decoded BIOS region. When BIOSWP = 0 (write protect: default), BIOSCS# is only asserted for BIOS read cycles.
1	**Keyboard Controller Address Enable (KBCAE):** When KBCAE = 1 (enabled: default), the keyboard controller chip select signal (KBCCS#) and the XBUSOE# signals are generated for accesses to the keyboard controller address locations 60h, 62h, 64h, and 66h. When KBCAE = 0 (disabled), KBCCS# and XBUSOE# are not generated for these accesses.
0	**RTC Address Enable (RTCAE):** When RTCAE = 1 (enabled: default), the RTCCS#, RTCALE, and XBUSOE# signals are generated for accesses to the RTC address locations 70–77h. When RTCAE = 0 (disable), the RTCCS#, RTCALE, and XBUSOE# signals are not generated for accesses to these addresses.

3.3.12 HOSTSEL—HOST SELECT REGISTER

Address Offset: 50h
Default Value: 00000xx0 (x = Depends on hardware strapping options)
Attribute: Read/Write
Size: 8 Bits

The HOSTSEL Register enables/disables the L1 cache, indicates the clock configuration selected by hardware strapping options, and selects the L1 caching policy.

Bit	Description
7:4	**Reserved**
3	**L1 Caching Policy Select (L1CPSEL):** L1CPSEL selects the caching policy for the L1 cache. When L1CPSEL = 1, the L1 caching policy is write-back and when L1CPSEL = 0, the caching policy is write-through.
2:1	**Clock Configuration Status (CLKCONFS):** Clock configuration is determined by hardware strapping options on the SIDLE# and CMDV# signal pins at power-up (see Section 4.15, Clocks). This field is read only. CLKCONFS indicates the clock mode and frequencies selected by the strapping options. **Bits[2:1] — HCLK — PCICLK — Clock Mode** 00 — 25 MHz — 25 MHz — 1x 01 — 33 MHz — 33 MHz — 1x 10 — Reserved 11 — Reserved
0	**L1 Cache Enable (L1CE):** L1CE enables/disables the first level cache in the CPU. When L1CE = 1 (enable), the PSC responds to the CPU with KEN# asserted for cacheable memory cycles. When L1CE = 0 (disable), the KEN# signal is always negated to the CPU. This prevents new cache line fills to either the first level or second level caches.

3.3.13 DFC—DETURBO FREQUENCY CONTROL REGISTER

Address Offset: 51h
Default Value: 80h
Attribute: Read /Write
Size: 8 Bits

Some old software packages that rely on the operating speed of the processor do not work on today's faster systems. To maintain backward compatibility with these software packages, the 82420EX PCIset provides a mechanism to emulate the operating speed of PC/AT systems. This emulation is achieved with the deturbo mode (enabled/disabled via the Turbo/Reset Control Register). When the deturbo mode is enabled, the PSC periodically asserts the HOLD signal to slow down the effective speed of the CPU. The frequency of the HOLD assertion is fixed to once in 1024 Host Clocks. The duty cycle of the HOLD active period is controlled by the DFC Register.

Bit	Description
7:0	**Deturbo Mode Frequency Adjustment Value:** This 8-bit value effectively defines the duty cycle of the HOLD signal. The value programmed into this register is compared against a free running 8-bit counter running at 1/4 the CPU clock. When the counter is greater than the value specified in this register, HOLD is asserted to the CPU. HOLD is negated when the counter value is equal to or smaller than the contents of this register. HOLD is negated when the counter rolls over to 00h. Note that the deturbo counter does not start until HLDA is returned by the CPU. The deturbo emulation speed is directly proportional to the value in this register. The smaller the value in this register the lower the deturbo emulation speed.

3.3.14 SCC—SECONDARY (L2) CACHE CONTROL REGISTER

Address Offset: 52–53h
Default Value: 0000h
Attribute: Read /Write
Size: 16 Bits

This 16-bit register defines the L2 cache operations. SCC enables/disables the L2 cache, adjusts cache size, selects the cache write policy, defines the cache SRAM type, and selects various read/write cache cycle times. In addition, a cache miss can be forced for each access permitting software to initialize the cache. Cache hits can also be forced permitting software to determine the size of the L2 cache memory.

NOTE:
The L2 timings must be programmed at least as fast as the DRAM timings.

Bit	Description
15:13	**Reserved**
12	**Hit Dirty Write Cycle Timing (HDWRTIME):** When HDWRTIME = 1, the PSC performs 0 wait-state accesses (2-1-1-1) for hit dirty write cycles. When HDWRTIME = 0 (default), the access time for write hit dirty cycles is determined by WRTIME (bit 11). See Section 3.3.14.1.
11	**Write Cycle Timing (WRTIME):** When WRTIME = 1, the timing for PSC L2 cache write accesses is 3-2-2-2. In this case, the WRTIME bit is ignored for write hit dirty cycles. When WRTIME = 0 (default), the timing for PSC L2 cache write accesses is 4-2-2-2. See Section 3.3.14.1.
10:9	**Subsequent Read Timing (SUBRD):** This field determines the access time for subsequent reads to the L2 cache as shown below. **Bits[10:9] — Subsequent Cache Read Timing** 00 — X-3-3-3 (default) 01 — X-2-2-2 10 — X-1-1-1 11 — Reserved
8	**Initial Read Timing (INITRD):** This bit determines the access time for initial reads to the L2 cache. When INITRD = 1, the initial read timing is 2-X-X-X. When INITRD = 0 (default), the initial read timing is 3-X-X-X.

Bit	Description
7	**Reserved**
6	**Cache Force Hit (L2FHIT):** When L2FHIT = 0, the cache operation is normal. When L2FHIT = 1, the L2 data SRAMs are accessed, for cacheable data reads and writes, as if they were main memory. Since all data reads and writes hit the cache, none of the data cycles go to main memory. Thus, BIOS can determine the L2 cache size and configuration during POST. While L2FHIT = 1 forces a hit for cacheable data cycles, all code reads are forced to be non-cacheable. Thus, the CPU can perform code read cycles from main memory without the L2 cache generating write-back cycles and without the L2 generating code-read-allocate cycles (that may interfere with the L2 sizing algorithm). When in L2FHIT mode, the primary (L1) cache must be disabled. L2 configuration is determined by setting the cache in Interleaved mode and performing line write/read. The L2 cache size is determined by setting L2SIZE (from 64 KBytes and up) and performing a write to location (K + cache-size) and a read to location (K). When L2SIZE = 000, the L2FHIT bit has no effect.
5	**L2 Cache Force Miss Clean (L2FMISS):** When L2FMISS = 0 (default), the L2 Cache operation is normal. When L2FMISS = 1, all cacheable accesses to L2 are forced to be a cache miss. This bit is used to initialize the cache with valid locations. BIOS can set this bit and read a block of main memory equal to the cache size. This fills the cache with valid data. Once the cache is initialized, software sets this bit to 0, and the PSC keeps the cache coherent with main memory. When L2SIZE = 000 (L2 disabled), the L2FMISS bit has no effect.
4	**Cache Configuration (L2CONF):** This bit determines the configuration of the L2 cache SRAMs. For an interleaved memory configuration, L2CONF = 1 and for a non-interleaved configuration, L2CONF = 0 (default).
3	**Cache Write Policy (L2WPOL):** This bit determines the L2 cache policy. When WRPOL = 1, the L2 cache policy is write-back. When WRPOL = 0 (default), the cache policy is write-through.
2:0	**Cache Size (L2SIZE):** This field determines the L2 cache size as shown below.

Bits[2:0]	Cache Size	Bits[2:0]	Cache Size
000	L2 Cache Disabled (default)	100	512 KBytes
001	64 KBytes	101	Reserved
010	128 KBytes	110	Reserved
011	256 KBytes	111	Reserved

3.3.14.1 L2 Write Timing

Bits 11 and 12 control the write timing for the L2 cache controller. Bit 12 = 1, bit 11 = 0 is an invalid combination which will cause the PSC to lock up. The following table shows the various bit 11, 12 combinations and how they program the L2 cache controller write timings.

Bit 12	Bit 11	L2 Cache Write Timing
0	0	4-2-2-2
0	1	3-2-2-2
1	0	Invalid
1	1	2-1-1-1 (hit dirty write cycles)

3.3.15 DRAMC—DRAM CONTROL REGISTER

Address Offset: 56–57h
Default Value: 0000h
Access: Read/Write
Size: 16 bits

The DRAMC Register selects various DRAM interface timing parameters. This register also enables/disables fast page mode for DRAM access, enables/disables CAS pipelining, and provides a refresh test option.

NOTE:
The L2 timings must be programmed at least as fast as the DRAM timings.

Bit	Description
15	**Muxed Address Hold Time (MAH):** This bit determines the number of clocks from RAS# or CAS# active before MA can be changed. When MAH is 1, the hold time is 0.5 active clocks and, when MAH is 0 (default), the hold time is 1.0 active clock.
14	**Muxed Address Setup Time (MASU):** This bit determines the number of clocks from Muxed Address driven to RAS# or CAS# active. When MASU is 1, the address setup is 0.5 active clocks and, when MASU is 0 (default), the address setup is 1.0 active clocks.
13	**CAS Write Timing (CASWR):** This bit determines the number of clocks CAS# remains active during a write access, and inactive between accesses as shown below. **Bit 13** — **Active, Inactive Clocks** 0 — 2 Active, 1 Inactive (default) 1 — 1 Active, 1 Inactive
12:11	**CAS Read Timing (CASRD):** This field determines the number of clocks CAS# remains active during a read access and inactive between accesses as shown below. **Bits[12:11]** — **Active, Inactive Clocks** 00 — 3 Active, 1 Inactive (default) 01 — 2 Active, 1 Inactive 10 — 1.5 Active, 0.5 Inactive 11 — 1 Active, 1 Inactive
10:8	**RAS Precharge Timing (RASPRE):** This field determines the minimum number of Host Bus clocks that RAS# remains inactive as shown below. **Bits[10:8]** — **RAS Pre-Charge Time** — **Bits[10:8]** — **RAS Pre-Charge Time** 000 — 4 Clocks (default) — 100 — 1.5 Clocks 001 — 3 Clocks — 101 — 1 Clock 010 — 2 Clocks — 110 — Reserved 011 — Reserved — 111 — Reserved
7:6	**Reserved**
5	**Pipelined CAS Enable (PCASEN):** When PCASEN = 1, the DRAM controller does not provide any time between CAS[3:0] # negation and CAS[7:4] # assertion (interleaved row only). When PCASEN = 0 (default), the DRAM controller provides 1 Host Bus clock between CAS[3:0] # negation and CAS[7:4] # assertion (interleaved row only).

Bit	Description
4	**Refresh Test Enable (REFTSTE):** When REFTST = 1, a test mode for the refresh generator is enabled. In this mode, a refresh request is generated every 32 HCLK cycles. When REFTST = 0 (default), the DRAM controller generates a refresh cycle every 15 μs.
3	**Fast Page Write Enable (FPWE):** This bit permits the PSC to keep the currently accessed DRAM page active following a CPU write cycle. When FPWE = 1, the PSC keeps the page open (keeps RAS# asserted) following a write cycle to main memory. When FPWE = 0 (default), the PSC closes the page (negates RAS#) following a write cycle to main memory, creating a row miss for every CPU write.
2	**Fast Page Data Read Enable (FPDRE):** This bit permits the PSC to keep the currently accessed DRAM page active following a CPU data read cycle. When FPDRE = 1, the PSC keeps the page open (keeps RAS# asserted) following a data read cycle to main memory. When FPDRE = 0 (default), the PSC closes the page (negates RAS#) following a data read cycle to main memory, creating a row miss for every CPU data read.
1	**Fast Page Code Read Enable (FPCRE):** This bit permits the PSC to keep the currently accessed DRAM page active following a CPU code read cycle. When FPCRE = 1, the PSC keeps the page open (keeps RAS# asserted) following a code read cycle to main memory. When FPCRE = 0 (default), the PSC closes the page (negates RAS#) following a code read cycle to main memory, creating a row miss for every CPU code read.
0	**Reserved**

3.3.16 PAM[6:0]—PROGRAMMABLE ATTRIBUTE MAP REGISTERS

Address Offset: PAM6(5Fh), PAM5(5Eh), PAM4(5Dh), PAM3(5Ch)
PAM2(5Bh), PAM1(5Ah) PAM0(59h)
Default Value: PAM[6:0] = 00h
Attribute: Read/Write

The 82420EX PCIset allows programmable memory and cacheability attributes on 13 memory segments of various sizes in the ISA compatibility hole—640 KByte to 1 MByte address range. Seven Programmable Attribute Map (PAM) Registers support these features. Four bits specify cacheability and memory attributes for each memory segment. These attributes are:

RE Read Enable. When RE = 1, the CPU read accesses to the corresponding memory segment are directed to main memory. Conversely, when RE = 0, the CPU read accesses are forwarded to PCI and, if not claimed on PCI, are forwarded to ISA.

WE Write Enable. When WE = 1, the PCI write accesses to the corresponding memory segment are directed to main memory. Conversely, when WE = 0, the CPU write accesses forwarded to PCI and, if not claimed on PCI, are forwarded to ISA.

CE Cache Enable. When CE = 1, the corresponding memory segment is cacheable. It is illegal to set CE = 1 and RE = 0 for the same segment. When CE = 1 and WE = 0, the corresponding memory range is not cached in the L1 cache (KEN# is negated on CPU accesses). However, it is cached and write protected in the L2 cache. The L2 cache handles cached write protected ranges as follows:

Code read (L2 miss): L2 line is allocated, data is read from main memory.
Data read (L2 miss): data is read from main memory.
Any read (L2 hit): data is read from the L2 cache
Any write: subtractively decoded to PCI Bus.

PE PCI Enable. When PE = 1, the corresponding memory range is accessible by PCI masters, as a function of the RE, WE and CE bits setting. When PE = 0, the corresponding memory range is inaccessible by PCI masters (the PCI master cycles are either claimed by PCI slaves or sent to ISA).

Each PAM Register controls two ranges as shown in Table 9.

NOTE:
The combination RE = 0 and CE = 1 is illegal.

Table 9. PAM Registers and Associated Memory Ranges

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

DOS Application Area (00000h–9FFFFh)

The 640 KByte DOS application area always has read, write, and cacheability attributes enabled and are not programmable for the 0–640 KByte region.

Video Buffer Area (A0000h–BFFFFh)

This 128 KByte area is not controlled by attribute bits. It is always subtractively decoded to ISA.

Expansion Area (C0000h–DFFFFh)

This 128 KByte area is divided into eight 16 KByte segments. Each segment can be assigned one of four Read/Write states: read-only, write-only, read/write, or disabled (memory that is disabled is not remapped elsewhere). Cacheability status can also be specified for each segment.

Extended System BIOS Area (E0000h–EFFFFh)

This 64 KByte area is divided into four 16 KByte segments. Each segment can be assigned independent cacheability, read, and write attributes. Memory segments that are disabled are not remapped elsewhere.

System BIOS Area (F0000h–FFFFFh)

This area is a single 64 KByte segment. This segment can be assigned cacheability, read, and write attributes and PCI enabled.

Extended Memory Area (100000h–FFFFFFFFh)

The extended memory area can be split into several parts;

- BIOS area from 4 GByte to (4 GByte minus 512 KByte) (aliased on ISA at 16 MByte minus 15.5 MByte)
- Main memory from 1 MByte to a maximum of 128 MBytes
- PCI memory space from TOM to 128 MBytes or, (2 GBytes minus 128 MBytes) to (2 GByte plus 128 MByte), or 4 GByte to (4 GByte minus 128 MByte)

On power-up or reset the CPU vectors to the Flash BIOS area, mapped in the range of 4 GByte to 4 GByte–512 KByte. However, this area is physically mapped on ISA. Since these addresses are in the upper 4 GByte range, the request is directed to PCI. The 82420EX PCIset strips the upper address bits to effectively map the BIOS on ISA in the area between 16 MByte to 15.5 MByte.

The main memory space can occupy extended memory from a minimum of 1 MByte up to 128 MBytes. This memory is cacheable. The following areas may be occupied by PCI memory: the address space on PCI from TOM to 128 MBytes, between the Flash BIOS (4 GByte minus 512 KByte) and (4 GByte minus 128 MByte), and the range from (2 GBytes minus 128 MBytes) to (2 GByte plus 128 MByte) may be occupied by PCI memory. This memory space is not cacheable.

3.3.17 DRB—DRAM ROW BOUNDARY REGISTERS

Address Offset: DRB4(64h), DRB3(63h) DRB2(62h)
DRB1(61h), DRB0(60h)
Default Value: 01h (for each DRB)
Access: Read/Write
Size: 8 bits

The PSC supports up to 5 rows of DRAM. When populated, each row contains 32 (non-interleaved) or 64 (interleaved) bits of data. The DRAM Row Boundary registers define upper and lower addresses for each DRAM row. Contents of these 8-bit registers represent the amount of memory in MBytes.

DRB0 = Total amount of memory in row 0 (in MBytes)
DRB1 = Total amount of memory in row 0 + row 1 (in MBytes)
DRB2 = Total amount of memory in row 0 + row 1 + row 2 (in MBytes)
DRB3 = Total amount of memory in row 0 + row 1 + row 2 + row 3 (in MBytes)
DRB4 = Total amount of memory in row 0 + row 1 + row 2 + row 3 + row4 (in MBytes)

The DRAM array can be configured with SIMMs that have address depths of 256 KByte, 1 MByte, and 4 MByte. Each register defines an address range that causes a particular RAS# line to assert (e.g. if the first DRAM row is 2 MBytes in size, then accesses within the 0 to 2 MByte range causes the RAS0# line to be asserted). The DRAM Row Boundary (DRB) Registers are programmed with an 8-bit upper address limit value.

Bit	Description
7:0	**Memory Boundary in MBytes:** This 8-bit value is used to determine the upper address limit of this row (i.e., this row - previous row = row size). Unpopulated rows have a value equal to the previous row (row size = 0). The value programmed into DRB4 always reflects the maximum amount of DRAM in the system.

Example 1:
If SIMM0 contains a 256K x 36 SIMM,(which is equivalent to 1 MByte DRAM), DRB0 is set to 01h. If this is the only SIMM in the system, DRB[4:1] are each set to 01h.

Example 2:
One way to achieve maximum main memory is to populate SIMMs 0-3 with 8M x 36 double-sided SIMMs (which have 32 MBytes each). In this case, DRB[4:0] would be programmed as follows: DRB0 = 20h, DRB1 = 40h, DRB2 = 60h, DRB3 = 80h, DRB4 = 80h.

3.3.18 PIRQ1RC/PIRQ0RC—PIRQ ROUTE CONTROL REGISTERS

Address Offset: 66h (PIRQ0RC)
67h (PIRQ1RC)
Default Value: PIRQ0RC 80h
PIRQ1RC 80h
Attribute: Read/Write
Size: 8 bits

The PIRQ1RC/PIRQ0RC Registers control the routing of PIRQ[1:0] signals to the internal IRQ inputs of the interrupt controller. Each PIRQx# can be independently routed to any one of 11 interrupts. One or both PIRQx# lines can be routed to the same IRQx input. Note that the IRQ selected through bits[3:0] must be set to level sensitive mode in the corresponding ELCR Register.

Bit	Description
7	**PIRQx Interrupt Signal Routing Enable:** When bit 7 = 0 (enabled), PIRQx# is routed to the IRQ selected by bits[3:0] of this register. When bit 7 = 1 (disabled: default), the PIRQx# signal is not routed to any IRQ line.
6:4	**Reserved:** Read as zeros.
3:0	**PIRQx Interrupt Signal Routing:** When bit 7 = 0, bits[3:0] select how each PIRQx# is routed to each internal 8259 IRQx. The routing for different values of this field are shown below.

Bits[3:0]	PIRQx Interrupt Routing	Bits[3:0]	PIRQx Interrupt Routing
0000	Reserved (default)	1000	Reserved
0001	Reserved	1001	IRQ9
0010	Reserved	1010	IRQ10
0011	IRQ3	1011	IRQ11
0100	IRQ4	1100	IRQ12
0101	IRQ5	1101	Reserved
0110	IRQ6	1110	IRQ14
0111	IRQ7	1111	IRQ15

3.3.19 DMH—DRAM MEMORY HOLE REGISTER

Address Offset: 68h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

The DMH Register defines a hole in main memory between 1 MByte and 16 MBytes. ISA memory accesses to the region defined by the memory hole are not forwarded to main memory. The ISA cycle is confined to the ISA Bus.

Bit	Description
7	**Memory Hole Enable (MHE):** When MHOLEE = 1, the memory hole is enabled and all ISA master and DMA accesses within the programmed hole are confined to the ISA Bus. All CPU and PCI master accesses within the hole are forwarded to the PCI/ISA Bus. When MHOLEE = 0 (default), the memory hole is disabled.
6:4	**Memory Hole Size (MHSIZE):** This field selects the memory hole size as shown in the table below.
3:0	**Memory Hole Start Address (MHSTRT):** This four bit field defines the starting address of the memory hole. Bits[3:0] correspond to A[23:20], respectively. The memory hole starting address can be between 1 MByte and 16 MBytes, with 1 MByte granularity. Note that the top of the memory hole range must be below 16 MBytes. It is the responsibility of the BIOS to set the hole size and starting address accordingly.

Bits[6:4]	Memory Hole Size	Bits[6:4]	Memory Hole Size
000	1 MByte (default)	100	Reserved
001	2 MBytes	101	Reserved
010	Reserved	110	Reserved
011	4 MBytes	111	8 MBytes

3.3.20 TOM—TOP OF MEMORY

Address Offset: 69h
Default Value: 02h
Attribute: Read/Write
Size: 8 bits

The 82420EX PCIset supports up to 128 MBytes of system memory. The Top Of Memory Register must be set by the BIOS to the value of the DRB4 Register plus the memory hole size. For example, the top of memory for a system with 16 MBytes of DRAMs, and a 1 MByte Hole (somewhere between 1 and 16 MBytes), is at 17 MBytes.

The TOM Register is programmed with an 8-bit upper address limit value. This upper address limit is compared to A[31:30,26:20] of the Host address bus to determine if main memory is being targeted. When A[31:30,26:20] <TOM, and the access is not to the memory hole, main memory is being targeted. Otherwise, a PCI or ISA region is being targeted. Bits[7:0] of this register correspond to A[31:30,26:20].

Note that SMRAM can be placed at the top of memory between TOM-64 KByte and TOM. Note, also, that the maximum supported DRAM size is 128 MBytes minus the Memory Hole size.

For use with operating systems other than Windows*, DOS*, and OS/2*, the TOM register should not be programmed with a value of greater than 127M.

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

3.3.21 SMRAMCON—SMRAM CONTROL REGISTER

Address Offset: 70h
Default Value: 00h
Attribute: Read/Write
Size: 8 Bits

The SMRAMCON Register sets the SMRAM location and attributes. SMRAM is always located in main memory and it is always non-cacheable. The memory block that shares the same bus address range with SMRAM is also non-cacheable (even if it is defined as cacheable by other configuration settings).

Bit	Description
7	**Reserved**
6	**SMRAM Space Open (SMOPN):** This bit manually opens SMRAM space. When SMOPN = 1, CPU accesses (only CPU accesses) to SMRAM are re-mapped (see SMBASE description). BIOS uses this bit to manually open the SMRAM when the CPU is not in SMM. SMOPN is used by the BIOS to initialize the SMRAM. Setting the SMOPN bit to 1 has no effect when the CPU is in SMM (SMLCK bit is 1) or when the CPU is not the current system master (HLDA = 1).
5	**SMRAM Close (SMCLS):** This bit manually closes SMRAM space. The SMI handler uses SMCLS to access the physical memory block that shares the same bus address range with SMRAM. When SMCLS = 1, re-mapping of SMRAM (code and data) is disabled. This permits the CPU to access the data in system memory that is aliased by SMM memory, even when the CPU is in SMM. Note that SMCLS affects data accesses only; code read cycles are not affected.
4	**SMRAM Lock (SMLCK):** SMLCK locks SMRAM space from manual opening. When SMLCK = 1, the SMOPN function is disabled, as well as write protecting the SMBASE. The SMLCK bit is a write once bit. This means that once set, this bit cannot be cleared by software. Only a CPURST clears this bit. SMLCK permits BIOS, after initialization is complete, to protect the SMRAM from other programs. Once SMLCK is set to 1, no manual opens of the SMRAM are possible.
3	**Reserved**
2:0	**SMRAM Base Address (SMBASE):** SMBASE selects the SMRAM segment. Based on this selection, the SMRAM address is re-mapped as shown below. The setting of the SMRAM range forces the CPU bus range to be non-cacheable, regardless of other bit settings. The following table describes some SMBASE values and attributes.

Bits[2:0]	CPU Range	DRAM Range	Non-CPU Cycles Are Sent to:	Comments
000				Reserved
001				Reserved
010	A0000–AFFFFh	A0000–AFFFFh	Subtractively to PCI/ISA	PCI/ISA graphic frame buffer region. Region can not be used as SMRAM if it is also used as a graphic frame buffer of a Host Bus device.
011	B0000–BFFFFh	B0000–BFFFFh		
100	C0000–CFFFFh	A0000–AFFFFh	Main memory or subtractively PCI/ISA, Function of registers setting.	
101	D0000–DFFFFh	A0000–AFFFFh		
110	E0000–EFFFFh	A0000–AFFFFh		
111	F0000–FFFFFh	A0000–AFFFFh		

3.3.22 SMICNTL—SMI CONTROL REGISTER

Address Offset: A0h
Default Value: 08h
Attribute: Read/Write
Size: 8 Bits

The SMICNTL Register provides Fast Off Timer control, STPCLK# enable/disable, and throttle control. This register also enables/disables the system management interrupt (SMI).

NOTE:
Bits[4:3] = 01 can be used to freeze the Fast Off Timer when in SMM. Freezing the Fast Off Timer prevents time-outs from occurring while executing SMM code. This prevents the system from being confused by asynchronous events that could happen while servicing SMM code.

Bit	Description
7:5	**Reserved**
4:3	**Fast Off Timer Control (CTMRCNTL):** This field enables/disables the Fast Off Timer and when enabled, selects the timer's counting granularity as shown below. **Bits[4:3]** — **Count Granularity for 33 MHz Host Bus Operation** — **Count Granularity for 25 MHz Host Bus Operation** 00 — 1 Minute — 1.32 Minutes 01 — Disabled (default) — Disabled (default) 10 — 1 HCLKIN — 1 HCLKIN 11 — 1 msec — 1.32 msec
2	**STPCLK# Signal Throttle Enable (CSTPCLKTHE):** This bit enables/disables control of the STPCLK# high/low times by the clock throttle timers. When bit 2 = 1, the STPCLK# signal throttle control is enabled. When enabled (and bit 1 = 1, enabling the STPCLK# signal), the high and low times for the STPCLK# signal are controlled by the Clock Throttle STPCLK# High Timer and Clock Throttle STPCLK# Low Timer Registers, respectively. When bit 2 = 0 (default), the throttle control of the STPCLK# signal is disabled.
1	**STPCLK# Signal Enable (CSTPCLKE):** This bit permits software to place the CPU into a low power state. When bit 1 = 1, the STPCLK# signal is enabled and a read from the APMC Register causes STPCLK# to be asserted. When bit 1 = 0 (default), the STPCLK# signal is disabled and is negated (high). Software can set this bit to 0 by writing a 0 to it.
0	**SMI# Gate (CSMIGATE):** When bit 0 = 1, the SMI# signal is enabled and a system management interrupt condition causes the SMI# signal to be asserted. When bit 0 = 0 (default), the SMI# signal is masked and negated. This bit only affects the SMI# signal and does not affect the detection/ recording of SMI events (i.e., this bit does not affect the SMI status bits in the SMIREQ Register). Thus, SMI conditions can be pending when this bit is set to 1. If an SMI is pending when this bit is set to 1, the SMI# signal is asserted.

3.3.23 SMIEN—SMI ENABLE REGISTER

Address Offset: A2–A3h
Default Value: 0000h
Attribute: Read/Write
Size: 16 Bits

This register enables the generation of SMI (asserting the SMI# signal) for the associated hardware events (bits[5:0]), external SMI signal (bit 6), and software events (bit 7). When a hardware event is enabled, the occurrence of a corresponding event results in the assertion of SMI#, if enabled via the SMICNTL Register. The SMI# is asserted independent of the current power state (Power-On or Fast Off). The default for all sources in this register is disabled.

Bit	Description
15:8	**Reserved**
7	**APMC Write SMI Enable:** This bit enables SMI for writes to the APMC Register. When bit 7 = 1, writes to the APMC Register generate an SMI. When bit 7 = 0, writes to the APMC Register do not generate an SMI.
6	**EXTSMI# Signal SMI Enable:** When bit 6 = 1, asserting the EXTSMI# input signal generates an SMI. When bit 6 = 0, asserting EXTSMI# does not generate an SMI.
5	**Fast Off Timer SMI Enable:** This bit enables the Fast Off Timer to generate an SMI. When bit 5 = 1, the timer generates an SMI when it decrements to zero. When bit 5 = 0, the timer does not generate an SMI.
4	**IRQ12 SMI Enable (PS/2 Mouse Interrupt):** This bit enables the IRQ12 signal to generate an SMI. When bit 4 = 1, asserting the IRQ12 input signal generates an SMI. When bit 4 = 0, asserting IRQ12 does not generate an SMI.
3	**IRQ8 SMI Enable (RTC Alarm Interrupt):** This bit enables the IRQ8 signal to generate an SMI. When bit 3 = 1, asserting the IRQ8 input signal generates an SMI. When bit 3 = 0, asserting IRQ8 does not generate an SMI.
2	**IRQ4 SMI Enable (COM2/COM4 Interrupt or Mouse):** This bit enables the IRQ4 signal to generate an SMI. When bit 2 = 1, asserting the IRQ4 input signal generates an SMI. When bit 2 = 0, asserting IRQ4 does not generate an SMI.
1	**IRQ3 SMI Enable (COM1/COM3 Interrupt or Mouse):** This bit enables the IRQ3 signal to generate an SMI. When bit 1 = 1, asserting the IRQ3 input signal generates an SMI. When bit 1 = 0, asserting IRQ3 does not generate an SMI.
0	**IRQ1 SMI Enable (Keyboard Interrupt):** This bit enables the IRQ1 signal to generate an SMI. When bit 0 = 1, asserting the IRQ1 input signal generates an SMI. When bit 0 = 0, asserting IRQ1 does not generate an SMI.

3.3.24 SEE—SYSTEM EVENT ENABLE

Address Offset: A4–A7h
Default Value: 00000000h
Attribute: Read/Write
Size: 32 Bits

This register enables hardware events as system events or break events for power management control. Note that all of the functional bits in the SEE Register provide system event control. In addition, all bits provide break event control. The default for each system/break event in this register is disabled.

System Events: Activity by these events can keep the system from powering down. When a system event is enabled, the corresponding hardware event activity prevents a Fast Off powerdown condition. Anytime the corresponding hardware event occurs (signal is asserted), the Fast Off Timer is re-loaded with its initial count.

Break Events: These events can awaken a powered down system. When a break event is enabled, the corresponding hardware event activity powers up the system by negating STPCLK#. Note that STPCLK# is not negated until the stop grant special cycle has been generated by the CPU. Thus, from the time that STPCLK# is asserted until the stop grant cycle is returned, the occurrence of subsequent break events are latched in the IB.

NOTE:

Bit 30 in this register is used as a global break event and should be set to 1 if ISA cards that generate IRQ's by driving them low and then high and keeping them high until another interrupt is generated, are supported. Refer to the bit description.

SRESET is always enabled as a break event. However, SRESET only causes a break event after a stop grant special cycle has been received. If SRESET is asserted while STPCLK# is active and then negated before the stop grant cycle is received, SRESET does not cause a break event.

Bit	Description
31	**Fast Off SMI Enable (FSMIEN):** When bit 31 = 1 (enabled), an SMI causes a system event that re-loads the Fast Off Timer and a break event that negates the STPCLK# signal. When bit 31 = 0 (disabled), an SMI does not re-load the Fast Off Timer or negate the STPCLK# signal.
30	**INTR Enable (FINTREN):** When bit 30 of this register is set to 1, INTR will be used as a global break event. In this case, any IRQ that is generated will cause the system to power-up via the negation of STPCLK#, regardless of the state of bits 0, 1, and 3 through 15 in this register. When this bit is set to 0, INTR is not used as a break event and bits 0, 1, and 3 through 15 can be used to individually enable/disable break events. Note that this bit has no effect on the setting of system events and only effects the break event function.
29	**Fast Off NMI Enable (FNMIEN):** When bit 29 = 1 (enabled), an NMI (e.g., parity error) causes a system event that re-loads the Fast Off Timer and a break event that negates the STPCLK# signal. When bit 29 = 0 (disabled), an SMI does not re-load the Fast Off Timer or negate the STPCLK# signal.
28:16	**Reserved**

Bit	Description
15:3	**Fast Off IRQ[15:3] Enable:** These bits are used to prevent the system from entering Fast Off and break any current powerdown state when the selected hardware interrupt occurs. When a bit = 1 (enabled), the corresponding interrupt causes a system event that re-loads the Fast Off Timer and a break event that negates the STPCLK# signal. When a bit = 0 (disabled), the corresponding interrupt does not re-load the Fast Off Timer or negate the STPCLK# signal.
2	**Reserved**
1:0	**Fast Off IRQ[1:0] Enable:** These bits are used to prevent the system from entering Fast Off and break any current powerdown state when the selected hardware interrupt occurs. When a bit = 1, the corresponding interrupt causes a system event that re-loads the Fast Off Timer and a break event that negates the STPCLK# signal. When a bit = 0 (disabled), the corresponding interrupt does not re-load the Fast Off Timer or negate the STPCLK# signal.

3.3.25 FTMR—FAST OFF TIMER REGISTER

Address Offset: A8h
Default Value: 0Fh
Attribute: Read/Write
Size: 8 Bits

The Fast Off Timer is used to indicate (through an SMI) that the system has been idle for a pre-programmed period of time. The Fast Off Timer consists of a count-down timer and the count down value programmed into this register. The Fast Off Timer count down value is (x + 1) where x equals the value programmed in the Fast Off Timer register and the unit of measurement is in minutes or msec, depending on the value of bits 4–3 in the SMI Control register. The Fast Off Timer count down value is loaded into the Fast Off Timer when an enabled system event occurs. When the timer expires, an SMI special cycle is generated. When the Fast Off Timer is enabled (bits[4:3] = 00, 10, or 11 in the SMICNTL register), the timer counts down from the Fast Off Timer count down value. The count time interval is programmable (via the SMICNTL Register). When the Fast Off Timer reaches 00h, an SMI is generated and the timer is re-loaded with the Fast Off Timer count down value. If an enabled system event occurs before the Fast Off Timer reaches 00h, the Fast Off Timer is re-loaded with the Fast Off Timer count down value. Note that the Fast Off Timer should never be programmed to a value of 00h.

NOTE:
Before writing to the FTMR Register, the Fast Off Timer must be stopped by setting bits[4:3] to 01 in the SMICNTL Register. The Fast Off Timer will begin decrementing when the these bits are subsequently set to 00, 10, or 11.

Bit	Description
7:0	**Fast Off Timer Value:** Bits[7:0] contain value x, where the Fast Off Timer count down value is (x + 1). A read from the FTMR Register returns the value last written.

3.3.26 SMIREQ—SMI REQUEST REGISTER

Address Offset: AA–ABh
Default Value: 00h
Attribute: Read/Write
Size: 16 Bits

The SMIREQ Register contains status bits indicating the cause of an SMI. When an enabled event causes an SMI, the IB automatically sets the corresponding event's status bit to 1. Software sets the status bits to 0 by writing a 0 to them.

The SMI handler can query the status bits to see what caused the SMI and then branch to the appropriate routine. As the individual routines complete, the handler resets the appropriate status bit by writing a 0 to the corresponding bit.

Each of the SMIREQ bits is set by the IB in response to the activation of the corresponding SMI event. If the SMI event is still active when the corresponding SMIREQ bit is set to 0, the IB does not set the status bit back to a 1 (i.e., there is only one status indication per active SMI event).

When an IRQx signal is asserted, the corresponding RIRQx bit is set to a 1. If the IRQx signal is still active when software sets the RIRQx bit to 0, RIRQx is not set back to a 1. The IRQx may be negated before software sets the RIRQx bit to 0. If the RIRQx bit is set to 0 at the same time a new IRQx is activated, RIRQx remains set to 1. This indicates to the SMI handler that a new SMI event has been detected.

NOTE:
The SMIREQ bits are set, cleared, or read independently of each other and independently of the CSMIGATE bit in the SMICNTL Register.

Bit	Description
15:8	**Reserved**
7	**APM SMI Status (RAPMC):** The IB sets this bit to 1 to indicate that a write to the APM Control Register caused an SMI. Software sets this bit to a 0 by writing a 0 to it.
6	**EXTSMI# SMI Status (REXT):** The IB sets this bit to 1 to indicate that EXTSMI# caused an SMI. Software sets this bit to a 0 by writing a 0 to it.
5	**Fast Off Timer Expired Status (RFOT):** The IB sets this bit to 1 to indicate that the Fast Off Timer expired and caused an SMI. Software sets this bit to a 0 by writing a 0 to it.
4	**IRQ12 Request SMI Status (RIRQ12):** The IB sets this bit to 1 to indicate that IRQ12 caused an SMI. Software sets this bit to a 0 by writing a 0 to it.
3	**IRQ8# Request SMI Status:** The IB sets this bit to 1 to indicate that IRQ8# caused an SMI. Software sets this bit to a 0 by writing a 0 to it.
2	**IRQ4 Request SMI Status:** The IB sets this bit to 1 to indicate that IRQ4 caused an SMI. Software sets this bit to a 0 by writing a 0 to it.
1	**IRQ3 Request SMI Status:** The IB sets this bit to 1 to indicate that IRQ3 caused an SMI. Software sets this bit to a 0 by writing a 0 to it.
0	**IRQ1 Request SMI Status:** The IB sets this bit to 1 to indicate that IRQ1 caused an SMI. Software sets this bit to a 0 by writing a 0 to it.

3.3.27 CTLTMRL—CLOCK THROTTLE STPCLK# LOW TIMER

Address Offset: ACh
Default Value: 00h
Attribute: Read/Write
Size: 8 Bits

The duration of the STPCLK# asserted period when bit 2 in the SMICNTL Register is set to 1 is (x + 1) where x equals the value programmed in this register. The value in this register plus 1 is loaded into the STPCLK# Timer when STPCLK# is asserted. However, the timer does not start until the Stop Grant Bus Cycle is received. The STPCLK# timer counts using a 32 μs clock.

Bit	Description
7:0	**Clock Throttle STPCLK# Low Timer Value:** Bits[7:0] define the value x, where the Clock Throttle STPCLK# Low Timer count down value is (x + 1). (x + 1) defines the duration of the STPCLK# asserted period during clock throttling.

3.3.28 CTLTMRH—CLOCK THROTTLE STPCLK# HIGH TIMER

Address Offset: AEh
Default Value: 00h
Attribute: Read/Write
Size: 8 Bits

The duration of the STPCLK# negated period when bit 2 in the SMICNTL Register is set to 1 is (x + 1) where x equals the value programmed in this register. The value in this register plus 1 is loaded into the STPCLK# Timer when STPCLK# is negated. The STPCLK# timer counts using a 32 μs clock.

Bit	Description
7:0	**Clock Throttle STPCLK# High Timer Value:** Bits[7:0] define the value x, where the Clock Throttle STPCLK# High Timer count down value is (x + 1). (x + 1) defines the duration of the STPCLK# negated period during clock throttling.

3.4 ISA-Compatible Registers

This section describes the ISA-Compatible registers consisting of the DMA, interrupt controller, timer/counter, X-Bus control, NMI control, clock/reset, and advanced power management registers. Some of the registers are only accessible from the CPU/PCI Buses while others can be accessed by the CPU, PCI, or ISA Buses (see Table 7).

3.4.1 DMA REGISTER DESCRIPTION

The IB contains DMA circuitry that incorporates the functionality of two 82C37 DMA controllers. The two DMA controllers consist of two logical channel groupschannels [3:0] (Controller 1DMA1) and channels [7:4] (Controller 2DMA2).

This section describes the DMA registers. Unless otherwise stated, a CPURST sets each register to its default value. In addition, the DMA Master Clear Command (address 00Dh for channels [3:0] and 0DAh for channels [7:4]) permits software to set the DMA Command, DMA Status, DMA Request, and internal First/Last Flip-Flop Registers to their default values. The DMA Master Clear Command also sets the mask registers to their default values.

3.4.1.1 DCOM—DMA Command Register

I/O Address: Channels [3:0]—08h
Channels [7:4]—0D0h
Default Value: 00h
Attribute: Write Only
Size: 8 bits

This 8-bit register enables/disables the DMA channel groups, selects the priority scheme for responding to DMA requests, and selects the DMA request signal (DREQ) sense level. Following a CPURST or DMA Master Clear, both DMA1 and DMA2 are enabled in fixed priority and the DREQ sense level is active high.

Bit	Description
7	**DACK# Active Level (DACK[3:0,(7:5)]#):** Bit 7 controls the DMA channel request acknowledge (DACK#) assertion level. When bit 1 = 1, DACK# is an active high signal. When bit 1 = 0 (default), DACK# is an active low signal.
6	**DREQ Sense Assert Level (DREQ[3:0, (7:5)]):** Bit 6 controls the DREQx signal assertion level that the DMA controller detects as an active DMA channel request. Note that the DREQ channel assertion sensitivity is assigned by channel group, not per individual channel. When bit 6 = 0 (default), the DREQx sense assert level is active high. When bit 6 = 1, the DREQx sense assert level is active low. Following CPURST, the DREQx sense assert level is active high.
5	**Reserved:** Must be 0 when programming this register.
4	**DMA Group Arbitration Priority:** For priority resolution, the DMA consists of two logical channel groups—channels [3:0] (Controller 1—DMA1) and channels [7:4] (Controller 2—DMA2). Each group can be assigned fixed or rotating priority. Thus, both groups can be assigned fixed priority, one group can be assigned fixed priority and the other rotating priority, or both groups can be assigned rotating priority. When bit 4 = 0 (default), fixed priority is assigned to the channel. For fixed priority, the priority ordering is 0 (highest priority), 1, 2, 3, 5, 6, and 7 (lowest priority). Channels [3:0] of DMA-1 assume the priority position of Channel 4 in DMA-2, thus taking priority over channels 5, 6, and 7. Following CPURST, each group is initialized in fixed priority. When bit 4 = 1, rotating priority is assigned to the channel group. For rotating priority, the priority chain rotates so that the last channel serviced is assigned the lowest priority in the channel group ([3:0] or [7:5]). Channels [3:0] rotate as a group of 4. They are always placed between Channel 5 and Channel 7 in the priority list. Channel [7:5] rotate as part of a group of 4. That is, channels [7:5] form the first three positions in the rotation, while channels [3:0] comprise the fourth position in the arbitration.
3	**Reserved:** Must be 0 when programming this register.
2	**DMA Channel Group Enable:** When bit 2 = 1, the DMA channel group is disabled. Note that disabling channel group [7:4] also disables channel group [3:0], which is cascaded through channel 4. When bit 2 = 0 (default), the DMA channel group is enabled. Following CPURST, both channel groups are enabled.
1:0	**Reserved:** Must be 0 when programming this register.

3.4.1.2 DCM—DMA Channel Mode Register

I/O Address: Channels [3:0]—0Bh
Channels [7:4]—0D6h
Default Value: Bits[7:2] = 0, Bits[1:0] = undefined
Attribute: Write Only
Size: 8 bits

The Channel Mode Register controls DMA transfer type, transfer mode, address increment/decrement, and autoinitialization. The DMA transfer mode for channel 4 defaults to cascade and cannot be programmed for any mode other than DMA transfer mode.

Bit	Description
7:6	**DMA Transfer Mode:** Bits[7:6] select the DMA transfer mode as shown below. **Bits[7:6] — DMA Transfer Mode** 00 — Demand (default) 01 — Single 10 — Block 11 — Cascade
5	**Address Increment/Decrement Select:** Bit 5 controls address increment/decrement during multi-byte DMA transfers. When bit 5 = 0 (default), address increment is selected. When bit 5 = 1, address decrement is selected.
4	**Autoinitialize Enable:** When bit 4 = 1, the DMA restores the base page, address, and word count information to their respective current registers following a terminal count (TC). When bit 4 = 0 (default), the autoinitialize feature is disabled and the DMA does not restore the above mentioned registers.
3:2	**DMA Transfer Type:** This field selects verify, write, or read data transfer types as shown below. Write transfers move data from an I/O device to memory. Read transfers move data from memory to an I/O device. Verify transfers are pseudo transfers; addresses are generated as in a normal read or write transfer. However, with Verify transfers, the ISA memory and I/O cycle lines are not driven. When the channel is programmed for cascade (bits[7:6] = 11), the transfer type bits are irrelevant. **Bits[3:2] — DMA Transfer Type** 00 — Verify (default) 01 — Write 10 — Read 11 — Illegal: do not write
1:0	**DMA Channel Select:** This field select the DMA Channel Mode Register that will be written by bits[7:2] as shown below. These bits are undefined after a hard reset. **Bits[1:0] — DMA Channel Select** 00 — Channel 0(4) 01 — Channel 1(5) 10 — Channel 2(6) 11 — Channel 3(7)

3.4.1.3 DREQ—DMA Request Register

I/O Address: Channels [3:0]—09h
Channels [7:4]—0D2h
Default Value: Bits[1:0] = undefined, Bits[7:2] = 0
Attribute: Write Only
Size: 8 bits

The DMA Request Register is used by software to initiate a DMA request. The DMA responds to the software request as though DREQx is asserted. These requests are non-maskable and subject to prioritization by the priority encoder. When a TC is generated, the channel's request bit is set to 0. For software DMA requests, the channel must be in Block Mode. Note that the DMA Request Register status for DMA1 and DMA2 can be obtained from bits[7:4] of the DMA Status Register. The request bit for each channel is set to its default value by a CPURST or a Master Clear. The register is not affected by the RSTDRV output.

Bit	Description
7:3	**Reserved:** Must Be 0 when programming this register.
2	**DMA Channel Service Request:** When bit 2 = 1, a software DMA transfer is requested for the channel specified by bits[1:0]. When bit 2 = 0 (default), software DMA transfers are not requested for the channel specified by bits[1:0].
1:0	**DMA Channel Select:** This field selects the DMA channel to be written by bit 2 as shown below. **Bits[1:0] — DMA Channel Select** 00 — Channel 0(4) 01 — Channel 1(5) 10 — Channel 2(6) 11 — Channel 3(7)

3.4.1.4 WSMB—Write Single Mask Bit Register

I/O Address: Channels [3:0]—0Ah
Channels [7:4]—0D4h
Default Value: Bits[1:0] = undefined, Bit 2 = 1, Bits[7:3] = 0
Attribute: Write Only
Size: 8 bits

The WSMB Register permits the masking of the incoming DMA requests (DREQx) for each channel. A channel's mask bit is automatically set when the Current Byte/Word Count Register reaches terminal count, unless the channel is programmed for autoinitialization. This register is set to its default value by a CPURST or a Master Clear. Setting the entire register disables all DMA requests until a Clear Mask Register instruction allows them to occur. This instruction format is similar to the format used with the DMA Request Register.

NOTE:
Individually masking DMA channel 4 (DMA controller 2, channel 0) automatically masks DMA channels [3:0], as this channel group is logically cascaded onto channel 4. Setting this mask bit disables the incoming DREQ's for channels [3:0].

Bit	Description
7:3	**Reserved:** Must be 0 when programming this register.
2	**Channel Mask Select:** When bit 2 = 1 (default), DREQ is masked (disabled) for the channel selected by bits[1:0]. When bit 2 = 0, DREQ is not masked (enabled) for the channel selected by bits[1:0].
1:0	**DMA Channel Select:** This field selects the DMA channel to be written by bit 2 as shown below. **Bits[1:0]** — **DMA Channel Select** 00 — Channel 0(4) 01 — Channel 1(5) 10 — Channel 2(6) 11 — Channel 3(7)

3.4.1.5 WAMB—Write All Mask Bits Register

I/O Address: Channels [3:0]—0Fh
Channels [7:4]—0DEh
Default Value: Bit[3:0] = 1, Bit[7:4] = 0
Attribute: Read/Write
Size: 8 bits

This register enables/disables the incoming DREQx signals. All four channels can be simultaneously enabled/disabled instead of enabling/disabling each channel individually, as is the case with the Write Single Mask Bit Register.

Unlike the WSMB Register, the WAMB Register includes a status read to check the current mask status of the selected DMA channel group. A channel's mask bit is automatically set to 1 when the Current Byte/Word Count Register reaches terminal count, unless the channel is programmed for autoinitialization. All mask bits are set to 1 (disable) by CPURST or a Master Clear. Setting these bits to 1 disables all DMA requests until a Clear Mask Register instruction enables the requests.

NOTES:

1. Individually masking DMA channel 4 (DMA controller 2, channel 0) automatically masks DMA channels [3:0], as this channel group is logically cascaded onto channel 4.
2. Masking DMA controller 2 with a write to address 0DEh also masks DREQ assertions from the DMA controller, as this channel group is logically cascaded onto channel 4. When DMA channel 4 is masked, so are DMA channels [3:0].

Bit	Description
7:4	**Reserved:** Must be 0 when programming this register.
3:0	**Channel Mask Bits:** Setting the bit(s) to a 1 (default) disables the corresponding DREQ(s). Setting the bit(s) to a 0 enables the corresponding DREQ(s). When read, bits[3:0] indicate the DMA channel [3:0] ([7:4]) mask status.

3.4.1.6 DS—DMA Status Register

I/O Address: Channels [3:0]—08h
Channels [7:4]—0D0h
Default Value: 00h
Attribute: Read Only
Size: 8 bits

This register indicates which channels have reached terminal count and which channels have a pending DMA request.

Bit	Description
7:4	**Channel Request Status:** When a valid DMA request is pending for a channel (on its DREQ signal line), the corresponding bit is set to 1. When a DMA request is not pending for a particular channel, the corresponding bit is set to 0. The source of the DREQ may be hardware or a software request. Note that channel 4 does not have DREQ or DACK lines, so the response for a read of DMA2 status for channel 4 is irrelevant.
3:0	**Channel Terminal Count Status:** When a channel reaches terminal count (TC), its status bit is set to 1. If TC has not been reached, the status bit is set to 0. Note that channel 4 is programmed for cascade, and is not used for a DMA transfer. Therefore, the TC bit response for a status read on DMA2 for channel 4 is irrelevant. Bits[3:0] are set to 0 upon CPURST and on a read of the DS Register.

3.4.1.7 DB&CA—DMA Base And Current Address Registers (8237 Compatible Segment)

I/O Address: DMA Channel 0—000h, DMA Channel 1—002h, DMA Channel 2—004h,
DMA Channel 3—006h, DMA Channel 4—0C0h, DMA Channel 5—0C4h,
DMA Channel 6—0C8h, DMA Channel 7—0CCh
Default Value: Undefined
Attribute: Read/Write
Size: 16 bits per channel

Each channel has a 16-bit Current Address Register. This register contains the value of the 16 least significant bits of the full 27-bit address used during DMA transfers. The address is automatically incremented or decremented after each transfer and the intermediate values of the address are stored in the Current Address Register during the transfer. The Host CPU reads/writes the register in successive 8-bit bytes. This register is not accessible by ISA Bus masters. The programmer must issue the Clear Byte Pointer Flip-Flop Command to reset the internal byte pointer and correctly align the write prior to programming the Current Address Register. After clearing the Byte Pointer Flip-Flop, the first read/write accesses the low byte (bits[7:0]), and the second read/write accesses the high byte (bits[15:8]). Note that a mixed sequence of read and write cycles continues to toggle the Byte Pointer Flip-Flop, and successive reads and writes from this register alternate between the low byte and the high byte. An autoinitialize re-initializes the Current Address Register back to its original value following a TC. Autoinitialize occurs only after a TC.

Each channel has a Base Address Register located at the same address as the corresponding Current Address Register. These registers store the original value of their associated Current Address Registers. During autoinitialize these values are used to restore the Current Address Registers to the original values. The Base Registers are written simultaneously with their corresponding Current Address Register in successive 8-bit bytes. The Base Registers are write only.

Bit	Description
15:0	**Base and Current Address[15:0]:** These bits represent the 16 least significant address bits used during DMA transfers. Together with the DMA Low Page Register, they form the ISA-Compatible 24-bit DMA address. As an extension of the ISA-Compatible functionality, the DMA High Page Register completes the 27-bit DMA address generation, supporting DMA transfers throughout the full 128 MBytes of main memory. Upon CPURST or Master Clear, the value of these bits are undefined.

3.4.1.8 DB&CBW—DMA Base And Current Byte/Word Count Registers (8237 Compatible Segment)

I/O Address: DMA Channel 0—001h, DMA Channel 1—003h, DMA Channel 2—005h, DMA Channel 3—007h, DMA Channel 4—0C2h, DMA Channel 5—0C6h, DMA Channel 6—0CAh, DMA Channel 7—0CEh
Default Value: Undefined
Attribute: Read/Write
Size: 16 bits per channel

Each channel has a 16-bit Current Byte/Word Count Register that determines the number of transfers to be performed. The actual number of transfers is one more than the number programmed in the Current Byte/Word Count Register (i.e., programming a count of 100 results in 101 transfers). The byte/word count is decremented after each transfer. The intermediate value of the byte/word count is stored in the register during the transfer. When the value in the register goes from 0000h to FFFFh, a TC is generated.

Following the end of a DMA service, the register may also be re-initialized by an autoinitialization back to its original value. Autoinitialize can only occur when a TC occurs. If it is not autoinitialized, this register has a count of FFFFh after TC.

For transfers to/from an 8-bit I/O, the byte/word count indicates the number of bytes to be transferred. This applies to DMA channels [3:0]. For transfers to/from a 16-bit I/O, with shifted address, the byte/word count indicates the number of 16-bit words to be transferred. This applies to DMA channels [7:5].

Each channel has a Base Byte/Word Count Register located at the same I/O address as the corresponding Current Byte/Word Count Register. These registers store the original value of their associated Current Byte/Word Count Registers. During autoinitialize, these values are used to restore the Current Registers to their original values. The Base Registers are written simultaneously with their corresponding Current Register in successive 8-bit bytes. The Base Registers cannot be read by external agents.

Bit	Description
15:0	**Base and Current Byte/ Word Count:** These bits represent the 16 byte/word count bits used when counting down a DMA transfer. Upon CPURST or Master Clear, the value of these bits are undefined.

3.4.1.9 DMLPG—DMA Memory Low Page Registers

I/O Address: DMA Channel 0—087h, DMA Channel 1—083h, DMA Channel 2—081h, DMA Channel 3—082h, DMA Channel 5—08Bh, DMA Channel 6—089h, DMA Channel 7—08Ah
Default Value: Undefined
Attribute Read/Write
Size: 8 bits per channel

Each channel has an 8-bit Low Page Register. The DMA memory Low Page Register contains bits[23:16] of the 27-bit address. The register works in conjunction with the DMA controller's High Page Register and Current Address Register to define the complete address (27 bits) for the DMA channel. This register is static throughout the DMA transfer. Following an autoinitialization, this register retains the original programmed value. Autoinitialize takes place only after a TC.

Bit	Description
7:0	**DMA Low Page Address Bits[23:16]:** These bits represent the eight second most significant address bits when forming the 27-bit address for a DMA transfer. Upon CPURST or Master Clear, the value of these bits are undefined.

3.4.1.10 DMHPG—DMA Memory High Page Register

I/O Address: DMA Channel 0—487h, DMA Channel 1—483h, DMA Channel 2—481h, DMA Channel 3—482h, DMA Channel 5—48Bh, DMA Channel 6—489h, DMA Channel 7—48Ah
Default Value: Undefined
Attribute Read/Write
Size: 8 bits per channel

Each channel has an 8-bit High Page Register. The DMA Memory High Page Register contains the three most significant bits of the 27-bit address. The register works in conjunction with the Current Address Register and Low Page Register to define the complete 27-bit address for the DMA channel. This register is static throughout the DMA transfer. Following an autoinitialization, this register retains the original programmed value. Autoinitialize occurs only after a TC.

Bit	Description
7:3	**Reserved:** Must be 0 when programming this register.
2:0	**DMA High Page [26:24]:** These bits represent the three most significant address bits when forming the 27-bit address for a DMA transfer. Following the programming of a channel's Current Address Register or Low Page Register, this register is initialized to 00h. Upon CPURST or Master Clear, the value of these bits are undefined.

3.4.1.11 DCLBP—DMA Clear Byte Pointer Register

I/O Address: Channels [3:0]—00Ch
Channels [7:4]—0D8h
Default Value: Undefined
Attribute: Write Only
Size: 8 bits

Writing to this register executes the Clear Byte Pointer Command. This command is executed prior to writing/reading new address or word count information to/from the DMA. This command initializes the byte pointer flip-flop to a known state so that subsequent byte accesses to the 16-bit register contents address upper and lower bytes in the correct sequence.

The clear byte pointer command clears the internal flip-flop used to address the upper or lower byte of the 16-bit Address and Word Count Registers. The latch is also cleared at power-on by CPURST and by the Master Clear Command. The Host CPU may read or write a 16-bit DMA controller register by performing two consecutive accesses to the I/O port. The Clear Byte Pointer Command precedes the first access. The first I/O write to the register address loads the least significant byte, and the second access automatically accesses the most significant byte.

Bit	Description
7:0	**Clear Byte Pointer:** No specific pattern. The command is invoked with a write to the I/O address.

3.4.1.12 DMCL—DMA Master Clear Register

I/O Address: Channel [3:0]—00Dh
Channel [7:4]—0DAh
Default Value: Undefined
Attribute: Write Only
Size: 8 bit

This software command has the same effect as the hardware reset. The Command, Status, Request, and Internal First/Last Flip-Flop registers are cleared and the Mask Register is set. The DMA controller enters the idle cycle.

Bit	Description
7:0	**Master Clear Command:** No specific pattern. This command is invoked with a write to the I/O address.

3.4.1.13 DCLM—DMA Clear Mask Register

I/O Address: Channel [3:0]—00Eh
Channel [7:4]—0DCh
Default Value: Undefined
Attribute: Write Only
Size: 8 bit

This command clears the mask bits of all four channels, enabling them to accept DMA requests. I/O port 0Eh is used for channels 0–3 and I/O port 0DCh is used for channels 4–7.

Bit	Description
7:0	**Clear Mask Register Command:** No specific pattern. This command is invoked with a write to the I/O port address.

3.4.2 TIMER/COUNTER REGISTER DESCRIPTION

There are three counters that are equivalent to those found in the 82C54 Programmable Interval Timer. The counters are controlled by timer/counter registers that can be accessed from either the CPU, PCI Bus, or ISA Bus.

3.4.2.1 TCW—Timer Control Word Register

I/O Address: 043h
Default Value: Undefined
Attribute: Write Only
Size: 8 bits

The Timer Control Word Register specifies the counter selection, the operating mode, the counter byte programming order and size of the count value, and whether the counter counts down in a 16-bit binary or binary-coded decimal (BCD) format. After writing the control word, a new count can be written at any time. The new value takes effect according to the programmed mode.

There are six programmable counting modes. Typically, Timer Counters 0 and 2 are programmed for Mode 3, the Square Wave Mode, while Counter 1 is programmed in Mode 2, the Rate Generator Mode.

Two latch commands are selected through the Timer Control Word Register. The Read Back Command is selected when bits[7:6] = 11 and the Counter Latch Command is selected when bits[5:4] = 00. When either of these two commands are selected, the meaning of the other bits in the register changes.

Following CPURST, the control words for each register are undefined and each timer must be programmed for the counters to be in a known state. Note however, that some counter/timer functions are set to known states following CPURST. Each counter OUT signal is set to 0 (and the Timer Counter 2 OUT status bit in the NMISC Register is 0). The SPKR output, interrupt controller input IRQ0 (internal), and the internally generated refresh request are each set to 0 following CPURST.

Bit	Description
7:6	**Counter Select:** The Counter Selection bits select the counter the control word acts upon or the Read Back Command as shown below. **Bits[7:6] — Counter Select** 00 — Counter 0 01 — Counter 1 10 — Counter 2 11 — Read Back Command
5:4	**Read/Write Select:** This field selects the count register read/write programming mode or the Counter Latch Command as shown below. The read/write programming selection chosen indicates the programming sequence that must follow when initializing the counter specified in bits[7:6]. If a counter is programmed to read/write two byte counts, note that a program must not transfer control between writing the first and second byte to another routine that also writes into that same counter. Otherwise, the counter will be loaded with an incorrect count. The count must be completely loaded with both bytes. Note that the actual counter programming occurs by accessing I/O addresses 040h, 041h, and 042h for counters 0, 1, and 2, respectively. **Bits[5:4] — Read/Write Programming Select** 00 — Counter Latch Command 01 — R/W Least Significant Byte (LSB) 10 — R/W Most Significant Byte (MSB) 11 — R/W LSB, Then MSB
3:1	**Counter Mode Selection:** This field selects one of six possible modes of operation for the counter as shown below. **Bits[3:1] — Counter Mode** 000 — Out Signal On End of Count 001 — Hardware Re-triggerable one-shot X10 — Rate Generator (divide by n counter) X11 — Square Wave Output 100 — Software Triggered Strobe 101 — Hardware Triggered Strobe
0	**Binary/BCD Countdown Select:** When bit 0 = 0, a binary countdown is used. The largest possible binary count is 2^{16}. When bit 0 = 1, a binary-coded decimal (BCD) count is used. The largest BCD count allowed is 10^4.

1

Read Back Command

The Read Back Command provides the count value, programmed mode, and current states of the OUT pin and Null count flag of the selected counter or counters. The Read Back Command is written to the Timer Control Word Register that latches the current states of the above mentioned variables. The value of the counter and its status may then be read by I/O access to the counter address.

Status and/or count may be latched on one, two, or all three of the counters by selecting the counter during the register write. The count latched remains latched until read, regardless of further latch commands. The count must be read before newer latch commands latch a new count. The status latched by the Read Back Command also remains latched until after a read of the Counter Access Ports Register. Thus, the status and count are unlatched only after a counter read of the Timer Status Byte Format Register, the Counter Access Ports Register, or the Timer Status Byte Register and Counter Access Ports Register in succession.

Both count and status of the selected counter(s) may be latched simultaneously by setting both bit 5 and bit 4 to 0. This is functionally the same as issuing two consecutive, separate Read Back Commands. As mentioned above, if multiple count and/or status Read Back Commands are issued to the same counter(s) without any intervening reads, all but the first are ignored.

If both count and status of a counter are latched, the first read operation from that counter returns the latched status, regardless of which was latched first. The next one or two reads (depending on whether the counter is programmed for one or two byte counts) returns the latched count. Subsequent reads return an unlatched count.

Bit	Description
7:6	**Read Back Command:** When bits[7:6] = 11 during a write to the Timer Control Word Register, the Read Back Command is selected. As noted above, the normal meanings (mode, countdown, R/W select) of the bits in the control register at I/O address 043h change when the Read Back Command is selected. Following the Read Back Command, I/O reads from the selected counter's I/O addresses produce the current latch status, the current latched count, or both if bits[5:4] = 00.
5	**Latch Count of Selected Counters:** When bit 5 = 1, the count is not latched. When bit 5 = 0, the current count value of the selected counters is latched.
4	**Latch Status of Selected Counters:** When bit 4 = 1, the status is not latched. When bit 4 = 0, the status of the selected counters is latched. The status byte format is described in Section 3.4.2.2, Interval Timer Status Byte Format Register.
3	**Counter 2 Select:** When bit 3 = 1, Counter 2 is selected for the latch command selected with bits 4 and 5. When bit 3 = 0, status and/or count is not latched.
2	**Counter 1 Select:** When bit 2 = 1, Counter 1 is selected for the latch command selected with bits 4 and 5. When bit 2 = 0, status and/or count is not latched.
1	**Counter 0 Select:** When bit 1 = 1, Counter 0 is selected for the latch command selected with bits 4 and 5. When bit 1 = 0, status and/or count is not latched.
0	**Reserved:** Must be 0 when programming this register.

Counter Latch Command

The Counter Latch Command latches the current count value at the time the command is issued. This command is used to insure that the count read from the counter is accurate (particularly when reading a two-byte count). The count value is then read from each counter's count register (via the Counter Access Ports Register). One, two, or all three counters may be latched with one Counter Latch Command.

If a Counter is latched once and then, some time later, latched again before the count is read, the second Counter Latch Command is ignored. The count read is the count at the time the first Counter Latch Command was issued.

The count must be read according to the programmed format. Specifically, if the counter is programmed for two byte counts, two bytes must be read. The two bytes do not have to be read one right after the other (read, write, or programming operations for other counters may be inserted between the reads).

NOTES:

1. If a counter is programmed to read/write two byte counts, a program must not transfer control between reading the first and second byte to another routine that also reads from that same counter. Otherwise, an incorrect count will be read. Finish reading the latched two-byte count before transferring control to another routine.
2. The Timer Counter Register bit definitions are different during the Counter Latch Command than for a normal Timer Counter Register write.

Bit	Description
7:6	**Counter Selection:** This field selects the counter for latching by the Counter Latch Command as shown below. **Bits[7:6] — Counter Select** 00 — Counter 0 01 — Counter 1 10 — Counter 2 11 — Not Used; Do Not Write
5:4	**Counter Latch Command:** When bits[5:4] = 00 during a write to the Timer Control Word Register, the Counter Latch Command is selected. As noted above, the normal meanings (mode, countdown, R/W select) of the bits in the control register at I/O address 043h change when the Counter Latch Command is selected. Following the Counter Latch Command, I/O reads from the selected counter's I/O addresses produce the current latched count.
3:0	**Reserved:** Must be 0 when programming this register.

3.4.2.2 TMSTAT—Interval Timer Status Byte Format Register

I/O Address: Counter 0—040h
Counter 1—041h
Counter 2—042h
Default Value: Bits[6:0] = undefined, Bit 7 = 0
Attribute: Read Only
Size: 8 bits per counter

Each counter's status byte can be read following an Interval Timer Read Back Command. The Read Back Command is programmed through the Timer Control Word Register. If latch status is chosen (bit 4 = 0, Read Back Command) as a read back option for a given counter, the next read from the counter's Counter Access Ports Register returns the status byte. The status byte returns the countdown type (either BCD or binary), the counter operational mode, the read/write selection status, the Null count (also referred to as the count register status), and the current state of the counter OUT pin.

Bit	Description
7	**Counter OUT Pin State:** When bit 7 = 1, the OUT pin of the counter is 1. When bit 7 = 0 (default) , the OUT pin of the counter is 0.
6	**Count Register Status:** Null Count (also referred to as the Count Status Register) indicates when the last count written to the Count Register (CR) has been loaded into the counting element (CE). The exact time this happens depends on the counter mode. However, until the count is loaded into the counting element (CE), it can't be read from the counter. If the count is latched or read before the load time, the count value returned does not reflect the new count written to the register. When bit 6 = 0, the count has been transferred from CR to CE and is available for reading. When bit 6 = 1, the Null count condition exists. The count has not been transferred from CR to CE and is not yet available for reading.
5:4	**Read/Write Selection Status:** Bits[5:4] reflect the read/write selection made through bits[5:4] of the Timer Control Word Register. The binary codes returned during the status read match the codes used to program the counter read/write selection. **Bits[5:4] — Read/Write Programming Select Status** 00 — Counter Latch Command 01 — R/W Least Significant Byte (LSB) 10 — R/W Most Significant Byte (MSB) 11 — R/W LSB, Then MSB
3:1	**Mode Selection Status:** Bits[3:1] return the counter mode programming made through bits[3:1] of the Timer Control Word Register. The binary code returned matches the code used to program the counter mode, as listed under the bit function above. **Bits[3:1] — Counter Mode** 000 — Out Signal On End of Count 001 — Hardware Re-triggerable one-shot X10 — Rate Generator (divide by n counter) X11 — Square Wave Output 100 — Software Triggered Strobe 101 — Hardware Triggered Strobe
0	**Countdown Type Status:** Bit 0 reflects the current countdown type—either 0 for binary countdown or a 1 for binary coded decimal (BCD) countdown.

3.4.2.3 CAPS—Counter Access Ports Register

I/O Address: Counter 0, System Timer—040h
Counter 1, Refresh Request—041h
Counter 2, Speaker Tone—042h
Default Value: Undefined
Attribute: Read/Write
Size: 8 bits per counter

These I/O addresses provide access for a) writing count values to the Count Registers, b) reading the current count value from the counter by either an I/O read, after a counter-latch command, or after a Read Back Command, and c) reading the status byte following a Read Back Command.

Bit	Description
7:0	**Counter Port Byte:** Each counter I/O address is used to program the 16-bit Count Register. The order of programming, either LSB only, MSB only, or LSB then MSB, is defined by the Timer Control Word Register. This register I/O address also reads the current count from the Count Register and returns the status of the counter programming following a Read Back Command.

3.4.3 INTERRUPT CONTROLLER REGISTER DESCRIPTION

The 82420EX PCIset contains an ISA-Compatible interrupt controller that incorporates the functionality of two 82C59 interrupt controllers. The interrupt registers control the operation of the interrupt controller and can be accessed from the CPU or PCI Bus. In addition, some of the registers can be accessed from the ISA Bus.

3.4.3.1 ICW1—Initialization Command Word 1 Register

I/O Address: INT CNTRL-1—020h
INT CNTRL-2—0A0h
Default Value: Undefined
Attribute: Write Only
Size: 8 bits per controller

A write to Initialization Command Word 1 starts the interrupt controller initialization sequence. Addresses 020h and 0A0h are referred to as the base addresses of CNTRL-1 and CNTRL-2, respectively. An I/O write to the CNTRL-1 or CNTRL-2 base address with bit 4 equal to 1 is interpreted as ICW1. For the 82420EX PCIset-based ISA systems, three I/O writes to "base address + 1" must follow the ICW1. The first write to "base address + 1" performs ICW2, the second write performs ICW3, and the third write performs ICW4. ICW1 starts the initialization sequence during which the following automatically occur:

1. The Interrupt Mask Register is cleared.
2. IRQ7 input is assigned priority 7.
3. The slave mode address is set to 7.
4. Special Mask Mode is cleared and Status Read is set to IRR.
5. If IC4 was set to 0, then all functions selected by ICW4 are set to 0. However, in the IB implementation, ICW4 must be programmed and IC4 must be set to a 1.

ICW1 has three significant functions in the IB interrupt controller configuration. ICW4 is needed, so bit 0 must be programmed to a 1. Since there are two interrupt controllers in the system, bit 1 (SNGL) must be programmed to a 0 on both CNTRL-1 and CNTRL-2, to indicate a cascade configuration. The IB provides separate registers (ELCR Registers) to program level or edge sensitive interrupt IRQ lines. Thus, bit 3 (LTIM in the 82C59) is not used. Bit 4 must be a 1 when programming ICW1. This bit indicates that ICW1, and not OCW2 or OCW3, will be programmed during the write to this port.

Bit	Description
7:5	**ICW/OCW Select:** Bits[7:5] are MCS-85 implementation specific bits. These bits are not used and should be 000 when programming the interrupt controller.
4	**ICW/OCW Select:** Bit 4 = 1 selects ICW1. OCW2 and OCW3 are also addressed at the same port as ICW1. A "1" on this bit at any time will force the interrupt controller to interpret the write as an ICW1. The controller will then expect to see ICW2, ICW3, and ICW4. After the fixed initialization sequence to ICW1, ICW2, ICW3, and ICW4, the controller base address is used to write to OCW2 and OCW3. Bit 4 is a "0" on writes to these registers.
3	**Edge/Level Bank Select (LTIM):** This bit is disabled. Its 82C59 Interrupt Controller function is replaced by the Edge/Level Triggered Control Registers (ELCR). The ELCR registers allow the interrupts to be programmed for edge or level mode on an interrupt by interrupt basis.
2	**ADI:** Bit 2 (ADI) is a MCS-85 implementation specific bit. This bit is not used and should be 0 when programming the IB.
1	**Single/Cascade Select (SNGL):** SNGL must be programmed to a 0 to indicate that two interrupt controllers are operating in cascade mode on the IB.
0	**ICW4 Write Required (ICW4):** This bit must be set to a 1. IC4 indicates that ICW4 needs to be programmed. The IB requires that ICW4 be programmed to indicate that the controllers are operating in an 80x86 type system.

3.4.3.2 ICW2—Initialization Command Word 2 Register

I/O Address: INT CNTRL-1—021h
INT CNTRL-2—0A1h
Default Value: Undefined
Attribute: Write Only
Size: 8 bits per controller

ICW2 is used to initialize the interrupt controller with the five most significant bits of the interrupt vector address. The value programmed for bits[7:3] is used by the Host CPU to define the base address in the interrupt vector table for the interrupt routines associated with each IRQ on the controller. Typical ISA ICW2 values are 08h for CNTRL-1 and 70h for CNTRL-2.

Bit	Description
7:3	**Interrupt Vector Base Address:** Bits[7:3] define the base address in the interrupt vector table for the interrupt routines associated with each interrupt request level input. For CNTRL-1, a typical value is 00001b, and for CNTRL-2, 01110b. The interrupt controller combines a binary code representing the interrupt level to receive service with this base address to form the interrupt vector that is driven onto the bus. For example, the complete interrupt vector for IRQ0 (CNTRL-1), would be 0000 1000b (CNTRL-1 [7:3] = 00001b and 000b representing IRQ0). This vector is used by the CPU to point to the address information that defines the start of the interrupt routine.
2:0	**Interrupt Request Level:** When writing ICW2, this field should be 000. During an interrupt acknowledge cycle, this field is programmed by the interrupt controller with the interrupt code representing the interrupt level to be serviced. This interrupt code is combined with bits[7:3] to form the complete interrupt vector driven onto the data bus during the second INTA# cycle. The 3-bit binary codes are: 000 represents IRQ0 (IRQ8), 001 IRQ1 (IRQ9), 010 IRQ2 (IRQ10), 011 IRQ3 (IRQ11), 100 IRQ4 (IRQ12), 101 IRQ5(IRQ13), 110 IRQ6(IRQ14), and 111 IRQ7 (IRQ15).

3.4.3.3 ICW3—Initialization Command Word 3 Register

I/O Address: INT CNTRL-1—021h
Default Value: Undefined
Attribute: Write Only
Size: 8 bits

The meaning of ICW3 differs between CNTRL-1 and CNTRL-2. On CNTRL-1, the master controller, ICW3 indicates which CNTRL-1 IRQ line physically connects the INTR output of CNTRL-2 to CNTRL-1. ICW3 must be programmed to 04h, indicating the cascade of the CNTRL-2 INTR output to the IRQ2 input of CNTRL-1.

An interrupt request on IRQ2 causes CNTRL-1 to enable CNTRL-2 to present the interrupt vector address during the second interrupt acknowledge cycle.

Bit	Description
7:3	**Not Used:** Must be 0.
2	**Cascaded Interrupt Controller IRQ Connection:** Bit 2 must always be programmed to a 1. This bit indicates that CNTRL-2, the slave controller, is cascaded on interrupt request line two (IRQ2). When an interrupt request is asserted to CNTRL-2, the IRQ goes through the priority resolver. After the slave controller priority resolution is finished, the INTR output of CNTRL-2 is asserted. However, this INTR assertion does not go directly to the CPU. Instead, the INTR assertion cascades into IRQ2 on CNTRL-1. IRQ2 must go through the priority resolution process on CNTRL-1. If it wins the priority resolution on CNTRL-1 and the CNTRL-1 INTR signal is asserted to the CPU, the returning interrupt acknowledge cycle is really destined for CNTRL-2. The interrupt was originally requested at CNTRL-2, so the interrupt acknowledge is destined for CNTRL-2, and not a response for IRQ2 on CNTRL-1. When an interrupt request from IRQ2 wins the priority arbitration, in reality an interrupt from CNTRL-2 has won the arbitration. Because bit 2 of ICW3 on the master is set to 1, the master knows which identification code to broadcast on the internal cascade lines, alerting the slave controller that it is responsible for driving the interrupt vector during the second INTA# pulse.
1:0	**Not Used:** Must be 0.

3.4.3.4 ICW3—Initialization Command Word 3 Register

I/O Address: INT CNTRL-2—0A1h
Default Value: Undefined
Attribute: Write Only
Size: 8 bits

On CNTRL-2 (the slave controller), ICW3 is the slave identification code broadcast by CNTRL-1 from the trailing edge of the first INTA# pulse to the trailing edge of the second INTA# pulse. CNTRL-2 compares the value programmed in ICW3 with the incoming identification code. The code is broadcast over three internal cascade lines. ICW3 must be programmed to 02h for CNTRL-2. When 010b is broadcast by CNTRL-1 during the INTA# sequence, CNTRL-2 assumes responsibility for broadcasting the interrupt vector during the second interrupt acknowledge cycle.

As an illustration, consider an interrupt request on IRQ2 of CNTRL-1. By definition, a request on IRQ2 must have been asserted by CNTRL-2. If IRQ2 wins the priority resolution on CNTRL-1, the interrupt acknowledge cycle returned by the CPU following the interrupt is destined for CNTRL-2, not CNTRL-1. CNTRL-1 will see the INTA# signal, and knowing that the actual destination is CNTRL-2, will broadcast a slave identification code across the internal cascade lines. CNTRL-2 compares this incoming value with the 010b stored in ICW3. Following a positive decode of the incoming message from CNTRL-1, CNTRL-2 drives the appropriate interrupt vector onto the data bus during the second interrupt acknowledge cycle.

Bit	Description
7:3	**Reserved:** Must be 0 when programming this register.
2:0	**Slave Identification Code:** The Slave Identification code must be programmed to 010b during the initialization sequence. The code stored in ICW3 is compared to the incoming slave identification code broadcast by the master controller during interrupt acknowledge cycles.

3.4.3.5 ICW4—Initialization Command Word 4 Register

I/O Address: INT CNTRL-1—021h
INT CNTRL-2—0A1h
Default Value: 01h
Attribute: Write Only
Size: 8 bits

Both interrupt controllers must have ICW4 programmed as part of their initialization sequence. Minimally, the microprocessor mode bit (bit 0) must be set to a 1 to indicate an Intel architecture-based platform. Failure to program this bit will result in improper controller operation during interrupt acknowledge cycles. Additionally, the Automatic End of Interrupt (AEOI) may be selected, as well as the Special Fully Nested Mode (SFNM) of operation.

Bit	Description
7:5	**Reserved:** Must be zero when programming this register.
4	**Special Fully Nested Mode (SFNM):** When bit 4 = 1, the special fully nested mode is enabled. When bit 4 = 0, the special fully nested mode is disabled (this is the normal mode).
3	**Buffered Mode (BUFM):** Must be 0 (non-buffered mode).
2	**Master/Slave in Buffered Mode:** Must be 0.
1	**Automatic End of Interrupt (AEO):** When bit 1 = 1, the automatic end of interrupt mode is selected. When bit 1 = 0, the normal end of interrupt is selected. This bit should normally be programmed to 0.
0	**Microprocessor Mode:** The Microprocessor Mode bit must be programmed to 1 to indicate that the interrupt controller is operating in an Intel Architecture-based system. Never program this bit to 0.

3.4.3.6 OCW1—Operational Control Word 1 Register

I/O Address: INT CNTRL-1—021h
INT CNTRL-2—0A1h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

OCW1 sets and clears the mask bits in the Interrupt Mask Register (IMR). Each interrupt request line may be selectively masked or unmasked any time after initialization. Note that masking IRQ2 on CNTRL-1 also masks all of controller 2's interrupt requests (IRQ[15:8]). Reading OCW1 returns the controller's mask status.

The IMR stores the bits that mask the interrupt lines to be masked. The IMR operates on the IRR. Masking of a higher priority input does not affect the interrupt request lines of lower priority. Unlike status reads of the ISR and IRR, for reading the IMR, no OCW3 is needed. The output data bus contains the IMR when a read occurs to I/O address 021h or 0A1h (OCW1). All writes to OCW1 must occur following the ICW1-ICW4 initialization sequence, since the same I/O addresses are used for OCW1, ICW2, ICW3 and ICW4.

Bit	Description
7:0	**Interrupt Request Mask (Mask [7:0]):** When a 1 is written to any bit in this register, the corresponding IRQx line is masked. For example, if bit 4 = 1, IRQ4 is masked. Interrupt requests on IRQ4 will not set channel 4's interrupt request register (IRR) bit as long as the channel is masked. When a bit = 0 (default), the corresponding IRQx mask bit is cleared, and interrupt requests are accepted by the controller. Note that masking IRQ2 on CNTRL-1 also masks the interrupt requests from CNTRL-2, which is physically cascaded to IRQ2.

Bit	Interrupt	Bit	Interrupt
0	IRQ0	4	IRQ4
1	IRQ1	5	IRQ5
2	IRQ2	6	IRQ6
3	IRQ3	7	IRQ7

3.4.3.7 OCW2—Operational Control Word 2 Register

I/O Address: INT CNTRL-1—020h
INT CNTRL-2—0A0h
Default Value: Bit[4:0] = undefined, Bit[7:5] = 001
Attribute: Write Only
Size: 8 bits

OCW2 controls both the Rotate Mode and the End of Interrupt Mode, and combinations of the two. OCW2 also selects individual interrupt channels during three of the seven commands. The three low order bits (labeled L2, L1 and L0) are used when bit 6 is set to a 1 during the command. Following a CPURST or ICW initialization, the controller enters the fully nested mode of operation. Non-specific EOI without rotation is the default. Both rotation mode and specific EOI mode are disabled following initialization.

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>7:5</td><td>Rotate and EOI Codes: These three bits control the Rotate and End of Interrupt modes and combinations of the two as shown below. Bit 7 = R(rotate), Bit 6 = SL, and Bit 5 = EOI.

<table>
<tr><th>Bits[7:5]</th><th>Rotate and EOI Modes</th><th>Bit[7:5]</th><th>Rotate and EOI Modes</th></tr>
<tr><td>001</td><td>Non-Specific EOI Command</td><td>000</td><td>Rotate in Auto EOI Mode (clear)</td></tr>
<tr><td>011</td><td>Specific EOI Command</td><td>111</td><td>Rotate on Specific EOI Command*</td></tr>
<tr><td>101</td><td>Rotate on Non-Specific EOI Command</td><td>110</td><td>Set Priority Command*</td></tr>
<tr><td>100</td><td>Rotate in Auto EOI Mode (set)</td><td>010</td><td>No Operation</td></tr>
</table>
NOTE:
*Interrupt Select Levels are used.</td></tr>
<tr><td>4:3</td><td>OCW2 Select: When selecting OCW2, bits 3 and 4 must both be 0. If bit 4 is a 1, the interrupt controller interprets the write to this port as an ICW1. Therefore, always ensure that these bits are both 0 when writing an OCW2.</td></tr>
<tr><td>2:0</td><td>Interrupt Level Select (L2, L1, L0): L2, L1, and L0 determine the interrupt level acted on when the SL bit is active. A simple binary code, outlined below, selects the channel for the command to act on. When the SL bit is inactive, these bits do not have a defined function. In this case, programming L2, L1 and L0 to 0 is sufficient.

<table>
<tr><th>Bits[2:0]</th><th>Interrupt Level</th><th>Bits[2:0]</th><th>Interrupt Level</th></tr>
<tr><td>000</td><td>IRQ0(8)</td><td>100</td><td>IRQ4(12)</td></tr>
<tr><td>001</td><td>IRQ1(9)</td><td>101</td><td>IRQ5(13)</td></tr>
<tr><td>010</td><td>IRQ2(10)</td><td>110</td><td>IRQ6(14)</td></tr>
<tr><td>011</td><td>IRQ3(11)</td><td>111</td><td>IRQ7(15)</td></tr>
</table></td></tr>
</table>

3.4.3.8 OCW3—Operational Control Word 3 Register

I/O Address: INT CNTRL-1—020h
INT CNTRL-2—0A0h
Default Value: Bit[6,0] = 0, Bit[7,4:2] = undefined, Bit[5,1] = 1
Attribute: Read/Write
Size: 8 bits

OCW3 serves three important functions—Enable Special Mask Mode, Poll Mode control, and IRR/ISR register read control. First, OCW3 is used to set or reset the Special Mask Mode (SMM). The Special Mask Mode can be used by an interrupt service routine to dynamically alter the system priority structure while the routine is executing, through selective enabling/disabling of the other channel's mask bits.

Second, the Poll Mode is enabled when a write to OCW3 is issued with bit 2 equal to 1. The next I/O read to the interrupt controller is treated like an interrupt acknowledge—a binary code representing the highest priority level interrupt request is released onto the bus.

Third, OCW3 provides control for reading the In-Service Register (ISR) and the Interrupt Request Register (IRR). Either the ISR or IRR is selected for reading with a write to OCW3. Bits 0 and 1 carry the encoded command to select either register. The next I/O read to the OCW3 port address returns the register status specified during the previous write. The register specified for a status read is retained by the interrupt controller. Therefore, a write to OCW3 prior to every status read command is unnecessary, provided the status read desired is from the register selected with the last OCW3 write.

Bit	Description
7	**Reserved:** Must be zero when programming this register.
6	**Special Mask Mode (SMM):** If SMME = 1 and SMM = 1 the interrupt controller enters Special Mask Mode. If SMME = 1 and SMM = 0 (default), the interrupt controller is in normal mask mode. When SMME = 0, SMM has no effect.
5	**Special Mask Mode Enable (SMME):** When ESMM = 1 (default), the SMM bit is enabled to set or reset the Special Mask Mode. When ESMM = 0, the SMM bit becomes a "don't care".
4:3	**OCW3 Select:** When selecting OCW3, bit 3 must be a 1 and bit 4 must be 0. If bit 4 = 1, the interrupt controller interprets the write to this port as an ICW1. Therefore, always ensure that bits[4:3] = 01 when writing an OCW3.
2	**Poll Mode Command:** When bit 2 = 0, the Poll command is not issued. When bit 2 = 1, the next I/O read to the interrupt controller is treated as an interrupt acknowledge cycle. An encoded byte is driven onto the data bus, representing the highest priority level requesting service.
1:0	**Register Read Command:** Bits[1:0] provide control for reading the In-Service Register (ISR) and the Interrupt Request Register (IRR). When bit 1 = 0, bit 0 does not affect the register read selection. When bit 1 = 1, bit 0 selects the register status returned following an OCW3 read. If bit 0 = 0, the IRR is read. If bit 0 = 1, the ISR is read. Following ICW initialization, the default OCW3 I/O address read is "read IRR". To retain the current selection (read ISR or read IRR), always write a 0 to bit 1 when programming this register. The selected register can be read repeatedly without reprogramming OCW3. To select a new status register, OCW3 must be reprogrammed prior to attempting the read.

3.4.3.9 ELCR1—Edge/Level Triggered Register

I/O Address: INT CNTRL-1—4D0h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

The ELCR1 Register selects either edge triggered or level sensitive operations for the IRQ[7:3] signals. In edge triggered mode, the interrupt is recognized by a low to high transition. In level sensitive mode, the interrupt is recognized by a low level. Note that IRQ0, IRQ1 and IRQ2 are not programmable and are always edge sensitive. The default for IRQ[7:3] is edge triggered.

Bit	Description
7:3	**Bit 7 - Bit 3: IRQ[7:3] ECL:** Bit 7 to bit 3 select edge trigger or level sensitive modes for IRQ[7:3], respectively. When a bit is 1, the corresponding IRQx is in the level sensitive mode and when a bit is 0, the corresponding IRQx is in the edge trigger mode.
2:0	**Reserved:** Must be 0 when programming this register.

3.4.3.10 ELCR2—Edge/Level Triggered Register

I/O Address: INT CNTRL-2—4D1h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

The ELCR2 Register selects either edge triggered or level sensitive operations for the IRQ[15,14,12:9] signals. In edge triggered mode, the interrupt is recognized by a low to high transition. In level sensitive mode, the interrupt is recognized by a low level. Note that IRQ13 and IRQ8 are not programmable and are always edge sensitive. The default for IRQ[15,14,12:9] is edge triggered.

Bit	Description
7:6	**Bit 7 - Bit 6: IRQ[15,14] ECL:** Bit 7 and bit 6 select edge trigger or level sensitive modes for IRQ[15,14], respectively. When a bit is 1, the corresponding IRQx is in the level sensitive mode and when a bit is 0, the corresponding IRQx is in the edge trigger mode.
5	**Reserved:** Must be 0 when programming this register.
4:1	**Bit 4 - Bit 1: IRQ[12:9] ECL:** Bit 4 to bit 1 select edge trigger or level sensitive modes for IRQ[12:9], respectively. When a bit is 1, the corresponding IRQx is in the level sensitive mode and when a bit is 0, the corresponding IRQx is in the edge trigger mode.
0	**Reserved:** Must be zero when programming this register.

3.4.4 X-BUS REGISTER DESCRIPTION

There are two X-Bus registers described in this section—the Reset X-Bus IRQ[12,1] Register and the Coprocessor Error Register. These registers can be accessed from the CPU, PCI Bus, or ISA Bus.

3.4.4.1 RIRQ—Reset IRQ[12,1]

I/O Address: 60h
Default Value: NA
Attribute: Read only
Size: 8 bits

Address locations 60h and aliased address 62h are used to clear the mouse interrupt (IRQ12) and keyboard interrupt (IRQ1). When the mouse interrupt function is enabled (bit 4 in the X-Bus Chip Select Register is 1), the mouse interrupt function is provided on the IRQ12/M input signal. In this mode, a mouse interrupt generates an interrupt through IRQ12 to the Host CPU. A read of 60h releases IRQ12. If bit 4 = 0 in the X-Bus Chip Select Register, a read of 60h or 62h has no effect on IRQ12/M. Note, however, that a read of these addresses always clears the keyboard interrupt (IRQ1). Reads and writes to this register flow through to the ISA Bus.

Bit	Description
7:2	**Reset IRQ[12,1]:** No specific pattern. A read of address 60h executes the command.
1:0	**Reset IRQ[12,1]:** No specific pattern. A read of address 60h executes the command.

3.4.4.2 CPERR—Coprocessor Error Register

I/O Address: F0h
Default Value: NA
Attribute: Write only
Size: 8 bits

Writes to this address are monitored by the IB. Writing to address F0h causes the IB to drive IGNNE# low to the CPU (informing the CPU to ignore future coprocessor errors). The IB also negates IRQ13 (internal to the IB). Note, that IGNNE# is not asserted unless FERR# is active. Reads and writes to this register flow through to the ISA Bus.

Bit	Description
7:0	**Ignore Coprocessor Error Command:** No special pattern required: A write to address F0h executes the command.

3.4.5 NMI REGISTER DESCRIPTION

An NMI is an interrupt requiring immediate attention and has priority over the normal interrupt lines (IRQx). The IB indicates error conditions by generating a non-maskable interrupt. NMI interrupts (special cycles) are caused by the following conditions:

1. System errors on the PCI Bus. SERR# is driven low by a PCI resource when this error occurs.
2. Parity errors on the add-in memory boards on the ISA expansion bus. IOCHK# is driven low when this error occurs.
3. Main memory parity errors, through the SERR# signal.

There are two 8-bit registers that support NMI—The NMI Status and Control (NMISC) Register and the NMI Enable and Real Time Clock Address (NMIERTC) Register. These registers can be accessed from the CPU, PCI Bus, or ISA Bus. Note that masking the NMI signal for all sources via the NMIERCT Register does not affect the input NMI status conditions (i.e., bits 6 and 7 in the NMISC Register). This means that, if NMI is masked and then unmasked, an NMI will occur if an NMI had previously been detected. To ensure that all NMI requests are serviced, the NMI service routine software flow should be as follows:

1. NMI is detected by the CPU on the rising edge of the NMI input.
2. The CPU reads NMI status via the NMISC Register to determine the NMI source. Software may then set the status bits that caused the NMI to 0. Between the time the CPU reads the NMI sources and sets them to a 0, an NMI may have been generated by another source. In this case, the NMI signal remains asserted. If this happens, the new NMI source will not be recognized by the because there was no edge on NMI.
3. Software must then disable the NMI signals in the NMIERTC Register. This causes the NMI output to transition low then high if there are any pending NMI sources.

3.4.5.1 NMISC—NMI Status and Control Register

I/O Address: 061h
Default Value: 00U0 0000
Attribute: Read/Write
Size: 8 bits

This register provides status of various system components, speaker counter (Counter 2) output control, and gates the counter output that drives the SPKR signal.

Bit		Description
7	RO	**SERR# NMI Source Status:** System agents on the PCI Bus (PCI devices or main memory) assert the SERR# signal to report system errors. When the SERR# signal is asserted (and bit 2 of this register is 0), this bit is set to a 1. In addition, if bit 7 of the NMIERTC Register is 1, an NMI is generated. Software can clear this bit and the interrupt by setting bit 2 to 0 and then setting bit 2 to 1. This bit is read only. When writing to this register, bit 7 must be 0.
6	RO	**IOCHK# NMI Source Status:** Expansion boards on the ISA Bus assert IOCHK# to request high priority servicing (e.g., parity errors on memory cards). When the IOCHK# signal is asserted (and bit 3 of this register is 0), this bit is set to a 1. In addition, if bit 7 of the NMIERTC Register is 1, an NMI is generated. Software can clear this bit and the interrupt by setting bit 3 to 0 and then setting bit 3 to 1. This bit is read only. When writing to this register, bit 6 must be 0.
5	RO	**Timer Counter 2 OUT Status:** This bit reflects the current state of the Interval Timer Counter 2 OUT signal. Counter 2 must be programmed following a CPURST for this bit to have a determinate value. This bit is read only. When writing to this register, bit 5 must be 0.
4	RO	**Refresh Cycle Toggle:** The Refresh Cycle Toggle signal toggles from either 0 to 1 or 1 to 0 following every refresh cycle. This bit is read only. When writing to this register, bit 4 must be 0.
3	R/W	**IOCHK# NMI Enable:** When bit 3 = 1, IOCHK# NMI's are disabled and cleared, and bit 6 of this register is disabled (always reads 0). When bit 3 = 0 (default), bit 6 is enabled and, if bit 7 of the NMIERTC Register is 1, the IOCHK# NMI is enabled.
2	R/W	**PCI SERR# Enable:** When bit 2 = 1, SERR# NMI's are disabled and cleared, and bit 7 of this register is disabled (always reads 0). When bit 2 = 0 (default), bit 7 is enabled and, if bit 7 of the NMIERTC Register is 1, the SERR# NMI is enabled.
1	R/W	**Speaker Data Enable:** This bit enables/disables the SPKR output signal. When bit 1 = 1, the SPKR output signal is enabled and equivalent to the Counter 2 OUT signal. When bit 1 = 0 (default), the SPKR output is disabled and always 0.
0	R/W	**Timer Counter 2 Enable:** When bit 0 = 1, Timer Counter 2 counting is enabled. When bit 0 = 0 (default), Counter 2 counting is disabled.

3.4.5.2 NMIERTC—NMI Enable and Real Time Clock Address Register

I/O Address: 070h
Default Value: Bit[6:0] = undefined, Bit 7 = 1
Attribute: Write Only
Size: 8 bits

This register enables/disables all NMIs and provides a real time clock address pointer field to address memory locations. Do not modify the contents of this register without considering the effects on the state of the other bits. Reads and writes to this register address flow through to the ISA Bus.

Bit	Description
7	**NMI Enable (NMIE):** This bit provides a mask of the NMI output signal. When bit 7 = 1 (default), NMI is disabled for all sources and the NMI signal is negated. When bit 7 = 0, NMIs are enabled. Setting this bit to 1 does not clear or disable the NMI status conditions. Thus, if NMI is disabled then enabled via this register, an NMI will occur if one of the NMI status bits (6 or 7) is set in the NMISC Register.
6:0	**Real time Clock Address:** Used by the Real Time Clock on the Base I/O component to address memory locations.

3.4.6 POWER MANAGEMENT REGISTER DESCRIPTION

This section describes two power management registers—APMS and APMC Registers. These registers are located in normal I/O space and must be accessed (via the CPU or PCI Bus) with 8 bit accesses. Note that the rest of the power management registers are located in PCI configuration space (see Section 3.3, PCI Configuration Registers).

3.4.6.1 APMC—Advanced Power Management Control Port

I/O Address: 0B2h
Default Value: 00h
Attribute: Read/Write
Size: 8 Bits

This register passes data (APM Commands) between the OS and the SMI handler. In addition, writes can generate an SMI and reads can cause STPCLK# to be asserted. The IB operation is not affected by the data in this register.

Bit	Description
7:0	**APM Control Port (APMC):** Writes to this register store data in the APMC Register and reads return the last data written. In addition, writes generate an SMI, if bit 7 of the SMIEN Register and bit 0 of the SMICNTL Register are both is set to 1. Reads cause the STPCLK# signal to be asserted, if bit 1 of the SMICNTL Register is set to 1. Reads do not generate an SMI.

3.4.6.2 APMS—Advanced Power Management Status Port

I/O Address: 0B3h
Default Value: 00h
Attribute: Read/Write
Size: 8 Bits

This register passes status information between the OS and the SMI handler. The IB operation is not affected by the data in this register.

Bit	Description
7:0	**APM Status Port (APMS):** Writes store data in this register and reads return the last data written.

4.0 FUNCTIONAL DESCRIPTION

This section describes the 82420EX PCIset functions and hardware interfaces including the PCIset memory and I/O address map, system arbitration, DMA, interrupt controller, Timer/counter, power management, and clock/reset. The Clock/Reset section also covers the strapping options for the different hardware configuration modes and provides tables listing the state of each 82420EX PCIset output or bi-directional signal during a hard reset. The Host Bus, PCI Bus, ISA Bus, X-Bus, and the PSC/IB Link interfaces are described. The L2 cache and main memory DRAM interfaces and associated memory arrays are covered.

4.1 Memory and I/O Address Map

The 82420EX PCIset interfaces to three system Buses—CPU, PCI, and ISA Buses. The 82425EX PSC provides positive decode for certain I/O and memory space accesses on the CPU and PCI Buses. These decodes include accesses to the PCI Local Bus IDE (CPU only), main memory (CPU, ISA, and PCI), ISA-Compatible registers (CPU, ISA, and PCI), and the PSC's I/O Control Registers (CPU only). In addition, the PSC subtractively decodes certain CPU/PCI cycles.

The 82426EX IB provides positive decode for certain ISA I/O and memory space accesses. These decodes include accesses to the ISA-Compatible registers (for ISA master and DMA initiated cycles), main memory (for ISA and DMA initiated cycles), BIOS, X-Bus, and system events for SMM support. Note that DMA devices and ISA masters can not access the PCI or CPU Buses.

4.1.1 MEMORY ADDRESS MAP

The PSC positively decodes accesses to main memory space (128 MBytes maximum as programmed in the DRB Registers). Accesses in the following ranges are forwarded to the PCI Bus: TOM to 128 MBytes, (2 GByte minus 128 MByte) to (2 GByte plus 128 MByte), and 4 GByte to (4 GByte minus 128 MByte).

All other memory spaces are not intended for use.

NOTE:
The PSC does not decode host CPU address signals A[29:27].

All CPU/PCI accesses to the 0–640 KByte main memory region are forwarded to main memory (Table 10). Accesses to the video region (640–768 KBytes) are subtractively decoded to the ISA Bus. Accesses to the 768 KByte to 1 MByte region are controlled by attribute bits in the PAM Registers. Accesses to the region between 1 MByte and 128 MByte are either sent to main memory or subtractively decoded to either PCI or ISA.

Table 10. CPU/PCI 0 to 128 MByte Address Map

Memory Segment	Decode
16–128 MBytes	Positive decode(1,2)
Top of Hole to 16 MBytes	Positive decode(1,2)
Bottom of Hole to the Top of Hole	Subtractive decode
1 MByte to the Bottom of Hole	Positive decode(1,2)
768 KByte to 1 MByte	Positive decode or subtractively decoded to ISA(3)
640–768 KByte (Video)	Subtractively decoded to ISA
0–640 KByte	Positive decode(4)

NOTES:
1. As programmed in the DRAM Row Boundary Register.
2. For memory accesses that are > top of DRAM <16M, the PSC subtractively decodes and forwards to ISA.
3. The 82420EX allows 13 programmable memory and cacheability attributes on 13 memory segments of various sizes in the ISA compatiblity hole (768 KBytes to 1 MByte). Refer to the PAM Register description.
4. Always in DRAM.

All ISA master and DMA accesses to memory locations 0–640 KByte are forwarded to main memory. ISA memory accesses from 1 MByte to the top of memory are forwarded to main memory, except for accesses to the programmable memory hole. ISA accesses from 768 KByte to 1 MByte (except for E0000–EFFFFh, if forwarding is enabled) and accesses above the top of main memory are confined to the ISA Bus.

Table 11. DMA and ISA Master Accesses to Main Memory

Memory Space	Response
Top of main memory to 128 MByte	Confine to ISA
1 MByte to top of main memory	Forward to main memory(1)
768 KBytes to 1 MByte	Confine to ISA(2)
640–768 KByte (video)	Confine to ISA
0–640 KByte	Forward to main memory

NOTES:
1. Except accesses to programmed memory hole.
2. If bit 6 is 0 in the XBCSA Register and bit 6 is 1 in the PIRQ0 Register, accesses to E0000–EFFFFh are forwarded to main memory.

The 82420EX supports one hole in main memory, as defined by the MEMHOLE Register. CPU accesses in the memory hole are forwarded to the PCI Bus and, if not claimed, forwarded to the ISA Bus. PCI master accesses in the memory hole are subtractively decoded to ISA, if necessary. ISA master accesses are confined to the ISA Bus.

4.1.2 BIOS MEMORY SPACE

The 82420EX PCIset supports 512 KBytes of BIOS space. This includes the normal 128 KByte space plus an additional 384 KByte BIOS space (known as the enlarged BIOS area). All BIOS regions that are not shadowed in main memory are subtractively decoded.

The 128 KByte BIOS memory space is located at 000E0000–000FFFFFh (top of 1 MByte), and is aliased at FFFE0000–FFFFFFFFh (top of 4 GByte). This 128 KByte block is split into two 64 KByte blocks. CPU/PCI accesses to the top 64 KByte region (000F0000–000FFFFFh) that are not claimed by main memory or PCI, are forwarded to ISA. The subsequent ISA cycle always generates a BIOS chip select (asserts BIOSCS#).

CPU/PCI accesses to the bottom 64 KByte region (000E0000–000EFFFFh) that are not claimed by main memory or PCI are forwarded to ISA. The subsequent ISA cycle generates a BIOS chip select, if lower BIOS is enabled (via the XBCSA Register).

The additional 384 KByte region resides at FFF80000–FFFDFFFFh. When enabled (via the XBCSA Register), CPU/PCI memory accesses to this region are subtractively decoded to the ISA Bus and BIOS chip select is generated.

All ISA BIOS accesses within the F0000–FFFFFh region are confined to the ISA Bus, even if BIOS is shadowed in main memory. Accesses to the E0000–EFFFFh region are confined to the ISA Bus, when this BIOS region is enabled (via the XBCSA Register). When the region is disabled, accesses are forwarded to main memory, if forwarding is enabled (via the PREV Register). Note that bit 6 in the XBCSA Register overrides bit 4 in the PREV Register.

4.1.3 VIDEO FRAME BUFFER

The Video Frame Buffer can be mapped in the following ranges:

1. In the standard VGA range.
2. In a defined memory hole. In this case, DRAM size is limited to 128 MByte Memory minus the hole size. For example, if there is a 2 MByte Frame Buffer hole, somewhere between 1 MByte and 16 MByte, then the maximum DRAM size allowed is 128 MByte minus 2 MByte equal 126 MByte.
3. Above Top of Memory, but under 128 MByte. In this case, maximum DRAM size is limited to 128 MByte minus the Frame Buffer Size.
4. Above 128 MByte. There are three non-aliased ranges above 128 MByte, which can be used for Frame Buffer:

   (2 GByte minus 128 MByte) to 2 GByte — When HA[31:27] = 01111

   2 GByte to (2 GByte plus 128 MByte) — When HA[31:27] = 10000

   (4 GByte minus 128 MByte) to (4 GByte minus .5 MByte) — When HA[31:27] = 11111

4.1.4 I/O ACCESSES

The PSC positively decodes access to the I/O control registers, PCI configuration registers, and ISA-Compatible Registers. For details concerning accessing these registers, see Section 3.0, Register Description. In addition, the PSC positively decodes CPU I/O accesses to the IDE ports, when enabled. For IDE port accesses, see Section 4.5, PCI Local Bus IDE.

4.1.5 SMRAM: PROTECTED SMM MEMORY BLOCK

The 82420EX PCIset supports a dedicated 64 KBytes of SMM memory, called SMRAM. The SMRAM is accessible only when certain conditions are met. In normal operations, the SMRAM is hidden. SMRAM can be located at the A0000–F0000h segment. The SMRAM can be enabled/disabled and programmed via the SMRAM Control Register.

When SMRAM is hidden, the whole memory space can be accessed, excluding the SMRAM block. When the SMRAM is visible, most of the memory space is visible, in addition to the SMRAM block. Only the memory block that shares the same bus address ranges with SMRAM cannot be accessed in this case.

SMRAM is visible under the following conditions:

- The CPU is in SMM, performing a memory cycle in the SMRAM range, while the SMRAM is not manually closed. This is indicated by SMIACT# = 0, HLDA = 0, HA is in the SMRAM range (as programmed by SMBASE field of the SMRAMCON Register), and SMCLS = 0. The SMCLS bit only affects data cycles and is ignored for code read cycles.

- The CPU is not in SMM, performing a memory cycle in the SMRAM range, while the SMRAM is manually opened. This is indicated by SMIACT# = 1, HLDA = 0, HA is in the SMRAM range, and the SMOPN = 1.
- On each CPU access when SMIACT# is asserted (or SMOPN = 1), the main memory address is compared to the selected SMM memory address as determined by the SMBASE field. These bits allow the user to select from eight different 64 KByte main memory locations used for SMM memory.

4.2 PSC/IB Link Interface

The PSC and IB communicate using the PSC/IB interface. Interface communications include CPU/PCI accesses of the IB internal registers, CPU/PCI cycles forwarded to the ISA Bus, and ISA master or DMA accesses to main memory. The PSC/IB Link interface is a point-to-point communication connection between the PSC and the IB.

Four sideband signals synchronize data flow and bus ownership—Link Request (LREQ#), Link Grant (LGNT#), Command Valid (CMDV#), and Slave Idle (SIDLE#). LREQ# and LGNT# are used by the IB to arbitrate for link mastership. Only the IB drives LREQ# while only the PSC drives LGNT#. CMDV# is driven by the current link master, while SIDLE# is driven by the current link slave. Commands, addresses, and data are transferred between the PSC and IB using the host address bus signals (A[17:2]).

4.3 Host CPU Interface

The 82420EX PCIset provides a host interface to all of the Intel486 family of processors and upgrades.

4.3.1 HOST BUS SLAVE DEVICE

The PCIset can be configured (via the HOST Device Control Register) to support an Intel486 Host Bus Slave device (specifically, a graphics device). Two special signals (HDEV# and HRDY#) as defined by the VL Bus specification are used in the interface to the Host Bus slave. The PSC can be configured to monitor HDEV# for all memory and I/O ranges that are not positively decoded by the PSC. The PSC can be configured to monitor HRDY# and return RDY# to the CPU, based on HRDY#. The host device may include an I/O range, memory range or both I/O and memory ranges. In all cases, these ranges must not be programmed (positively decoded) by the PSC. The host device's memory ranges are non-cacheable.

NOTES:

1. DMA, ISA Masters and PCI masters cannot access the Host Bus device.
2. The PSC does not contain a time-out mechanism to recover a cycle when HDEV# is asserted but HRDY# is not asserted. When the Host Bus device asserts HDEV#, it is assumed that the HRDY# assertion will follow.
3. Host Bus Device—Read and Write fastest timing. During a CPU read (fastest timing programmed), the Host Bus device must not start driving the data bus until two Host Bus clocks after the active ADS# period. This is required so the L2 cache can drive the data bus for a 0 wait-state L2 read. During a CPU write, the Host Bus device can respond with HRDY# in the cycle following ADS#, to achieve a 0 wait-state write cycle.
4. If the Host bus slave device is implemented on the motherboard, the graphics controller and video DAC chip must be accessed through the host bus. The graphics ROM must also be accessed through the host bus or integrated into system BIOS. If the graphics ROM is not integrated into system BIOS, and the graphics ROM is shadowed into DRAM, the host device must not respond with HDEV# when the graphics ROM area is accessed. If the graphics ROM is not shadowed, the PSC PAM registers must be programmed not to respond to the graphics ROM area.
5. If the host bus slave device is implemented using a connector, the graphics controller and video DAC chip must be accessed through the host bus. The graphics ROM can be accessed through the ISA bus. However, when accessing the ROM BIOS, the card in the connector must latch the address with ADS#.
6. Not all host bus slave devices use all of the host address lines for decode. If this is the case, note that PCI memory and system memory is limited by the number of address lines used by the host bus device for decode (e.g. if [25:2] are only used by the host device, usable system memory is limited to 64 MByte, as the host device will alias anything above 64 MByte).

4.3.2 L1 CACHE SUPPORT

The 82420EX PCIset provides signals that support the CPU's L1 cache. For the S-Series CPUs, the signals are the PCD, KEN#, and EADS# signals. For the D-Series and P24T CPUs, the signals are the KEN#, EADS#, CACHE#, and HITM#. The P24T and the D-Series CPUs include certain signals that are not connected to the PCIset. These signals are fixed to 1 or 0, depending on the system configuration as discussed in Table 12.

4.3.3 SOFT AND HARD RESETS

The 82420EX PCIset generates soft reset (SRESET) and hard resets (PCIRST#, RSTDRV, and CPURST).

Soft Reset

SRESET/INIT is generated under the following conditions:

1. Programming the TRC Register (see TRC Register Description).
2. Keyboard KBDRST#: This signal from the keyboard controller is used to generate a soft reset to the CPU via the PSC's SRESET/INIT signal.
3. Shutdown special cycle: When the CPU executes a shutdown special cycle, SRESET is generated.

Hard Reset

The IB generates a hard reset under two conditions:

1. PWROK; When PWROK is driven low, the IB asserts PCIRST#, RSTDRV, and CPURST. These hard reset signals remain asserted until 2 HCLKIN cycles after the rising edge of PWROK.
2. Programming the TRC Register (see TRC Register description).

Reset Distribution

Figure 4 and Figure 5 show how the hard and soft resets are distributed in the system. The specific implementation depends on the CPU type as shown in the figures.

To ensure that SMI# is not generated during SRESET, an external "OR" gate must be used as shown in Figure 6.

Table 12. L1 Cache Signals Not Connected to the PSC

Signal	Description
INV	INV input is tied to 1 for P24T and D-Series processor configuration. When INV = 1, the cache line is invalidated as a result of snoop-hit cycle.
HIT	The CPU asserts HIT to indicate that an Inquire cycle hits a line. The PSC only needs to know when the L1 cache line needs to perform write-back, as a result of Inquire cycle. Since the need to perform write backs is indicated by HITM#, the HIT is not needed in the PSC.
FLUSH#	The PSC does not support L1 hardware flush. Since SMRAM must be configured in a non-cacheable region, there is no need for automatic FLUSH# in a 82420 PCIset-based system. Therefore, FLUSH# can be tied to 1. The PSC recognizes the FLUSH special cycles and responds with RDY# to allow external logic to activate FLUSH# (if desired).
BLEN#	BLEN# is tied to 0 for P24T and D-Series configurations to enable bursted write-back cycles.
WB/WT#	WB/WT# is tied to 1 for P24T and D-Series configurations to set all cache lines in write-back mode. Individual lines can be configured in write-through mode by software only.
EWBE#	P24T External Write Buffer Empty input is used to enforce correct cycle ordering in concurrent systems. In 82420 PCIset-based systems, there is only a single active master at a time. Thus, the PSC does not use the EWBE# signal.
PWT	PWT is used as an indication to cache a line in a write-through mode. The PSC L2 Cache update mode can not be set on a line by line basis and thus PWT is not used.

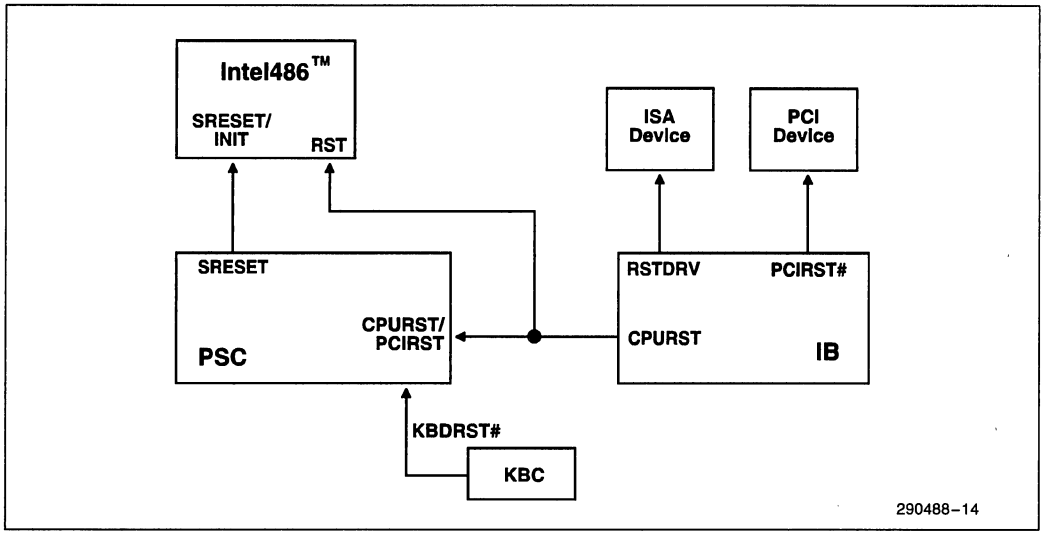


Figure 4. Reset Distribution for CPU's with Hard Reset and Soft Reset Inputs

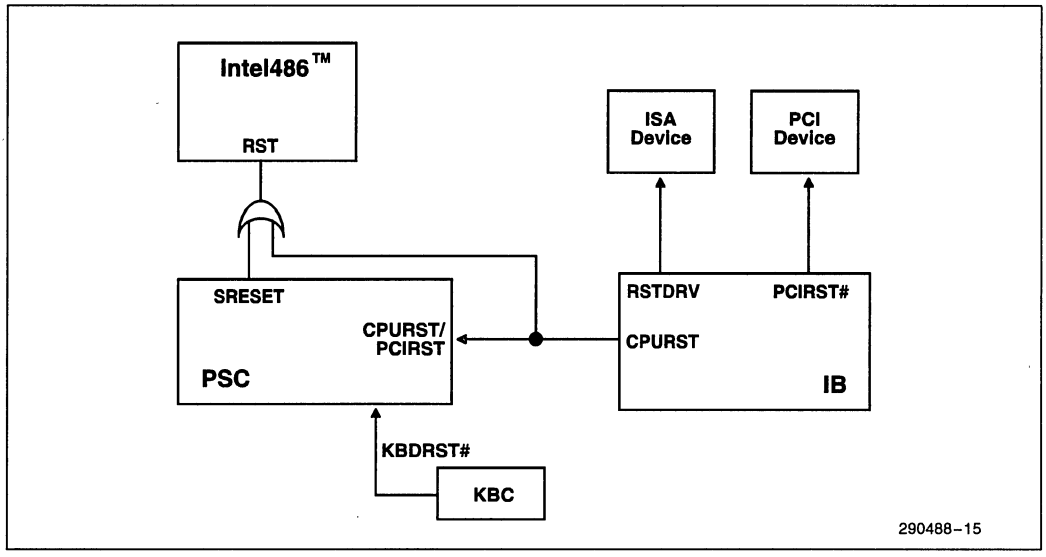


Figure 5. Reset Distribution for CPU's with One Reset Input

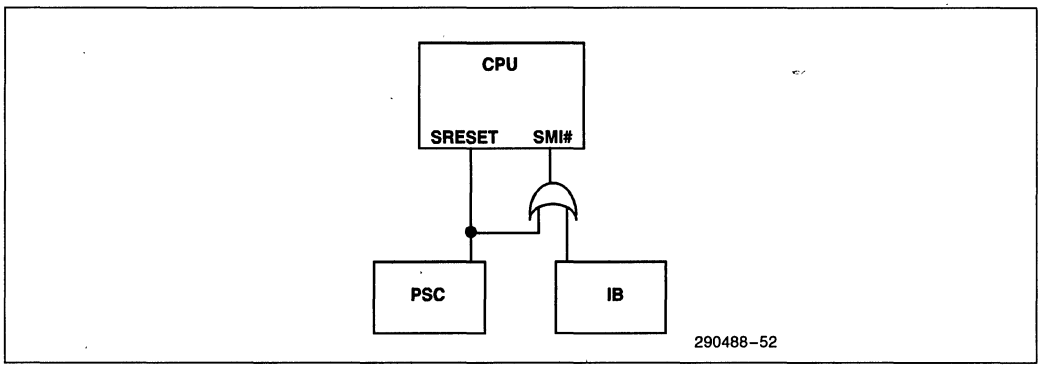


Figure 6. SMI# Gated with SRESET

4.3.4 KEYBOARD CONTROLLER (A20)

The 82420EX supports the generation of A20M# via the keyboard controller. To support the generation of A20M#, external logic may be required (Figure 7), depending on system requirements. "OR" gates 1 and 2 ensure that A20M# is negated during a SRESET or CPURST, respectively. "OR" gate 2 is not required if the KBC firmware forces KBCA20M# high during a hard reset. "OR" gate 3 and the inverter ensure that A20M# is negated when SMIACT# is asserted. "OR" gate 3 and the inverter are not required if the SMRAM is located under 1 MByte or at an even 1 MByte boundary and the SMRAM code does not need to go above the 1 MByte range while in SMM mode.

4.4 PCI Interface

The PSC has a standard master/slave PCI Bus interface. As a PCI device, the PSC can be either a master initiating a PCI Bus operation or a target responding to a PCI Bus operation. The PSC is a PCI Bus master for Host-to-PCI accesses and a target for PCI-to-Main memory accesses (or accesses that are forwarded to the ISA Bus). The Host can read or write configuration spaces, PCI memory space, and PCI I/O space.

NOTES:

1. PCI-to-Host accesses are not permitted. However, PCI-to-Main memory cycles that require the L1/L2 caches to be snooped, do invoke Host Bus cycles.
2. ISA-to-PCI accesses are not permitted.

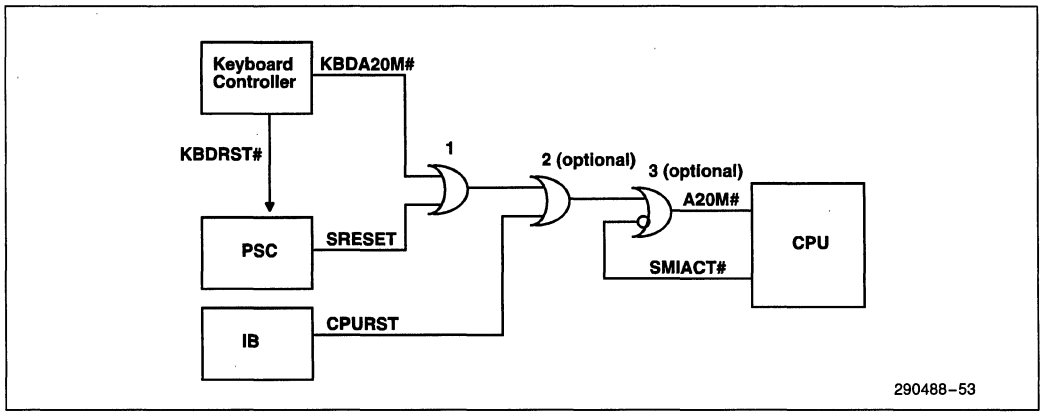


Figure 7. System Connection for Keyboard A20M# Generation

4.4.1 PCI BUS CYCLES SUPPORT

When the host initiates a bus cycle to a PCI device, the PSC becomes a PCI Bus master and translates the CPU cycle into the appropriate PCI Bus cycle. Post buffers permit the CPU to complete Host-to-PCI writes in zero wait-states.

When a PCI Bus master initiates a main memory access, the PSC becomes the target of the PCI Bus cycle and responds to the read/write access. As a PCI master, the PSC generates address parity for read and write cycles, and data parity for write cycles. As a target, the PSC generates data parity for read cycles. During PCI-to-Main memory accesses, the PSC automatically performs cache snoop operations on the Host Bus, if needed, to maintain data consistency.

PCI Bus commands indicate to the target the type of transaction desired by the master. These commands are presented on the C/BE[3:0]# signals during the address phase of a transfer. Table 13 summarizes The PSC's support of the PCI Bus commands.

PSC Supports Other PCI Bridges

The PCI Bus specification supports bridges that connect the system's local PCI Bus with other remote busses (PCI or others). The PSC supports the ability to connect bus bridges onto the local PCI Bus.

One type of PCI bridge interfaces the local PCI Bus to a set of slave (only) devices. In this case, the bridge performs protocol translation and may include write buffers (pointing away from the local PCI). An example of such a bridge is the PCI-to-PCMCIA bridge device (PPEC).

A second type of PCI bridge interfaces the local PCI Bus with another bus that supports masters and slaves—a remote PCI Bus. This type of bridge can generally include write buffers (and pre-fetchers) that are pointing in both directions (to local PCI Bus and away from local PCI Bus).

Table 13. Supported PCI Bus Commands

C/BE[3:0]#	Command Type	Supported As Target	Supported As Master
0000	Interrupt Acknowledge	No	No
0001	Special Cycle	No	No
0010	I/O Read	Yes	Yes
0011	I/O Write	Yes	Yes
0100	Reserved	—	—
0101	Reserved	—	—
0110	Memory Read	Yes	Yes
0111	Memory Write	Yes	Yes
1000	Reserved	—	—
1001	Reserved	—	—
1010	Configuration Read	No	Yes
1011	Configuration Write	No	Yes
1100	Memory Read Multiple	Yes(1)	No
1101	Dual Address Cycle	No	No
1110	Memory Read Line	Yes(1)	No
1111	Memory Write and Invalidate	Yes(2)	No

NOTES:
1. As a target, the PSC treats this command as a memory read command.
2. As a target, the PSC treats this command as a memory write command.

The PSC supports PCI-to-PCI bridges, with the following restrictions:

- The 82420EX PCIset does not allow more than a single active master in the entire system. This restriction prevents a remote PCI Bus master from performing an exclusive access that is claimed by the bridge (the target is on the local PCI Bus), while there is another active master in the system (that may be performing another exclusive access on the local PCI Bus).
- When a master is granted, it is guaranteed that the PSC's PCI write buffers are empty. Since the PSC does not know the status of other bridges's buffers (that point to the PCI) while it grants the CPU, the other bridge's buffers must be disabled.

4.4.2 HOST TO PCI CYCLES

Host bus accesses to PCI Bus are always in the Host Bus address range, as defined by A[31:30,26:2] and the four BE lines. The PCI address lines are driven during the address phase. AD[29:27] lines are driven to the value of A[30], during Host accesses to PCI.

The PSC has the ability to burst up to 32 back-to-back CPU memory writes on the PCI Bus. This function is controlled by the PCICON Register. The PSC is capable of merging 8/16-bit graphic write cycles to the same dword address into the same posted write buffer location (controlled by the PCICON Register). The merged data is then driven as a single dword cycle on the PCI Bus. Byte merging is performed in the compatible VGA range only.

4.4.3 PCI CYCLE TERMINATIONS

The PSC performs a master abort, received target abort, signaled target abort, and a disconnect (as either a master or slave) as described in this section.

Master Abort—PSC as a Master

The PSC performs two types of master abort when a PCI cycle is not claimed by PCI Bus devices.

Master-Abort of Type 1 is performed by the PSC for the following conditions:

- When the memory address is lower than 16 MBytes.
- When the memory address is higher than 16 MBytes, but it is an enabled BIOS range.
- When the I/O address is lower than 64 KBytes.

Type 1 master abort actions:

- Master abort is performed.
- The cycle is forwarded to the ISA Bus.

Master abort of Type 2 is performed by the PSC for the following conditions:

- When the memory address is higher than 16 Mbytes, and it is not an enabled BIOS range.
- I/O address is above 64 KBytes.
- When a configuration access to a PCI device is not claimed.

Type 2 master abort actions:

- Master abort is performed.
- Master abort status bit (DS Register) is set.
- For reads, data of all 1's is returned to the CPU.
- For writes, RDY# is activated to complete the CPU cycle.

Received Target Abort—PSC as a Master:

When a PSC driven cycle is target aborted, the PSC sets the Received Target Abort status bit to 1 (in the DS Register). In addition, when SERR# is enabled, this signal is asserted for a single PCICLK. RDY# is asserted to complete the CPU cycle.

NOTE:
When the CPU attempts an access configuration registers and the function number is not 000, data of all 1's is returned (if it is a read cycle) and Target Abort Status bit is set.

Disconnect—PSC as a Master

When the PSC, as a PCI master, generates a burst memory write, it can be disconnected by the PCI target. The PSC will retry the disconnected cycle before any arbitration changes can be performed, since the PSC write buffers must be emptied and the on-going CPU access must be completed before an arbitration transfer can take place.

Disconnect—PSC as a Target

The PSC, as a PCI target, performs a disconnect when burst PCI master accesses are destined to the ISA Bus. The disconnect is performed at the completion of the first data phase. In addition, for burst PCI master cycles to main memory, the PSC performs a disconnect at the completion of the last data phase in a line boundary.

4.4.4 EXCLUSIVE CYCLES

The PSC, as a PCI master, never performs LOCKed cycles. The CPU does not return active HLDA while it is performing a LOCKed sequence. Also, the CPU is the only active master, as long as HLDA is inactive. Thus, the PSC does not need to drive LOCK to guarantee the CPU atomic LOCK sequence. Note that the 82420EX PCIset supports a bus locking mechanism (i.e., when a PCI master performs locked accesses, the arbitration is not changed, until the locked sequence is completed).

4.4.5 Parity Support

As a master, the PSC generates address parity for read and write cycles, and data parity for write cycles. As a slave, the PSC generates data parity for read cycles. Even parity is generated using the PAR line in the PCICLK following the PCI address or data phase.

The PSC does not check parity or generate SERR#, based on the PCI parity. The PSC only generates SERR# (if enabled via the PCICOM Register), when a main memory read results in a parity error. When a main memory parity error is detected, the PSC activates SERR#, if enabled, for a single PCICLK.

When a main memory parity error is detected and SERR# generation is enabled, the MMPERR bit in the DS Register is set to 1. When SERR# is activated, the SERRS bit in the DS Register is set to 1.

4.5 PCI Local Bus Ide

The PSC has a full-function PCI Local Bus IDE Controller capable of generating high speed PCI Local Bus IDE cycles. The PCI IDE address, control, and data signals are multiplexed with the PCI AD signals (Figure 8). They are buffered by external TTL devices to drive the IDE connector. Only CPU accesses to IDE can use the PCI local bus path. PCI masters and ISA masters can not access the drive connected to the PCI Local Bus.

The PSC's IDE interface supports one IDE connector (two drives). An additional IDE connector could be connected to the ISA or PCI Bus. The PCI IDE interface can be programmed at either the primary address (1F0h–1F7h, 3F6h, 3F7h) or secondary address (170h–177h, 376h, 377h) locations.

The selected IDE's data port, as well as the control/status ports, are accessed through the PCI Local Bus path. The PSC provides data steering to route data between the IDE data bus and the correct Host Bus byte lane. However, the PSC does not support multiple assembly/disassembly cycles for data size mismatches. Data size matching is guaranteed by the IDE device driver. The PSC assumes that all data port accesses (1F0h, 170h) are 16 bits wide. If an 8-bit IDE drive is accessed, the PSC still drives 16 bits onto the PCI AD signals for data port writes to IDE, and drive HD(15:0) for data port reads. Accesses to the PCI Local Bus control and status ports are assumed to be 8-bit accesses and the PSC steers the data to the appropriate byte lane. Table 14 shows the I/O addresses for the various IDE data, control, and status ports:

The PSC controls the timing of the high speed accesses to the PCI IDE connector and provides programmable timing fields (via the LBIDE Register). This allows the PCI local bus IDE timing to be programmed to cover 25 MHz and 33 MHz PCI frequencies, and IDE Modes 1, 2, and 3. The programmable timing also allows additional flexibility in the event that still faster IDE modes are defined in the future. Note that, the faster timing applies to data port accesses only. Accesses to all other ports, except the data port, run with compatible timings.

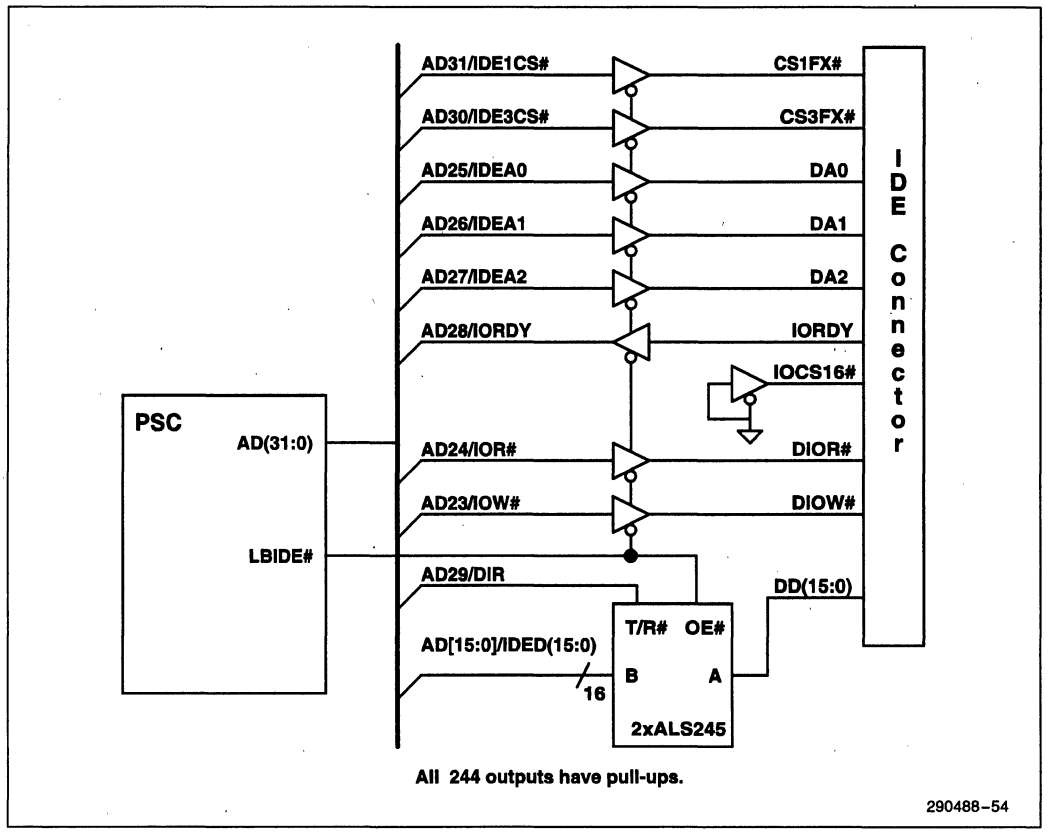


Figure 8. PSC PCI Local Bus IDE Connection

The PSC supports one connector that can be assigned at the primary or secondary address. This one connector can support two drives. The drive is selected through the Drive, Head port at I/O address 1x6h. The CPU writes bit 4 to a 0 to select drive 0, and writes bit 4 to a 1 to select drive 1. The PSC snoops writes to the enabled 1x6h (1F6h for primary, 176h for secondary) and keeps its own copy of bit 4 of I/O 1x6h. The fast timing bank can be programmed to apply to data port accesses to either drive 0, drive 1, or both. Accesses to the non-selected drive run with compatible timings.

Typically in a PC, when reading from port 3x7h, bits[6:0] are provided by the IDE drive and bit 7 is provided by the floppy disk controller as a reflection of the DSKCHG from the floppy disk drive. This occurs for both the primary and secondary locations. The PSC handles CPU I/O reads to port 3x7h in a unique fashion. For example, when the primary address range is enabled, the PSC splits the read to 3F7h and generates both a PSC/IB link interface bus cycle as well as a PCI Local Bus IDE cycle. The PSC takes bit 7 from the link cycle, merges it with bits[6:0] from the PCI local bus IDE cycle, and returns the complete 8 bits to the CPU. If the primary address range is not enabled, only the PSC/IB link interface bus cycle is generated. The same operation applies to 377h reads, when the secondary address range is enabled. This feature permits the PCI Local Bus IDE to be used in a system where, for example, the AIP is placed on an add-in card.

Table 14. IDE I/O Addresses

I/O Address	Port Function (R/W)
1x0h	Data
1x1h	Error/Features
1x2h	Sector Count
1x3h	Sector Number
1x4h	Cylinder Low
1x5h	Cylinder High
1x6h	Drive, Head
1x7h	Status/Command
3x6h	Alternate Status/Device Control
3x7h	Drive Address

NOTE:
x = F for Primary and 7 for Secondary

4.6 ISA Interface

The IB incorporates a fully ISA Bus compatible master and slave interface. The IB directly drives five ISA slots without external data or address buffers. The ISA interface also provides byte swap logic, I/O recovery support, wait-state generation, and SYSCLK generation. The ISA interface supports the following cycle types:

- CPU or PCI master initiated I/O and memory cycles to the ISA Bus.
- DMA compatible cycles between main memory and ISA I/O and between ISA I/O and ISA memory.
- ISA refresh cycles initiated by either the IB or an external ISA master.
- ISA master-initiated memory cycles to main memory and ISA master-initiated I/O cycles to the internal IB registers.

NOTES:

1. The IB does not grant the ISA Bus to an ISA master before gaining ownership of the system (i.e. Host and PCI Buses).
2. All cycles forwarded to main memory run as 16-bit extended cycles (i.e. IOCHRDY is negated until the cycle completes). Because the ISA Bus size is different from the main memory bus size, the data steering logic inside the IB steers the data to the correct byte lanes.

I/O Recovery Support

The I/O recovery mechanism in the IB is used to add additional recovery delay between the CPU or PCI master initiated 8-bit and 16-bit I/O cycles to the ISA Bus. The IB automatically forces a minimum delay of 3.5 SYSCLKs between back-to-back 8- and 16-bit I/O cycles to the ISA Bus. This delay is measured from the rising edge of the I/O command (IOR# or IOW#) to the falling edge of the next I/O command. If a delay of greater than 3.5 SYSCLKs is required, the ISA I/O Recovery Timer Register can be programmed to increase the delay in increments of SYSCLKs. No additional delay is inserted for back-to-back I/O sub-cycles generated as a result of byte assembly or disassembly.

SYSCLK Generation

The IB generates the ISA system clock (SYSCLK). SYSCLK is a divided down version of HCLKOUT and has a frequency of either 8.00 or 8.33 MHz, depending on the HCLKOUT frequency. The clock divisor value is determined by strapping options as discussed in the Clock section.

For CPU or PCI initiated cycles to the ISA Bus, SYSCLK is stretched to synchronize BALE falling to the rising edge of SYSCLK. During CPU or PCI initiated cycles to the IB, BALE is normally driven high, synchronized to the rising edge of SYSCLK and then driven low to initiate the cycle on the ISA Bus. However, if the cycle is aborted, BALE remains high and is not driven low until the next cycle to the ISA Bus.

Data Byte Swapping (ISA Master or DMA to ISA Device)

The data swap logic is integrated in the IB. For slaves that reside on the ISA Bus, data swapping is performed if the slave (I/O or memory) and ISA Bus master (or DMA) sizes differ and the upper (odd) byte of data is being accessed. The data swapping direction is determined by the cycle type (read or write). Table 15 shows when data swapping is provided during DMA and ISA master cycles to ISA slaves.

Table 15. DMA Data Swap

DMA I/O Device Size	ISA Memory Slave Size	Swap	Comments (I/O) ⟷ Memory
8-bit	8-bit	No	SD[7:0] ⟷ SD[7:0]
8-bit	16-bit	No	SD[7:0] ⟷ SD[7:0]
8-bit	16-bit	Yes	SD[7:0] ⟷ SD[15:8]
16-bit	8-bit	No	Not Supported
16-bit	16-bit	No	SD[15:0] ⟷ SD[15:0]

Table 16. 16-bit Master to 8-bit Slave Data Swap

SBHE#	SA0	SD[15:8]	SD[7:0]	Comments
0	0	Odd	Even	Word Transfer (data swapping not required)
0	1	Odd	Odd	Byte Swap(1, 2)
1	0	—	Even	Byte Transfer (data swapping not required)
1	1	—	—	Not Allowed

NOTES:
1. For ISA master read cycles, the IB swaps the data from the lower byte to the upper byte.
2. For ISA master write cycles, the IB swaps the data from the upper byte to the lower byte.

Wait-State Generation

The IB adds wait-states to the following cycles, if IOCHRDY is sampled negated (low). Wait-states are added as long as IOCHRDY remains low.

- During Refresh and IB master cycles (not including DMA) to the ISA Bus.
- During DMA compatible transfers between ISA I/O and ISA memory only.

For ISA master cycles targeted for the IB's internal registers or main memory, the IB always extends the cycle by driving IOCHRDY low until the transaction is complete.

Cycle Shortening

The IB shortens the following cycles, if ZEROWS# is sampled asserted (low).

- During IB master cycles (not including DMA) to 8-bit and 16-bit ISA memory.
- During IB master cycles (not including DMA) to 8-bit ISA I/O only.

For ISA master cycles targeted for the IB's internal registers or main memory, the IB does not assert ZEROWS#. If IOCHRDY and ZEROWS# are sampled low at the same time, IOCHRDY will take precedence and wait-states will be added.

4.7 X-Bus

The 82420EX PCIset provides the decode (chip selects) and X-Bus buffer control (XBUSOE# and XBUSTR#) for a Real Time Clock, Keyboard Controller, and BIOS (Figure 9). The chip selects are generated combinatorially from the ISA SA[16:0] and LA[23:17] address bus. (Note that the ISA master must drive SA[19:16] and LA[23:17] low when accessing I/O space.) The IB also provides PS/2 mouse support via the IRQ12/M signal and coprocessor functions (FERR# and IGNNE#). The chip selects and X-Bus buffer control lines can be enabled/disabled via the XBCSA Configuration Register.

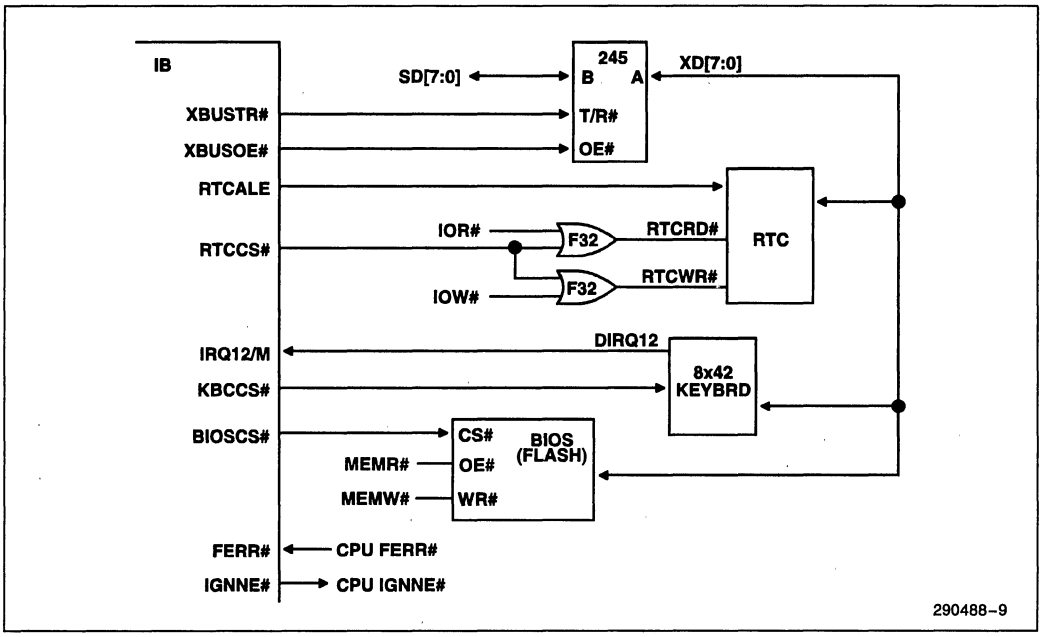


Figure 9. X-Bus Support

4.7.1 COPROCESSOR ERROR FUNCTION

The IB provides coprocessor error support for the CPU (enabled/disabled via the XBCSA Register). FERR# is tied directly to the coprocessor error signal of the CPU. If FERR# is asserted, an internal IRQ13 is generated and the INTR signal is asserted. When a write to I/O location F0h is detected, the IB negates IRQ13 (internal to the IB) and asserts IGNNE#. IGNNE# remains asserted until FERR# is negated. Note, that IGNNE# is not asserted unless FERR# is asserted.

4.7.2 MOUSE FUNCTION

The IB provides a mouse interrupt function (enabled/disabled via the XBCSA Register) on the IRQ12/M input signal. In this mode, a mouse interrupt generates an interrupt through IRQ12 to the Host CPU. The IB informs the CPU of this interrupt via a INTR. A read of 60h or 62h releases IRQ12. If bit 4=0 in the XBCSA Register, a read of address 60h or 62h has no effect on IRQ12/M. Reads and writes to this register flow through to the ISA Bus. For additional information, see the IRQ12/M description in the Signal Description.

4.8 System Arbitration

The 82420EX PCIset provides bus arbitration on the Host Bus, PCI Bus, and the PCI/IB Interface (to the ISA Bus). A device that is the master on any bus is the master of the entire system. (i.e., concurrency of more than one active master is not supported).

Signals associated with the system arbitration are the HOLD/HLDA signals (CPU Bus), PREQ[1:0]/PGNT[1:0]# signals (PCI Bus), and the LREQ#/LGNT# signals (PSC/IB Link Interface).

4.8.1 SYSTEM ARBITRATION SCHEME

When there are no active requests, the CPU owns the system. The system arbitration rotates between the PCI Bus, CPU Bus, and Link Interface Bus (on behalf of DMA and ISA Master devices), with the CPU permitted access every other transition.

NOTES:

1. The PSC, as a PCI master, never performs locked cycles. However, locked cycles are supported for PCI masters. When a PCI master performs a locked access, the arbitration is not changed until the locked sequence is completed.
2. After PGNT[1:0]# is asserted by the PSC, it is negated when FRAME# is sampled active (regardless the state of PREQ[1:0]#). The PCI master is expected to continue its current cycle (with potential multiple data phases), and then get-off the PCI Bus. The PSC does not release HOLD until the PCI Bus is idle. When a PCI master is re-tried by the PCI target, PGNT[1:0]# is already negated. Thus, the PCI master must get-off the bus. Since the PSC always gives the bus back to the CPU and the arbitration is rotated, PREQ[1:0]# can remain active as long as the PCI master has cycles to perform.
3. The PSC precludes fast back-to-back PCI master transactions. In addition, the PSC, as a PCI master, does not support fast back-to-back transactions.
4. When the PSC, as a PCI master, is re-tried, target-aborted or master-aborted, by a PCI target, the arbitration mechanism does not assert PGNT[1:0]# or LGNT#.

4.9 DMA Controller

The DMA circuitry incorporates the functionality of two 82C37 DMA controllers with seven independently programmable channels (Channels [3:0] and Channels [7:5]). The DMA supports 8/16-bit device size using ISA-compatible timings and 27-bit addressing as an extension of the ISA-compatible specification. The DMA channels can be programmed for either fixed (default) or rotating priority. The DMA controller also generates ISA refresh cycles. DMA Channel 4 is used to cascade the two controllers and default to cascade mode in the DMA Channel Mode (DCM) Register (Figure 10). In addition to accepting requests from DMA slaves, the DMA controller also responds to requests that are initiated by software. Software may initiate a DMA service request by setting any bit in the DMA Channel Request Register to a 1. The DMA controller for Channels [3:0] is referred to as "DMA-1" and the controller for Channels [7:4] is referred to as "DMA-2".

Each DMA channel is hardwired to the compatible settings for DMA device sizechannels [3:0] are hardwired to 8-bit, count-by-bytes transfers and channels [7:5] are hardwired to 16-bit, count-by-words (address shifted) transfers. The IB provides the timing control and data size translation necessary for the DMA transfer between the memory (ISA or main memory) and the ISA Bus I/O. ISA-Compatible DMA timing is supported. The DMA controller also features refresh address generation and auto-initialization following a DMA termination.

Note that a DMA device (I/O device) is always on the ISA Bus, but the memory referenced is located on either an ISA Bus device or in main memory. When the IB is running a DMA cycle, it drives the MEMR# or MEMW# strobes, if the address is less than 16 MBytes (000000–FFFFFFh). The IB always generates ISA-Compatible DMA memory cycles. The SMEMR# and SMEMW# are generated if the address is less than 1 MByte (0000000–00FFFFFh). To avoid aliasing problems when the address is greater than 16 MBytes (1000000–7FFFFFFh), the MEMR# or MEMW# strobe is not generated.

The channels can be programmed for any of four transfer modessingle, block, demand, or cascade. Each of the three active transfer modes (single, block, and demand), can perform three different types of transfers (read, write, or verify). Note that memory-to-memory transfers are not supported by the IB. The DMA supports fixed and rotating channel priorities.

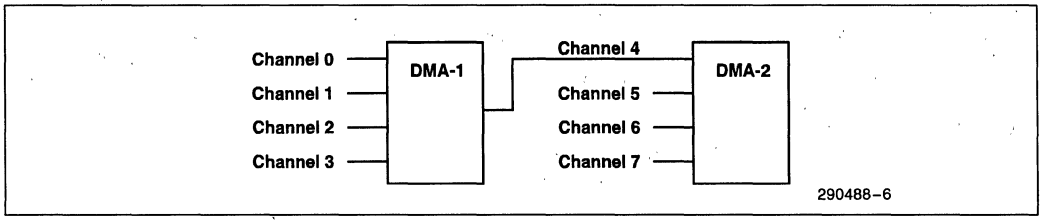


Figure 10. Internal DMA Controller

4.10 Interval Timer

The 82420EX PCIset contains three counters that are equivalent to those found in the 82C54 programmable interval timer. Each counter output provides a key system function. Counter 0 is connected to interrupt controller IRQ0 and provides a system timer interrupt for a time-of-day, diskette time-out, or other system timing functions. Counter 1 generates a refresh request signal and Counter 2 generates the tone for the speaker.

Counter 0 (System Timer)

This counter functions as the system timer by controlling the state of IRQ0 and is typically programmed for Mode 3 operation. The counter produces a square wave with a period equal to the product of the counter period (838 ns) and the initial count value. The counter loads the initial count value one counter period after software writes the count value to the counter I/O address. The counter initially asserts IRQ0 and decrements the count value by two each counter period. The counter negates IRQ0 when the count value reaches 0. It then re-loads the initial count value and again decrements the initial count value by two each counter period. The counter then asserts IRQ0 when the count value reaches 0, re-loads the initial count value, and repeats the cycle, alternately asserting and negating IRQ0.

Counter 1 (Refresh Request Signal)

This counter provides the refresh request signal and is typically programmed for Mode 2 operation. The counter negates refresh request for one counter period (838 ns) during each count cycle. The initial count value is loaded one counter period after being written to the counter I/O address. The counter initially asserts refresh request, and negates it for 1 counter period when the count value reaches 1. The counter then asserts refresh request and continues counting from the initial count value.

Counter 2 (Speaker Tone)

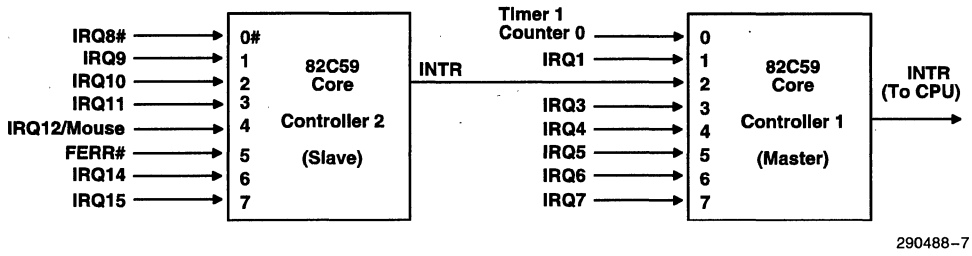
This counter provides the speaker tone and is typically programmed for Mode 3 operation. The counter provides a speaker frequency equal to the counter clock frequency (1.193 MHz) divided by the initial count value. The speaker must be enabled by a write to I/O address 061h.

4.11 Interrupt Controller

The 82420EX PCIset contains an ISA-compatible interrupt controller that incorporates the functionality of two 82C59 interrupt controllers. The two controllers (CNTRL-1and CNTRL-2) are cascaded allowing thirteen external and three internal interrupts (Figure 11). CNTRL-1 and CNTRL-2 are initialized separately and can be programmed to operate in different modes. CNTRL-1 is connected as the master interrupt controller and CNTRL-2 is connected as the slave interrupt controller. The three internal interrupts are used for internal functions only and are not available to the user. IRQ2 cascades the two controllers. IRQ0 provides a system timer interrupt and is tied to the Interval Timer, Counter 0. IRQ13 is connected internally to FERR# for coprocessor error support. The remaining thirteen interrupt lines (IRQ[15,14,12:9,8#,7:3,1]) are available for external system interrupts. Edge or level sense selection is programmable on an individual channel by channel basis. Interrupt steering permits two programmable interrupts (PIRQ0# and PIRQ1#) to be internally routed (steered) to one of eleven interrupts (IRQ[15,14,12:9,7:3]).

NOTES:

1. The standard external IRQ12 signal function or internally generated IRQ12/Mouse function are selected via the XBCSA Register.
2. The IB translates the CPU generated interrupt acknowledge cycle internally into the two INTA# pulses expected by the interrupt controller system.



NOTE:
IRQ0, IRQ2, and IRQ13 are internally connected and do not appear externally.

Figure 11. Block Diagram of the Interrupt Controller

4.11.1 PROGRAMMING THE INTERRUPT CONTROLLER

The Interrupt Controller accepts two types of command words generated by the CPU or bus master—Initialization Command Word (ICWx) and Operation Command Word (OCWx).

Initialization Command Words (ICWs)

Before normal operation can begin, each interrupt controller in the system must be initialized. In the 82C59, this is a two to four byte sequence. However, for the 82420EX PCIset, each controller must be initialized with a four byte sequence. This four byte sequence is required to configure the interrupt controller correctly for the IB implementation. This implementation is ISA compatible.

The four initialization command words are referred to by their acronyms: ICW1, ICW2, ICW3, and ICW4. These command registers are discussed in Section 3.0, Register Description. The sequence must be executed for CNTRL-1 and CNTRL-2. ICW1, ICW2, ICW3, and ICW4 must be written in order. Any divergence from this sequence, such as an attempt to program an OCW, will result in improper initialization of the interrupt controller and unexpected, erratic system behavior. It is suggested that CNTRL-2 be initialized first, followed by CNTRL-1.

Operation Command Words (OCWs):

These are the command words which dynamically reprogram the Interrupt Controller to operate in various interrupt modes. OCW1 masks interrupt lines. OCW2 controls rotation in interrupt rotation priority mode and the End of Interrupt (EOI). OCW3 sets up reads of the ISR and IRR, enables/disables the Special Mask Mode (SMM), and sets up the polled interrupt mode. The OCWs can be invoked any time after initialization.

4.11.2 EDGE AND LEVEL INTERRUPT TRIGGERED MODE

In ISA systems, this mode is programmed using bit 3 in ICW1. For the IB, this bit is disabled and the Edge/Level Control Registers (ELCR1 and ELCR2) are included that select edge and level triggered mode per interrupt input. The default programming is equivalent to programming the LTIM bit (ICW1 bit 3) to a 0 (all interrupts selected for edge triggered mode). Note, that IRQ[13,8#,2,1,0] can not be programmed for level sensitive mode.

In both the edge and level triggered modes, the IRQx input must remain active until after the falling edge of the first INTA#. If IRQx is negated before this time, a default IRQ7 occurs when the CPU acknowledges the interrupt. This can be a useful safeguard for detecting interrupts caused by spurious noise glitches on the IRQ inputs. To implement this feature, the IRQ7 routine is used for "clean up" by simply executing a return instruction, thus ignoring the interrupt. If IRQ7 is needed for other purposes, a default IRQ7 can still be detected by reading the ISR. A normal IRQ7 interrupt sets the corresponding ISR bit; a default IRQ7 does not set this bit. If a default IRQ7 routine occurs during a normal IRQ7 routine, however, the ISR remains set. In this case, it is necessary to keep track of whether or not the IRQ7 routine was previously entered. If another IRQ7 occurs, it is a default.

4.11.3 INTERRUPT STEERING

The IB contains two programmable interrupts (PIRQ0 and PIRQ1#) that can be internally routed to one of eleven interrupts (IRQ[15,14,12:9,7:3]) by programming the PIRQx Route Control Registers. One or both PIRQx# lines can be routed to the same IRQx input or interrupt steering can be disabled.

The PIRQx# lines are defined as active low, level sensitive to allow multiple interrupts on a PCI board to share a single line across the connector. When a PIRQx# is routed to a specified IRQ line, the software must change the IRQ's corresponding ELCR bit to level sensitive mode. Note, that this means that the selected IRQ can no longer be used by an ISA device, unless that ISA device can respond as an active low level sensitive interrupt.

4.12 L2 Cache

The L2 cache memory array contains a cache data RAM with a selectable storage capacity of either 64, 128, 256, or 512 KBytes. The cache data RAM is a direct-mapped memory array, write-through or write-back, that can be organized in either an interleaved or non-interleaved configuration. In addition to the cache data RAM, the L2 cache contains a RAM array that holds the tag address and a dirty bit that is associated with each line of data. Table 17 provides a summary of the L2 cache. A valid bit is not used in this architecture. The L2 cache is programmed via the SCC Register.

Table 17. L2 Cache Features

Feature	Description
Organization	Direct mapped
Capacity	64, 128, 256, or 512 KByte
Data Banks	1 or 2, depending on capacity
Line Size	16 bytes
Tag Size	8 bits
Cacheable Main Memory	8 MBytes to 128 MBytes
Allocation Policy	Allocate on CPU reads; no allocate on writes
Cache Policy	Non-cacheable, write-through (WT), and write-back (WB)

4.12.1 L2 CACHE SIZES/PERFORMANCE

The PSC allows four cache sizes. Table 18 shows the tag and data SRAMs used for various user settings. The PSC supports 15 ns tag SRAMs and 20 ns data SRAMs for the L2 cache at all frequencies. Table 19 shows the range of L2 performance achievable.

Table 18. L2 Options and Component List

Cache Size	Data RAMS	Tag Bits/Cacheable Main Memory	Tag Store
64 KByte	8 8K x 8	A[23:16]/16 MB	4K x 9
128 KByte	4 32K x 8	A[24:17]/32 MB	8K x 9
256 KByte	8 32K x 8	A[25:18]/64 MB	32K x 9
512 KByte	4 128K x 8	A[26:19]/128 MB	32K x 9

Table 19. Performance

Cycle Type	Tag Speed	Data Speed	25 MHz	33 MHz	L1
Interleaved Read	15 ns	15 ns/20 ns	2-1-1-1	2-1-1-1	WT/WB
Non-Interleaved Read	15 ns	15 ns	2-1-1-1	2-2-2-2(1)	WT/WB
Non-Interleaved Read	15 ns	20 ns	2-1-1-1	2-2-2-2	WT/WB
Interleaved Write	15 ns	15 ns/20 ns	2-1-1-1(2)	2-1-1-1(2)	WT/WB
Non-Interleaved Write	15 ns	15 ns/20 ns	2-1-1-1(2)	2-1-1-1(2)	WT
Non-Interleaved Write	15 ns	15 ns/20 ns	3-2-2-2	3-2-2-2	WB

NOTES:
1. 2-1-1-1 may be used only with minimum margin design (light Host Bus loading) and 12 ns Data SRAMs.
2. Programmable option and applies to cache hit dirty write cycles.

4.12.2 CACHE OPERATIONS

During a CPU memory read or write operation, the PSC searches the cache first. Then, if required, it searches main memory for addressed data locations. The L2 cache operation is determined by the cache policy (non-cacheable, write-through, or write-back) as determined by the Secondary Cache Control Register (see Section 3.0, Register Description). If the caching policy is non-cacheable, the cache is not accessed.

Write-through and write-back are two caching policies for updating main memory with data in the cache. For these policies, the cache operation is determined by the type of operation as follows:

CPU Write Cycle

If the caching policy is write-through and there is a cache hit, both the cache and main memory are updated. If there is a cache miss, only main memory is updated. The cache is not updated (no write-allocate).

If the caching policy is write-back and there is a cache hit, only the cache is updated; main memory is not affected. If there is a cache miss, only main memory is updated. The cache is not updated (no write-allocate).

CPU Read Cycle

If there is a cache hit, the cache operation is the same for both write-through and write-back. In this case, data is transferred from the cache to the CPU. Main memory is not accessed.

If there is a cache miss, the line containing the requested data is transferred from main memory to the cache. During the cache line update, a line fill (burst read) memory operation containing four dword transfers occurs on the Host Bus to bring in the new line. This occurs for both write-through and write-back. However, in the case of write-back, if the cache line fill is to a dirty line (D=1), the dirty line is first written back to main memory before the new line is brought into the cache. For a dirty line write-back operation, the PSC first performs a read from the dirty cache line and writes the data to main memory. Then, the PSC updates the cache (both L1 and L2 simultaneously) with the new line.

4.12.3 CACHE CONSISTENCY

The Snoop mechanism in the PSC ensures data consistency between cache (both L1 and L2 caches) and main memory. Note that, for write-back cache control, the term "Inquire" is sometimes used to describe the snooping operation. In this document, the term "Snoop" is used for both write-through and write-back cache policies.

The PSC monitors PCI master, ISA master and DMA accesses to main memory and when needed, initiates an inquire (snoop) cycle to the L1 and L2 caches. The snoop mechanism guarantees that consistent data is always delivered to the host CPU, PCI master, ISA master or DMA.

4.12.4 INITIALIZING THE L2 CACHE

The 82420EX PCIset L2 cache architecture does not use a valid bit. Instead, BIOS initializes the L2 cache with valid data. After initialization, the cache controller maintains data coherency between the cache and main memory by keeping all cache lines valid. The PSC cache controller has two special bits to support initialization—Force Hit (SCC Register bit 6) and Force Miss Clean (SCC Register bit 5).

BIOS can use the Force Hit bit to determine the size of the L2 cache. When Force Hit is enabled, BIOS can attempt to alias cache locations on writes. For example, to check a 128 KByte cache size, BIOS writes location "x" with value 00h. BIOS can then write location 128k + x with 11h. With Force Hit enabled and the cache in write-back mode, the write does not access main memory. When a value of 00h is read from location "x", the L2 cache is greater than 128 KBytes. If 11h is read, the L2 cache is smaller than 128 KBytes. This process is repeated for all cache boundaries.

The Force Miss Clean bit causes all accesses to the L2 cache to be treated as a clean miss. This allows BIOS to initialize the L2. At start-up BIOS enables Force Miss Clean and reads a block of memory equal to the cache size. This initializes the L2 cache with data that is coherent with main memory.

4.12.5 CACHE LINE DESCRIPTION

Each line consists of four dwords of data, a tag field and a Dirty (D) bit. The tag field and control bits are read/written by the PSC during normal cache operations and are not accessible by software.

D: Dirty

The Dirty bit is set to 1 by the PSC to indicate that data modified in the cache line has not been written back to main memory.

Tag: Real Address Tag

The PSC uses the Tag field for cache line hit/miss determination. The width of the Tag field is fixed at 8 bits. The table below shows the real address bits that are stored in the Tag field as a function of the cache sizes.

Cache Size	8-Bit Tag
64 KBytes	A[23:16]
128 KBytes	A[24:17]
256 KBytes	A[25:18]
512 KBytes	A[26:19]

Doubleword[3:0]

Each line of the cache data RAM contains four dwords.

4.12.6 L2 CACHE STRUCTURE

The tag is 8 bits plus a dirty bit. Either interleaved or non-interleaved organizations are permitted. The PSC will assert the COE[1:0]# and CWE[1:0]# signals to both banks, even when programmed for non-interleaved mode. The interleaved L2 can provide zero wait-state reads and writes. Figure 12 shows the interconnection between the PSC and an interleaved L2. The PSC has a variety of programmable access timings to support 25 MHz and 33 MHz. These options are controlled by the Secondary Cache Control Register (SCC).

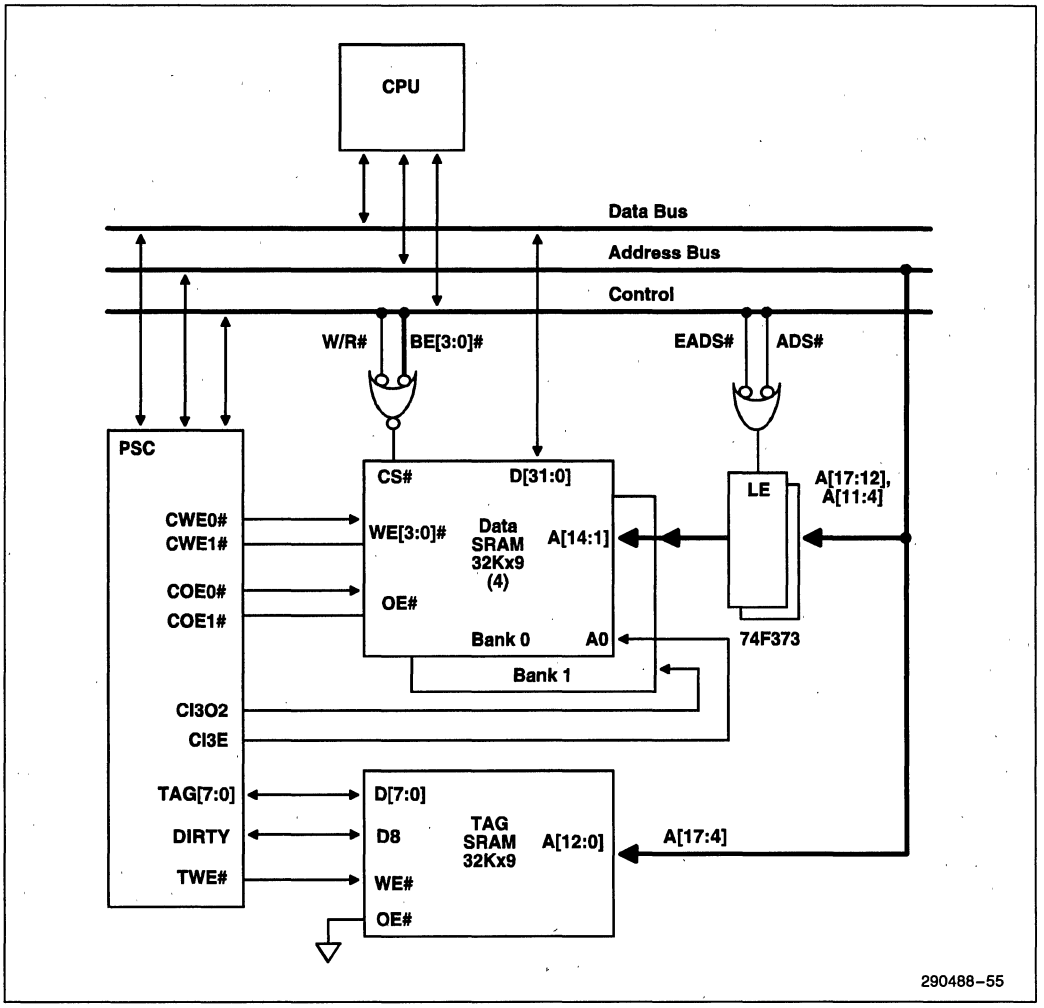


Figure 12. The PSC with an Interleaved L2 (256 KBytes Data)

4.13 Dram Interface

The DRAM controller interfaces main memory to the Host Bus, PCI Bus, and ISA Bus. The PSC provides the control signals, address lines, and data path control. A two-way interleaved DRAM organization is supported for optimum main memory performance.

Up to ten single-sided SIMMs or four double-sided and two single-sided SIMMs provide a maximum of 128 MBytes of main memory. The DRAM controller interface is fully configurable through a set of control registers (the DRAM Control Mode Register, the DRAM Memory Hole Register, and the five DRAM Row Boundary [DRB] Registers).

The PSC controls a 64-bit memory array (72-bit including parity) and/or a 32-bit memory array (36-bit including parity) ranging in size from 1 to 128 MBytes using industry standard 36-bit wide memory modules with fast page-mode DRAMs. Both single- and double-sided SIMMs are supported. The eleven multiplexed address lines (MA[10:0]) permit the use of 256Kx36, 1Mx36, and 4Mx36 SIMMs. Both interleaved and non-interleaved rows are supported simultaneously. Five RAS# lines enable up to five rows of DRAM. Eight CAS# lines allow byte control over the array during read and write operations. The PSC supports 70 ns DRAMs. Page mode accesses efficiently transfer data in bursts. Parity support is optional.

The PSC DRAM performance is controlled through programmable wait-states. Various DRAM timing parameters may be set in the DRAM Control Register. Programmable timings support 70 ns DRAMs at 25 MHz and 33 MHz. Programmable parameters include RAS precharge, CAS precharge, CAS low time, MA setup time, and MA hold time. The PSC provides RAS only refresh, de-coupled from ISA refresh, and hidden from any access.

4.13.1 DRAM ADDRESS TRANSLATION

The multiplexed row/column address to the DRAM memory array is provided by the MA[10:0] signals and is derived from the host address bus as defined by Table 20. The page size is 2 KBytes for non-interleaved rows and 4 KBytes for interleaved rows. The page offset address is driven over the MA[8:0] lines when driving the column address. In non-interleaved rows the PSC drives address bit 2 on the MA8 line, minimizing the multiplexing required. The MA[10:0] lines are translated from the address lines A[24:3] for all memory accesses.

4.13.2 DRAM STRUCTURE

Figure 13 illustrates an 8-SIMM configuration supporting single-sided SIMMs. A row in the DRAM array is made up of two SIMMs that share a common RAS# line. SIMM0 and SIMM1 comprise row 0 and are connected to RAS0#. Within any given row, the two SIMMs must be the same size. Among the four rows, SIMM densities can be mixed in any order (i.e., there are no restrictions on the ordering of SIMM densities among the four rows). Any row may also contain a single SIMM (non-interleaved). This allows the user to upgrade the 82420EX PCIset platform one SIMM at a time. Each row is controlled by up to 8 CAS lines. Any row that is populated with only one SIMM must be connected to the low order CAS lines (CAS[3:0]#). The MA[10:0] and WE# lines should be externally buffered if a load of more than two double-sided SIMMs is implemented. Two buffered copies of the signals are recommended to drive the four row array. Three buffered copies of the signals are recommended to drive the five row array.

Figure 14 illustrates a 3-SIMM configuration using one single-sided SIMM in row one, and two double sided SIMMs in row 2. In this configuration, single- and double-sided SIMMs can be mixed. For example, if a single-sided SIMM is installed into the socket marked SIMM0 (connected to RAS0#) and RAS1# is not connected, row 0 is then populated and row 1 is empty. Two double-sided SIMMs could then be installed in the sockets marked SIMM2 and SIMM3, populating rows 2 and 3. For systems with no more than 2 SIMMs, the 244's buffering MA[10:0] and WE#, as well as the 245's on the host data bus, may be omitted. (Note that the 245's on the Host Data Bus are recommended at 50 MHz, regardless of the number of SIMMs.)

Table 20. DRAM Address Translation for Interleaved Rows

MA[10:0]	10	9	8	7	6	5	4	3	2	1	0
Interleaved Rows											
Column Address	A23	A21	A11	A10	A9	A8	A7	A6	A5	A4	A3
Row Address	A24	A22	A20	A19	A18	A17	A16	A15	A14	A13	A12
Non-Interleaved Rows											
Column Address	A23	A21	A2	A10	A9	A8	A7	A6	A5	A4	A3
Row Address	A22	A20	A11	A19	A18	A17	A16	A15	A14	A13	A12

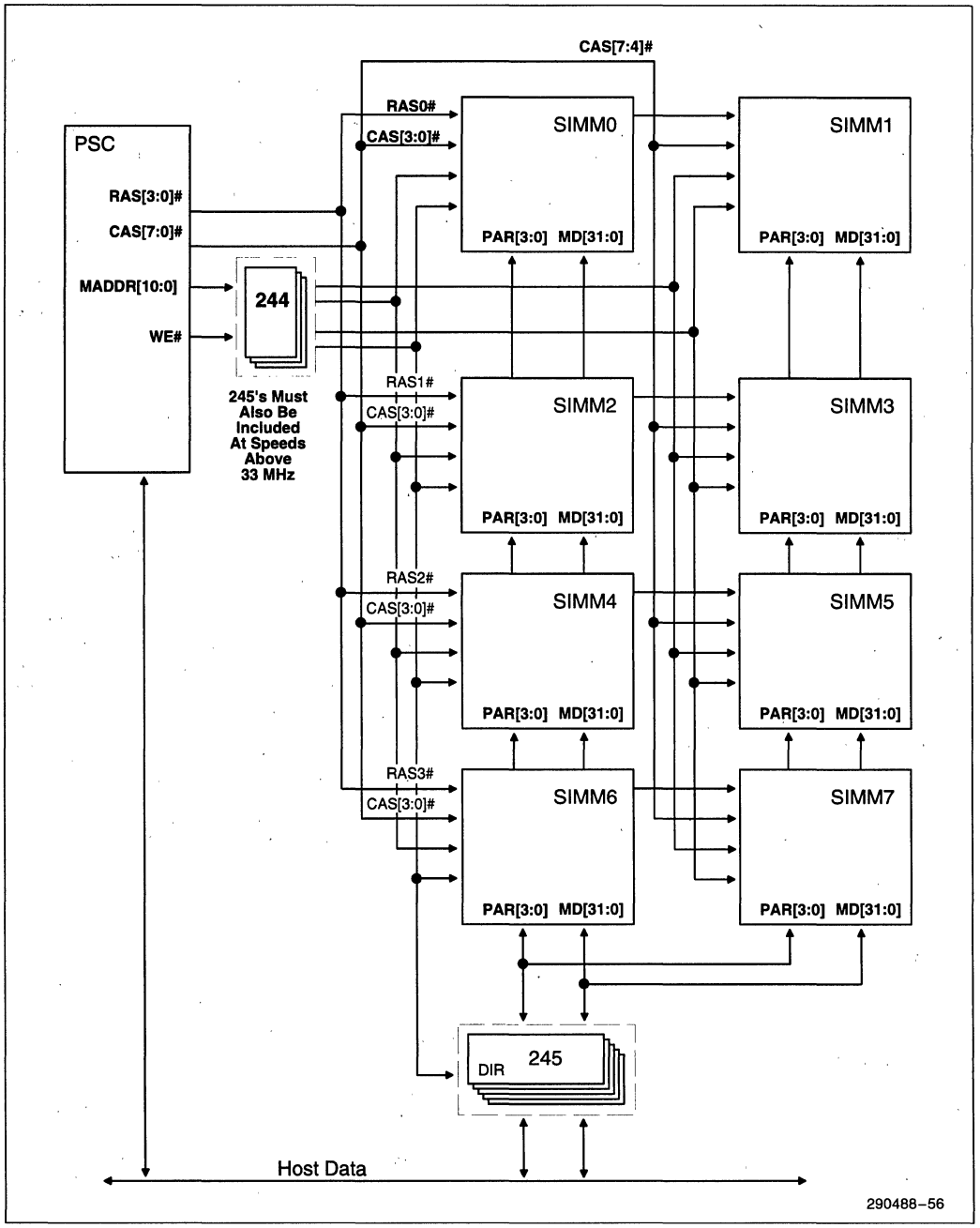


Figure 13. 8-SIMM Configuration Supporting Single-Sided SIMMs

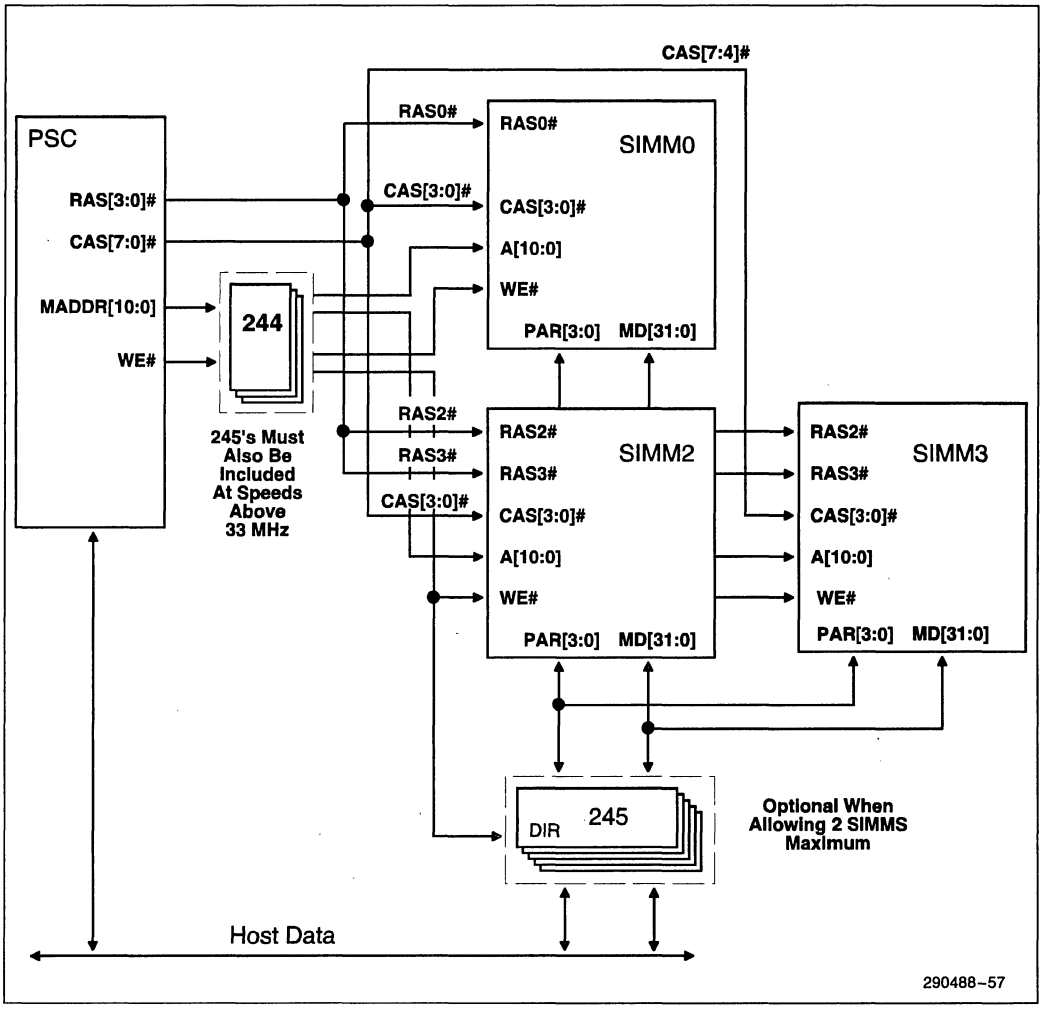


Figure 14. 3-SIMM Configuration (One Single-Sided SIMM in Row One and Two Double-Sided SIMMs in Row Two)

4.13.3 DRAM SIMM SIZE/DENSITY OPTIONS

Providing support for both interleaved and non-interleaved rows gives the user a wide range of DRAM population options. In any row the PSC supports address depths of 256K, 1M, or 4M. Each row may be populated with one or two SIMMs.

4.13.4 DRAM PAGE MODE

The PSC may be programmed to leave the RAS lines active after a DRAM access for faster page mode accesses. The mode is programmed in the DRAM Control Register. When Page Mode is enabled, the RAS lines remain active after the access.

The next access is considered a page hit if the access is to the same page. The PSC has a 2 KByte page size for non-interleaved rows and a 4 KByte page size when accessing an interleaved row. If the page mode is not enabled all accesses are row misses.

4.13.5 PROGRAMMABLE WAIT-STATES

4.13.5.1 RAS Precharge

RAS precharge impacts page miss accesses, as well as the refresh time. In a page miss, the active RAS line must be negated and a new row address strobed into the DRAMs. When negated, RAS precharge determines the number of cycles before the RAS line can be re-asserted. Similarly, the RAS precharge determines the time RAS must be negated before and after refresh.

4.13.5.2 CAS Read Time

CAS read time indicates how long to leave CAS low after asserted during a DRAM read and how long it is negated between access. In the case of interleaved DRAM access, the CAS high time implied by this setting is ignored as this high time is more constrained by the opposite banks low time.

4.13.5.3 CAS Write Time

CAS write time controls the CAS waveform for DRAM writes. The high time defined by the CAS write time setting is ignored for interleaved rows.

4.13.5.4 MA Setup Time

The MA setup time defines the number of cycles after MA is switched before RAS or CAS are driven active. DRAMs latch row and column address when RAS and CAS fall. The setup time of the addresses to RAS/CAS for all DRAMs is 0 ns. The PSC supports direct drive of the MA lines, which removes any external logic between the MA on the PSC and the DRAM. When there is no external buffering, the MA-to-DRAM path and the CAS-to-DRAM path are well matched. In these cases an aggressive MA setup time can be programmed. When the external buffers are present, there could be mismatch in the paths, and a more conservative MA setup should be chosen.

4.13.5.5 MA Hold Time

The MA hold time defines the number of clocks after RAS or CAS have been asserted before MA can be changed. This value is determined in a manner similar to MA setup time. DRAM requirements for 60 ns and 70 ns DRAMs range from 10 ns to 15 ns.

4.13.6 DRAM PERFORMANCE

Table 21 summarizes DRAM performance for various programming options for both 60 ns and 70 ns DRAMs. Other cycle constraints that are met by design include DRAM access from RAS falling and DRAM access from row address, and many others. All accesses shown below assume no wait-states required for other Host Bus devices (L2, etc.). If, as discussed in the previous section, buffers must be turned around and contention with other host devices avoided, the minimum lead off for read page hits will be lengthened by one cycle.

Table 21. DRAM Performance

System	RAS pre-charge	CAS Read	CAS Write	MA Setup	MA Hold	Clock Freq.	Performance No L2	Performance With L2
60 ns DRAM Min	1	1/1	1/1	.5	.5	25 MHz	Ird: 3/5/5-1-1-1 NIrd: 3/5/5-2-2-2 wr: 3/5/5-2-2-2	Ird: 3/5/5-1-1-1 NIrd: 3/5/5-2-2-2 Same as No L2
Margin	1.5	1.5/.5	1/1	1	.5	33 MHz	Ird: 3/6/7-2-2-2 NIrd: 3/6/7-2-2-2 wr: 3/5/6-2-2-2	Ird: 4/6/7-2-2-2 NIrd: 4/6/7-2-2-2 Same as No L2
60 ns DRAM High	1.5	1.5/.5	1/1	.5	.5	25 MHz	Ird: 3/5/6-2-2-2 NIrd: 3/5/6-2-2-2 wr: 3/5/6-2-2-2	Ird: 4/5/6-2-2-2 NIrd: 4/5/6-2-2-2 Same as No L2
Margin	1.5	1.5/.5	1/1	1	1	33 MHz	Ird: 3/7/7-2-2-2 NIrd: 3/7/7-2-2-2 wr: 3/6/7-2-2-2	Ird: 4/7/7-2-2-2 NIrd: 4/7/7-2-2-2 Same as No L2
70 ns DRAM Min	1.5	1.5/.5	1/1	.5	.5	25 MHz	Ird: 3/5/6-2-2-2 NIrd: 3/5/6-2-2-2 wr: 3/5/6-2-2-2	Ird: 4/5/6-2-2-2 NIrd: 4/5/6-2-2-2 Same as No L2
Margin	2	1.5/.5	1/1	1	.5	33 MHz	Ird: 3/6/7-2-2-2 NIrd: 3/6/7-2-2-2 wr: 3/5/7-2-2-2	Ird: 4/6/7-2-2-2 NIrd: 4/6/7-2-2-2 Same as No L2
70 ns DRAM High	2	1.5/.5	1/1	.5	.5	25 MHz	Ird: 3/5/7-2-2-2 NIrd: 3/5/7-2-2-2 wr: 3/5/6-2-2-2	Ird: 4/5/7-2-2-2 NIrd: 4/5/7-2-2-2 Same as No L2
Margin	3	2/1	1/1	1	.5	33 MHz	Ird: 4/7/9-2-2-2 NIrd: 4/7/9-3-3-3 wr: 3/6/8-2-2-2	Ird: 4/7/9-2-2-2 NIrd: 4/7/9-3-3-3 Same as No L2

NOTES
I = Interleaved
NI = Non-Interleaved

4.14 Power Management

The 82420EX PCIset has extensive power management capability permitting a system to operate in a low power state without being powered down. In a typical desktop personal computer there are two states—Power On and Power Off. Leaving a system powered on when not in use wastes power. The 82420EX PCIset provides a Fast On/Off feature that creates a third state called Fast Off (Figure 15). When in the Fast Off state, the system consumes less power than the Power On state.

The 82420EX PCIset's power management architecture is based on three functions—System Management Mode (SMM), Clock Control, and Advanced Power Management (APM). Software (called SMM code) controls the transitions between the Power On state and the Fast Off state. The IB invokes this software by generating an SMI to the CPU (asserting the SMI# signal). A variety of programmable events are provided that can generate an SMI. The SMM code places the system in either the Power On state or the Fast Off state.

A Fast On event is an event that instructs the computer (via an SMI to the CPU) to enter the Power On state in anticipation of system activity by the user. Fast On events are programmable and include moving the mouse, pressing a key on the keyboard, an external hardware event, an incoming call to a system FAX/Modem, a RTC alarm, or the operating system.

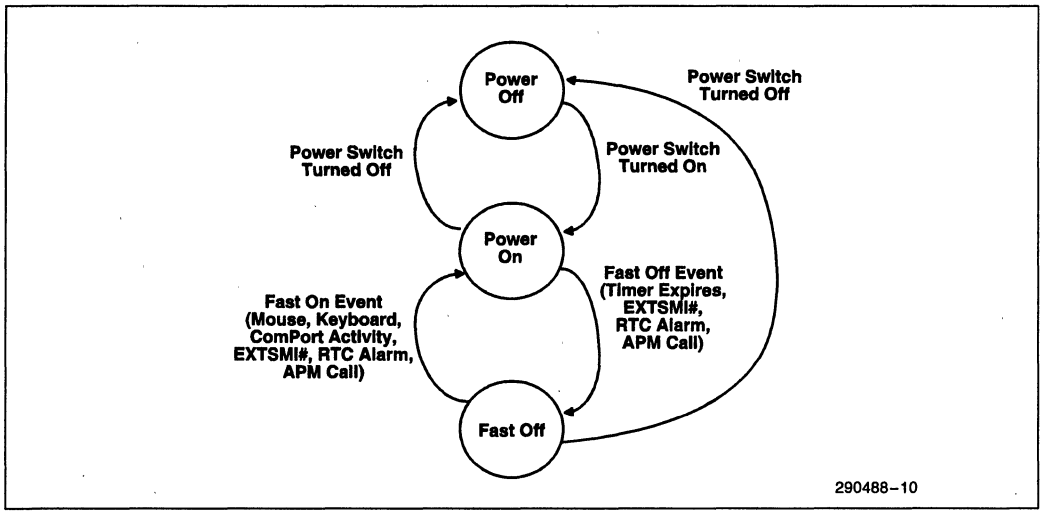


Figure 15. Fast On/Off Flow

4.14.1 SMM MODE

SMM mode is invoked by asserting the SMI# signal to the CPU. The PCIset provides a variety of programmable events that can generate an SMI. When the CPU receives an SMI, it enters SMM mode and executes SMM code out of SMRAM. The SMM code places the system in either the Power On state or the Fast Off state. In the Power On state, the computer system operates normally. In this state one of the four programmable events listed below can trigger an SMI.

1. A global idle timer called the Fast Off timer expires (an indication that the end user has not used the computer for a long period of time).
2. The EXTSMI# pin is asserted.
3. A RTC alarm interrupt is detected.
4. The operating system issues an APM call.

4.14.2 SMI SOURCES

The SMI# signal can be asserted by hardware events, an external SMI event (EXTSMI#), and software events (via the APMC and APMS Registers). Enable/disable bits (in the SMIEN Register) permit each event to be individually masked from generating an SMI. In addition, the SMI# signal can be globally enabled/disabled in the SMICNTL Register. Status of the individual events causing an SMI is provided in the SMIREQ Register. For detailed information on the SMI control/status registers, refer to Section 3.0, Register Description.

Hardware Interrupt Events

Hardware events are enabled/disabled from generating an SMI in the SMIEN Register. The hardware events consist of IRQ[12,8#,4,3,1] and the Fast Off Timer. When enabled, the occurrence of the corresponding hardware event generates an SMI (asserts the SMI# signal), regardless of the current power state of the system.

Fast Off Timer

The Fast Off Timer is used to indicate (through an SMI) that the system has been idle for a pre-programmed period of time. The timer counts down at a selectable rate from a programmed start value and when the count reaches 00h, an SMI is generated. The timer decrement rate can be set to 1 count every minute, ms, or HCLKIN (via the SMICNTL Register) and is re-loaded each time a System Event occurs.

System events are programmable events that can keep the system in the Power On state when there is system activity. These events are indicated by the assertion of IRQ[15:9,8#,7:3,1:0], NMI, or SMI signals.

In addition to system events, *break events* cause the system to transition from a Fast Off state to the Power On state. System events (and break events) are enabled/disabled in the SEE Register. When enabled and the associated hardware event occurs (signal is asserted), the Fast Off Timer is re-loaded with its initial count.

EXTSMI#

The EXTSMI# input pin provides the system designer the capability to invoke SMM with external hardware. For example, the EXTSMI# input could be connect to a "green button" permitting the user to enter the Fast Off state by depressing a button. The EXTSMI# generation of an SMI is enabled/disabled in the SMIEN Register.

Software Events

Software events (accessing the APMx Registers) indicate that the OS is passing power management information to the SMI handler. There are two Advanced Power Management (APM) registersAPM Control (APMC) and APM Status (APMS) Registers. These registers permit software to generate an SMI; by writing to the APMC Register. For example, the APMC can be used to pass an APM command between APM OS and BIOS and the APMS Register could be used to pass data between the OS and the SMI handler. For detailed descriptions of these registers See Section 3.0, Register Description. Note that the two APM Registers are located in normal I/O space. The remaining power management registers are located in PCI configuration space.

4.14.3 SMI (SMI#) AND INTERRUPT (INTR) ORDERING

To maintain the SMI#/INTR order, an interrupt blocking mechanism has been incorporated into the IB. The blocking mechanism blocks interrupt requests that can generate an SMI# from being processed by the interrupt controller until the SMI# has been serviced by the SMM code. This blocking mechanism is selective and only affects the IRQ[12,8#,4,3,1] signals that are enabled to generate an SMI# (via the SMIEN Register). In addition, the blocking mechanism is only invoked if the SMI# signal is unmasked (via the SMICNTL Register). Note that PIRQ[1,0]s routed to one of the dual-purpose interrupt request lines are also affected by the blocking mechanism. Thus, the following criteria applies to the blocking mechanism:

1. The assertion of IRQ[12,8#,4,3,1] are blocked if the interrupt request line is programmed for SMI (i.e., the interrupt request line is enabled for SMI via the SMIEN Register and the SMI# signal is not masked via the CSMIGATE bit in the SMICNTL Register).
2. A blocked IRQ request is unblocked and processed by the interrupt controller when software masks the SMI# signal by setting the CSMIGATE bit to 0 in the SMICNTL Register.
3. IRQs that are already asserted when SMI# is unmasked (CSMIGATE set from a 0 to 1) are not blocked and are processed by the interrupt controller.

The following are BIOS and hardware considerations regarding the SMI#/INTR ordering:

- To process blocked, active IRQs, software (SMM code only) should mask the SMI# signal. If SMI# is masked outside SMM code when an IRQ that can generate an SMI# and the INTR signal is active, the SMI# and INTR order is not guaranteed.
- The SMI software handler should use the SMIREQ Register status bits and not the interrupt controller IRR to dispatch the routine (vector to the appropriate SMI function). By using the SMIREQ Register, the SMI handler has the freedom to mask the SMI# signal before or after the execution of the SMI function. Note that the IRR is updated only when the SMI# signal is masked.
- The IB updates new active SMI sources while the system is in SMM, independent of the state of the mask/unmask of the SMI# signal. When the SMI handler completes the execution of a certain SMI function, it should check whether other active SMI sources exist and service these sources before executing the **RSM** instruction.
- When the SMIREQ Register indicates that all SMI sources are inactive, the SMI handler should unmask the SMI# signal and execute the **RSM** instruction. Note that, all active SMI sources (status bits not set to 0 in the SMIREQ Register), will generate a new SMI# to the CPU when SMI# is unmasked.
- The SMI handler should not check for active SMI sources, or execute the new sources, after the SMI# signal is unmasked. Such an SMI source will generate a new active SMI# and the CPU will latch the new SMI# (and recognize it after **RSM**). Thus, executing the SMI source before **RSM** will cause a spurious SMI# after the **RSM** execution.

4.14.4 CLOCK CONTROL

The CPU can be put in a low power state by asserting the STPCLK# signal. STPCLK# is an interrupt to the CPU. However, for this type of interrupt, the CPU does not generate an interrupt acknowledge cycle. Once the STPCLK# interrupt is executed, the CPU enters the Stop Grant state. In this state, the CPU's internal clocks are disabled and instruction execution is stopped. The Stop Grant state is exited when the STPCLK# signal is negated.

Software can assert STPCLK#, if enabled via the SMICNTL Register, by a read of the APMC Register. Note that STPCLK# can also be periodically asserted by using clock throttling as described below.

The IB automatically negates STPCLK# when a break event occurs (if enabled in the SEE Register) and the CPU stop grant special cycle has been received. Software can negate STPCLK# by disabling STPCLK# in the SMICNTL Register or by a write to the APMC Register.

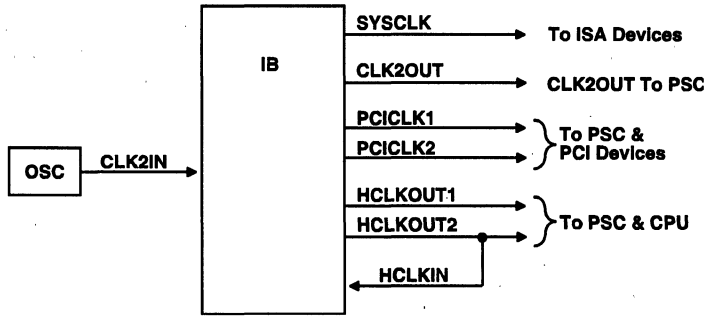
Clock Throttling (Emulating Clock Division)

Clock throttling permits the IB to periodically place the CPU in a low power state. This emulates clock division. When clock throttling is enabled, the CPU runs at full frequency for a pre-defined time period and then is stopped for a pre-defined time period. The Run/Stop time interval ratio emulates the clock division effect from a power/performance point of view. However, clock throttling is more effective than dividing the CPU frequency. For example, if the CPU is in the Stop Grant state and a system break event occurs, the CPU clock returns to full frequency. In addition, there is no recovery time latency to start the clock.

Two programmable 8-bit clock throttle timer control registers set the STPCLK# high (negate) and low (assert) times—the CTLTMRH and CTLTMRL Registers. The timer is clocked by a 32 μs internal clock. This allows a programmable timer interval for both the STPCLK# high and low times of 0-8 ms.

4.15 Clocks

The IB contains a clock generation unit that generates the system clocks. The IB generates HCLKOUTx (host clocks), PCICLKx (PCI clocks), SYSCLK (ISA System clock), and CLK2OUT (delayed version of CLK2IN) signals to the system (Figure 16). An external clock driver is not required. Two HCLKOUT signals and two PCICLK signals are provided to drive the loads. One of the HCLKOUT outputs is fed back to the HCLKIN pin to become the IB clock. CLK2OUT is used by the PSC. The IB uses a 2X clock input (CLK2IN) as the source to generate the system clocks. For CPU or PCI initiated cycles to the ISA Bus, SYSCLK is stretched to synchronize the falling edge BALE to the rising edge of the SYSCLK.



NOTE:
HCLKIN can be connected to either HCLKOUT1 or HCLKOUT2.

Figure 16. System Clock Distribution

There are three clock configurations (strapping options) that set the clock divisors as shown in Table 22. The IB samples the CMDV# and SIDLE# signals on the rising edge of PWROK to determine the clock divisor value. The IB CLK2IN pin is divided by either 1 or 2 to generate HCLKOUT[2,1] and PCICLK[2,1]. One of the HCLKOUT signals is fed back to the HCLKIN input of the IB and divided by either 3, 4, 5, or 6, to generate SYSCLK. Note that the configuration information provided in Table 22 is software accessible in the Host Bus Select Register.

4.15.1 CLOCK LAYOUT/LOADING RECOMMENDATION

A spice analysis was done on the HCLKOUT, PCICLK, and CLK2OUT signals to determine the loading requirements necessary to maintain the clock rising edge skew values shown below. The spice analysis was done at 100°C with a V_{CC} of 4.75V. These are worst case conditions considering the rising clock edge skew is the most critical. The motherboard impedance was varied from 50Ω to 90Ω.

Rising Edge Clock Skew Results/Recommendation

Clocks	Max. Skew Seen at the Receiver	Measured at
HCLK to HCLK	1.0 ns	1.5V to 1.5V(5)
CPU HCLK to PSC HCLKIN	1.0 ns	1.5V to 1.5V(5)
PCICLK to PCICLK	2.0 ns	1.5V to 1.5V(5)
PSC PCICLKIN to PSC HCLKIN	0.5 ns	1.5V to 2.5V(5)
HCLKOUT to PCICLK	2.0 ns	1.5V to 1.5V(5)

NOTES:
1. The skew values in the above table include an IB intrinsic skew between clock outputs of 0.5 ns.
2. To achieve the 0.5 ns clock skew shown for the CPU HCLKOUT to PSC HCLKOUT, it is required that the CPU's HCLKOUT and PSC's HCLKOUT use the same HCLKOUT from the IB.
3. The above skews were determined using a CPU load of 5 pF–15 pF. If an Intel486 SX is used, add 0.5 ns to the skew measurements for HCLKOUT to HCLKOUT and CPU HCLKOUT to PSC HCLKOUT. This is necessary because the clock capacitive loading for the Intel486 SX varies from 5 pF–20 pF.
4. Assuming that the series resistors are equal and loads are equal, the skew will vary 0.2 ns for every 1" difference in trace length seen between the clocks. Assuming that the series resistors are equal and the trace lengths are equal, the skew will vary 0.2 ns for every 5 pF difference in loading seen between the clocks. The skew is measured at the receiver.
5. All clock skew measurements were made from the 1.5V to 1.5V level on the rising edge of the clocks, with the exception of the PCICLK and HCLK inputs to the PSC. The skew between the HCLK input to the PSC and the PCICLK input to the PSC is measured from 1.5V on the rising edge of PCICLKIN to 2.5V on the rising edge of HCLKIN, as measured at the PSC. This skew must not exceed 0.5 ns.

Table 22. Clock Configurations

Strapping Options		CLK2IN	HCLKOUT	HCLKOUT Divisor	PCICLK	PCICLK Divisor	SYSCLK	SYSCLK Divisor
SIDLE#	CMDV#							
0	0	50 MHz	25 MHz	2	25 MHz	2	8.33 MHz	3
0	1	66 MHz	33 MHz	2	33 MHz	2	8.25 MHz	4
1	0	Reserved						
1	1	Reserved						

RECOMMENDED OPTIONS

In the following recommended option the series resisters shown in each option are required to improve the relative skew between clocks, and to limit the amount of ringing and undershoot seen on the lines. The undershoot was limited to approximately 0.5V to 0.8V, worst case voltage and temperature.

All the traces are labeled with a possible trace length ±0.3″ and a relative comparison of lengths using "A" ±0.3″ as shown in Figure 17 where "A" equals a route length of about 7.8″ (chosen by the PCB layout engineer). "A" should be kept between 5″ and 10″.

5.0 DESIGN CONSIDERATIONS

Design considerations are chip set related issues which affect all 82420EX PCIset designs. See your Intel representative and the 82420EX PCIset Value Flexible Motherboard Design Guide (297460) for the latest version of the design considerations.

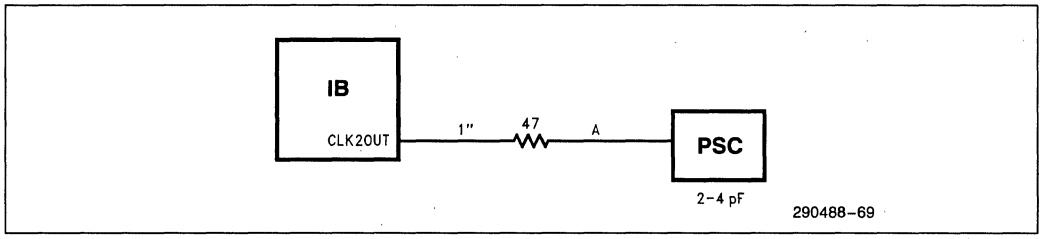


Figure 17. CLK2OUT OPTION

The following option includes two HCLKOUT loads + the IB.

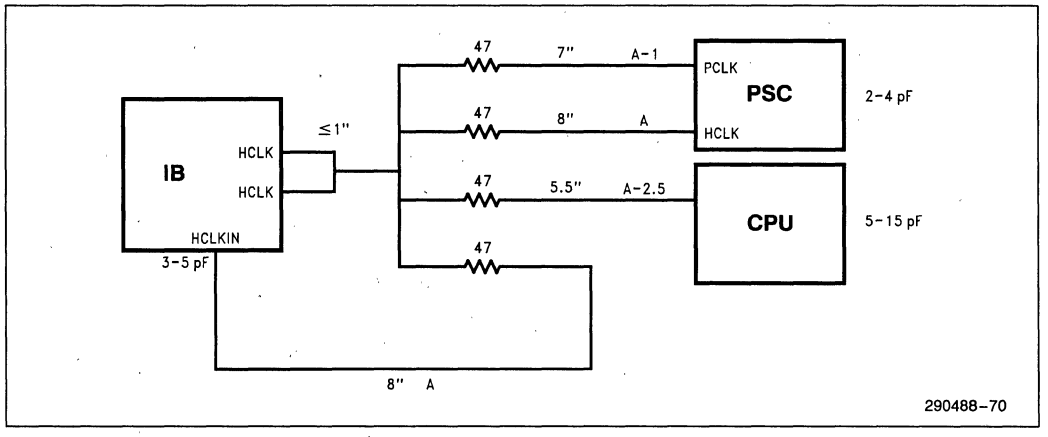


Figure 18. HCLKOUT OPTION

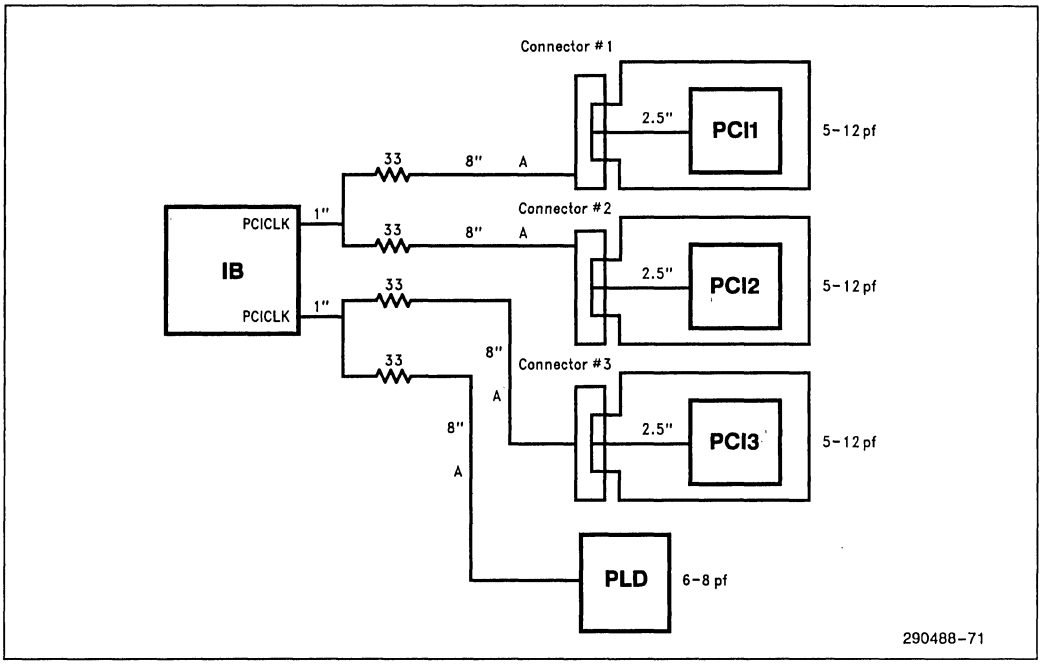


Figure 19. PCICLK OPTION

6.0 ELECTRICAL CHARACTERISTICS

This section provides the 82420EX PCIset maximum ratings, DC characteristics and AC characteristics including timing diagrams.

NOTICE: This data sheet contains information on products in the sampling and initial production phases of development. 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.*

6.1 Maximum Ratings

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

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

Supply Voltages
with Respect to Ground . . . −0.5V to V_{CC} + 0.5V

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

Power Consumption (IB) 0.5W

Power Consumption (PSC) 1.0W

6.2 PSC and IB DC Characteristics

DC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C)

Symbol	Parameter	Min	Max	Units	Test Conditions	Notes
V_{IL1}	Input Low Voltage	−0.5	0.8	V		
V_{IH1}	Input High Voltage	2.0		V		
V_{IL2}	Input Low Voltage		0.3*V_{CC}	V		10
V_{IH2}	Input High Voltage	0.7*V_{CC}		V		10
V_{T+}	TTL Schmitt Trigger, Rising Threshhold	1.9		V	V_{CC} = 5 V	11
V_{T-}	TTL Schmitt Trigger, Falling Threshhold		1.3	V	V_{CC} = 5 V	11
V_{T+}	Hysteresis Voltage	600		mV	V_{CC} = 5 V	11
V_{OL1}	Output Low Voltage (IB)		0.45	V	I_{OL} = 24 mA	1
V_{OH1}	Output High Voltage (IB)	2.4		V	I_{OH} = −3.0 mA	1
V_{OL2}	Output Low Voltage (IB)		0.4	V	I_{OL} = 4 mA	2
V_{OH2}	Output High Voltage (IB)	2.4		V	I_{OH} = −2 mA	2
V_{OL3}	Output Low Voltage (IB)		0.4	V	I_{OL} = 8 mA	3
V_{OH3}	Output High Voltage (IB)	2.4		V	I_{OH} = −2 mA	3
V_{OL4}	Output Low Voltage (PSC)		0.45	V	I_{OL} = 8 mA	4
V_{OH4}	Output High Voltage (PSC)	2.4		V	I_{OH} = −2 mA	4
V_{OL5}	Output Low Voltage (PSC)		0.45	V	I_{OL} = 4 mA	5
V_{OH5}	Output High Voltage (PSC)	2.4		V	I_{OH} = −2.0 mA	5
V_{OL6}	Output Low Voltage (PSC)		0.4	V	I_{OL} = 4 mA	6
V_{OH6}	Output High Voltage (PSC)	2.4		V	I_{OH} = −4 mA	6
V_{OL7}	Output Low Voltage		0.4	V	I_{OL} = 6 mA	7
V_{OH7}	Output High Voltage	2.4		V	I_{OH} = −2.0 mA	7
V_{OL8}	Output Low Voltage		0.4	V	I_{OL} = 4 mA	8
V_{OH8}	Output High Voltage	0.9*V_{CC}		V	I_{OH} = −2 mA	8
V_{OL9}	Output Low Voltage		0.4	V	I_{OL} = 8 mA	9
V_{OH9}	Output High Voltage	0.9*V_{CC}		V	I_{OH} = −2mA	9
I_{LI1}	Input Leakage Current		±10	A	0V<V_{IN}<V_{CC}	
I_{LI2}	Input Leakage Current		−350	A	0V<V_{IN}<V_{CC}	12
I_{LO}	Output Leakage		±10	A	0.45<V_{IN}<V_{CC}	

DC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C) (Continued)

Symbol	Parameter	Min	Max	Units	Test Conditions	Notes
C_{IN}	Capacitance Input		9	pF	@ 1 MHz	
C_{OUT}	Capacitance Output or I/O		9	pF	@ 1 MHz	
I_{CC}	V_{CC} Supply Current (IB)		100	mA		
I_{CC}	V_{CC} Supply Current (PSC)		200	mA		

NOTES:

1. V_{OL1}, V_{OH1} = SD[15:0], SA[19:0], LA[23:17], SBHE#, MEMR#, MEMW#, AEN, SPKR, BALE, SYSCLK, IOR#, IOW#, SMEMR#, SMEMW#, RSTDRV, REFRESH#, IOCHRDY, MEMCS16#, TC, DACK#
2. V_{OL2}, V_{OH2} = XBUSTR#, XBUSOE#, IGNNE#, RTCCS#, KBCCS#, BIOSCS#, RTCALE, INTR, NMI, CMDV#, SIDLE#, LREQ#, SMI#, STPCLK#
3. V_{OL3}, V_{OH3} = A[17:2], HCLKOUT1, HCLKOUT2, PCICLK1, PCICLK2, CPURST, PCIRST#
4. V_{OL4}, V_{OH4} = A[31,26:2], HD[31:0], HDP[3:0], BE[3:0]#, W/R#, RDY#, BRDY#, CI3E, CI3O2, CWE[1:0]#, COE[1:0]#, MA[10:0], RAS[4:0]#, CAS[7:0]#, WE#, LBIDE#
5. V_{OL5} V_{OH5} = HOLD, AHOLD, EADS#, KEN#, TWE#, TAG[8:0], AD[31:0], C/BE[3:0]#, PAR, CMDV#, SIDLE#, LGNT#, PGNT[1:0]
6. V_{OL6}, V_{OH6} = SRESET/INIT
7. V_{OL7}, V_{OH7} = FRAME#, IRDY#, TRDY#, STOP#, SERR#, DEVSEL#
8. V_{OL8}, V_{OH8} = CMDV#, SIDLE#, LREQ#, LGNT#
9. V_{OL9}, V_{OH9} = CLK2OUT, PCICLK1, PCICLK2, HCLKOUT1, HCLKOUT2
10. V_{IL2}, V_{IH2} = HCLKIN (IB), CMDV#, SIDLE#, LGNT#
11. This applies to PWROK, EXTSMI#, FERR#, IRQ[1,3-7,9-11,14,15],IRQ12M, PIRQ[1:0].
12. This applies to pins that include a weak internal pull-up (IRQ8#[IB], FERR#[IB], D/C#[PSC], ADS#[PSC], BLAST#[PSC], HITM#[PSC], SMI#[PSC], SMIACT#[PSC]).

6.3 IB AC Characteristics

This section provides the AC parameters and timing diagrams.

6.3.1 CLOCK/RESET TIMINGS

AC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C)

Symbol	Parameter	Min	Max	Units	Notes	Fig
t1a	HCLKIN Period Stability		0.1%			
t1b	HCLKIN Period	30.0	40.0	ns		18
t1c	HCLKIN High/Low Time	8.0		ns		18
t1d	HCLKIN Rise/Fall Time		2.0	ns		18
t1e	HCLKOUT[2,1] Period	30.0	40.0	ns		18
t1f	HCLKOUT[2,1] High/Low Time	12.0		ns		18
t1g	HCLKOUT[2,1] Rise/Fall Time		2.0	ns		18
t1h	CLK2IN Period Stability		0.1%			
t1i	CLK2IN Period	15	20.0	ns		18
t1j	CLK2IN High/Low Time	4.0		ns		18
t1k	CLK2IN Rise/Fall Time		1.5	ns		18
t1l	CLK2OUT Period	15	20	ns		18
t1o	PCICLK[2,1] Period	30	40	ns		18
t1p	PCICLK[2,1] High/Low Time	12.0		ns		18
t1q	PCICLK[2,1] Rise/Fall Time		3.0	ns		18
t1r	OSC Period	67	70	ns		18
t1s	OSC High/Low Time	20		ns		18
	ISA CLOCK TIMINGS					
	SYSCLK					
t1t	Period	120	125	ns		18
t1u	High/Low time	56		ns		18
t1v	Rise/Fall time		4	ns		18
	MISCELLANEOUS CLOCK TIMINGS					
t1w	PCICLK to HCLKOUT Skew		0.4	ns	1	19
	RESET TIMINGS					
t1x	CPURST, PCIRST#, RSTDRV Driven Inactive After PWROK is Driven Active High.	2		HCLKIN		20
t1y	CPURST, PCIRST#, RSTDRV Active Pulse Width. Initiated via the TRC Register.	1		ms		21
t1z	CPURST Valid Delay from HCLKIN Rising	3	17	ns		

NOTES:
1. Except when STPCLK# is active.

6.3.2 PSC/IB LINK INTERFACE TIMINGS

AC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C)

Symbol	Parameter	Min	Max	Units	Notes	Fig
t2a	CMDV#, SIDLE# Setup to HCLKIN Rising	11		ns		24
t2b	A[17:2] Setup to HCLKIN Rising	6		ns		24
t2c	SIDLE#, A[17:2] Hold from HCLKIN Rising	2		ns		24
t2c1	CMDV# Hold from HCLKIN Rising	2.6		ns		24
t2d	CMDV#, SIDLE#, A[17:2] Valid Delay from HCLKIN Rising	3	10	ns		25
t2e	LGNT# Setup to HCLKIN Rising	11		ns		24
t2f	LGNT# Hold from HCLKIN Rising	2		ns		24
t2g	LREQ# Valid Delay from HCLKIN Rising	2	9	ns		25
t2h	SIDLE# Driven Valid After CPURST is Driven High	2		HCLKIN		22

6.3.3 SYSTEM POWER MANAGEMENT TIMINGS

AC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C)

Symbol	Parameter	Min	Max	Units	Notes	Fig
	SMI#					
t3a	Valid Delay from HCLKIN	2	9	ns		25
t3b	Active Pulse Width	3		HCLKIN		23
t3c	Inactive Pulse Width	4		HCLKIN		23
	EXTSMI#					
t3d	Active Pulse Width	2		HCLKIN		23
t3e	Inactive Pulse Width	4		HCLKIN		23
t3f	Valid Setup to HCLKIN	6		ns		24
t3g	Valid Hold from HCLKIN	2		ns		24
	STPCLK#					
t3h	Valid Delay from HCLKIN	2	10	ns		25
t3i	STPCLK# Inactive Pulse Width	5		HCLKIN		23

6.3.4 ISA BUS AND X-BUS TIMINGS

ISA Bus and X-Bus AC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C)

Symbol	Parameter	Min	Max	Units	Type	Size	Notes	Fig
	IB AS MASTER TIMINGS							
	BALE							
t4a	BALE Pulse Width	52		ns	M,I/O	8,16		26,27,28,29
t4b	BALE Driven Active from MEMx#, IOx# Inactive	44		ns	M,I/O	8,16		26,27,28,29
	LA[23:17]							
t5a	LA[23:17] Valid Setup to BALE Inactive	150		ns	M	8,16	7	26,27
t5b	LA[23:17] Valid Hold from BALE Inactive	26		ns	M	8,16		26,27
t5c	LA[23:17] Valid Setup to MEMx# Active	150		ns	M	16		27
t5d	LA[23:17] Valid Setup to MEMx# Active	173		ns	M	8		26
t5e	LA[23:17] Invalid from MEMx# Active	39		ns	M	16		27
t5f	LA[23:17] Invalid from MEMx# Active	39		ns	M	8		26
	SA[19:0], SBHE#							
t6a	SA[19:0], SBHE# Valid Setup to MEMx# Active	34		ns	M	16	13	27
t6b	SA[19:0], SBHE# Valid Setup to IOx# Active	100		ns	I/O	16		29
t6c	SA[19:0], SBHE# Setup to MEMx#, IOx# Active	100		ns	M,I/O	8		26,28
t6d	SA[19:0], SBHE# Valid Setup to BALE Inactive	37		ns	M,I/O	8,16	13	26,27,28,29
t6e	SA[19:0], SBHE# Valid Hold from MEMx#, IOx# Inactive	41		ns	M,I/O	8,16		26,27,28,29

ISA Bus and X-Bus AC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C) (Continued)

Symbol	Parameter	Min	Max	Units	Type	Size	Notes	Fig
	MEMR#, MEMW#, IOR# and IOW#							
t7a	MEMx# Active Pulse Width (std)	225		ns	M	16		27
t7b	IOx# Active Pulse Width (std)	160		ns	I/O	16		29
t7c	MEMx# Active Pulse Width (nws)	105		ns	M	16	1	27
t7d	MEMx# or IOx# Active Pulse Width (std)	520		ns	M,I/O	8		26,28
t7e	MEMx# or IOx# Active Pulse Width (nws)	160		ns	M,I/O	8	1	26,28
t7f	MEMx# Inactive Pulse Width	103		ns	M	16		27
t7g	MEMx# Inactive Pulse Width	163		ns	M	8		26
t7h	IOx# Inactive Pulse Width	163		ns	I/O	8,16		28,29
t7i	MEMx#, IOx# Driven Inactive from IOCHRDY Active	120		ns	M,I/O	8,16		26,27,28,29
	SMEMR# and SMEMW#							
t8a	SMEMR# & SMEMW# Propagation Delay from MEMR# and MEMW#		5	ns	M	8,16		26,27
	Read Data							
t9a	Read Data Driven from MEMR#, IOR# Active	0		ns	M,I/O	8,16		26,28,29
t9b	Read Data Valid Setup to MEMR#, IOR#	20		ns	M,I/O	8,16		26,27,28
t9c	Read Data Valid Hold from MEMR#, IOR# Inactive	0		ns	M,I/O	8,16		26,27,28,29
t9d	Read Data Tri-stated from MEMR# and IOR# Inactive		41	ns	M,I/O	8,16		26,27,28,29
	Write Data							
t10a	Write Data Valid Setup to MEMW#, IOW# Active	30		ns	M,I/O	8,16		26,27,28,29
t10b	Write Data Valid Hold from MEMW#, IOW# Inactive	45		ns	M,I/O	8,16		26,27,28,29
t10c	Write Data Tri-Stated from MEMW#, IOW# Inactive		75	ns	M,I/O	8,16		26,27,28,29
t10d	Write Data Driven Valid after Read MEMR#, IOR# Inactive	41		ns	M,I/O	8,16		26,27,28,29

ISA Bus and X-Bus AC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C) (Continued)

Symbol	Parameter	Min	Max	Units	Type	Size	Notes	Fig
	MEMCS16#							
t11a	MEMCS16# Driven Active from LA[23:17] Valid		94	ns	M	16		27
t11b	MEMCS16# Inactive from LA[23:17] Valid		91	ns	M	8		26,27
t11c	MEMCS16# Valid Hold from LA[23:17] Invalid	0		ns	M	16		27
t11d	MEMCS16# Driven Active from SA[19:2] Valid		35	ns	M	16		27
	IOCS16#							
t12a	IOCS16# Driven Active from Valid SA[19:0]		123	ns	I/O	16		29
t12b	IOCS16# Inactive from Valid SA[19:0]		91	ns	I/O	8		28,29
t12c	IOCS16# Valid Hold from SA[19:0] Invalid	0		ns	I/O	16		29
t12d	IOCS16# Driven Active from IOx Active		75	ns	I/O	16		29
	ZEROWS#							
t13a	ZEROWS# Driven Active from MEMx# Active		27	ns	M	16		27
t13b	ZEROWS# Driven Active from MEMx#, IOx# Active		80	ns	M,I/O	8		26,28
t13c	ZEROWS# Driven Active from LA[23:17] Valid		180	ns	M	16		27
t13d	ZEROWS# Driven Active from LA[23:17] Valid		300	ns	M	8		26
t13e	ZEROWS# Driven Active from SA[19:0], SBHE# Valid		80	ns	M	16		27
t13f	ZEROWS# Driven Active from SA[19:0], SBHE# Valid		200	ns	M,I/O	8		26,28
	AEN							
t14a	AEN Valid Setup to IOx# Driven Active	111		ns	I/O	8,16		28,29
t14b	AEN Valid Setup to BALE Driven Inactive	111		ns	I/O	8,16		28,29
t14c	AEN Valid Hold from IOx# Driven Inactive	41		ns	I/O	8,16		28,29
	IOCHRDY							
t15a	IOCHRDY Driven Valid from MEMx#, IOx# Active		78	ns	M,I/O	16		27,29
t15b	IOCHRDY Driven Valid from MEMx#, IOx# Active		366	ns	M,I/O	8		26,28
t15e	IOCHRDY Inactive Pulse Width	120	15.6	ns	M,I/O	8,16		26,28,29

ISA Bus and X-Bus AC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C) (Continued)

Symbol	Parameter	Min	Max	Units	Type	Size	Notes	Fig
	IB AS SLAVE TIMINGS							
	LA[23:17]							
t16a	LA[23:17] Valid Setup to MEMx# Active	23		ns	M	16		30
t16b	LA[23:17] Invalid from MEMx# Active	28		ns	M	16		30
	SA[19:0],SBHE#							
t17a	SA[19:0],SBHE# Setup to MEMx# Active	23		ns	M	16		30
t17b	SA[19:0],SBHE# Setup to IOx# Active	89		ns	I/O	8		31
t17c	SA[19:0],SBHE# Valid Hold from MEMx#, IOx# Inactive	30		ns	M,I/O	8,16		30,31
	MEMR#, MEMW#, IOR#, IOW#							
t18a	MEMx# Active Pulse Width	214		ns	M	16		30
t18b	IOx# Active Pulse Width	509		ns	I/O	8		31
t18c	MEMx# Inactive Pulse Width	92		ns	M	16		30
t18d	IOx# Inactive Pulse Width	152		ns	I/O	8		31
	Read Data							
t19a	Read Data Valid from IOCHRDY Active		69	ns	M,I/O	8,16		30,31
t19b	Read Data Valid from IOR# Active		69	ns	I/O	8	11	31
t19c	Read Data Valid Hold from MEMR#, IOR# Inactive	0		ns	M,I/O	8,16		30,31
t19d	Read Data Tri-State from MEMR#, IOR# Inactive		30	ns	M,I/O	8,16		30,31
	Write Data							
t20a	Write Data Valid Setup to MEMW#, IOW# Active	−54		ns	M,I/O	8,16		30,31
t20b	Write Data Valid Hold from MEMW#, IOW# Inactive	14		ns	M,I/O	8,16		30,31
	MEMCS16#							
t21a	MEMCS16# Driven Active from Valid LA[23:17]		65	ns	M	16		30
t21b	MEMCS16# Float from Valid LA[23:17]		31	ns	M	16		30
t21c	MEMCS16# Valid Hold from LA[23:17] Invalid	0		ns	M	16		30
	IOCHRDY							
t22a	IOCHRDY Inactive from MEMx#, IOx# Active		25	ns	M,I/O	8,16		30,31
t22b	IOCHRDY Float from IOCHRDY Rising		70	ns	M,I/O	8,16	4	30,31
t22c	IOCHRDY Inactive Pulse Width	120	2.5	μs	M,I/O	8,16		30,31

ISA Bus and X-Bus AC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C) (Continued)

Symbol	Parameter	Min	Max	Units	Type	Size	Notes	Fig
	INTERRUPT AND NMI TIMINGS							
	NMI Timing							
t23a	SERR#, IOCHK# Active to NMI Driven Active		200	ns				32
	Interrupt Timing							
t24a	IRQ Inactive Pulse Width	100		ns				33
	ISA BUS MASTER TIMINGS							
	DACK#							
t26a	DACK# Inactive from DREQ Inactive	240		ns				34
	Tri-Stating and Driving the Bus							
t27a	IB Tri-States Address, Data, and Control Signals from DACK# Active		30	ns				34
t27b	IB Drives Address, Data, and Control Signals from DACK# Inactive	71		ns				34
	SMEMR# and SMEMW#							
t28a	SMEMR# & SMEMW# Valid from MEMR# and MEMW# Valid		20	ns				34
	DATA SWAP LOGIC TIMING (ISA Master to ISA Slave)							
t29a	SD[7:0] to SD[15:8] Propagation Delay		15	ns				35
t29b	SD[15:8] to SD[7:0] Propagation Delay		15	ns				35
t29c	IB Drives Data Bus from IOR#, IOW#, MEMR# or MEMW# Active		20	ns			2	35
t29d	IB Tri-States Bus from IOR#, MEMR#, or SMEMR# Inactive	5	20	ns			2,3	35
t29e	IB Tri-States Bus from IOW#, MEMW#, or SMEMW# Inactive	15	60	ns			2,3	35
	DMA COMPATIBLE TIMINGS							
	DREQ							
t30a	DREQ Active Hold from IOR# Active		558	ns			5	37
t30b	DREQ Active Hold from IOW# Active		315	ns			5	36
	DACK#							
t31a	DACK# Active to IOR# Active	73		ns				37
t31b	DACK# Active to IOW# Active	312		ns				36
t31c	DACK# Active Hold from IOR# Inactive	105		ns				37
t31d	DACK# Active Hold from IOW# Inactive	161		ns				36

ISA Bus and X-Bus AC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C) (Continued)

Symbol	Parameter	Min	Max	Units	Type	Size	Notes	Fig
	AEN and BALE							
t32a	AEN Active to IOx# Active	111		ns				36,37
t32b	AEN and BALE Inactive from IOx# Inactive	41		ns				36,37
	LA[23:19], SA[19:0], SBHE#							
t33a	LA[23:19],SA[19:0], SBHE# Valid Setup to MEMx# Active	99		ns				36,37
t33b	LA[23:19],SA[19:0], SBHE# Valid Hold from MEMx# Inactive	51		ns				36,37
	MEMR#, MEMW#, IOR#, IOW#							
t34a	IOW# and MEMW# Active Pulse Width	474		ns				36,37
t34b	MEMR# Active Pulse Width	520		ns				36
t34c	IOR# Active Pulse Width	769		ns				37
t34d	IOW# Inactive Pulse Width (continuous)	469		ns				36
t34e	IOR# Inactive Pulse Width (continuous)	167		ns				37
t34f	IOR# Active to MEMW# Active	235		ns				37
t34g	MEMR# Active to IOW# Active	0		ns				36
t34h	MEMR# Active Hold from IOW# Inactive	50		ns				36
t34i	IOR# Active Hold from MEMW# Inactive	50		ns				37
t34j	MEMx# Active Hold from IOCHRDY Active	120		ns				36,37
	SMEMR# and SMEMW#							
t35a	SMEMR# and SMEMW# Valid from MEMR# and MEMW# Valid		5	ns				36,37
	Read Data							
t36a	Read Data Valid from IOR# Active		237	ns				37
t36b	Read Data Valid Hold from IOR# Inactive	0		ns				37
t36c	Read Data Float from IOR# Inactive		61	ns				37
	Write Data							
t37a	Write Data Valid Setup to IOW# Inactive	252		ns				36
t37b	Write Data Valid Hold from IOW# Inactive	36		ns				36

ISA Bus and X-Bus AC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C) (Continued)

Symbol	Parameter	Min	Max	Units	Type	Size	Notes	Fig
	DATA SWAP LOGIC TIMING (ISA to ISA Transaction)							
t38a	SD[7:0] to SD[15:8] Propagation Delay		15	ns				38
t38b	SD[15:8] to SD[7:0] Propagation Delay		15	ns				38
t38c	IB Drives Data Bus from IOR# or MEMR# Active		20	ns			2	38
t38d	IB Tri-States Bus from IOR# or MEMR# Inactive		20	ns			2	38
	TC							
t39a	TC Active Setup to IOx# Inactive	511		ns			6	36,37
t39b	TC Active Hold from IOx# Inactive	71		ns			6	36,37
t39h	TC Pulse Width	700		ns				36,37
	IOCHRDY							
t40b	IOCHRDY Valid from MEMx# Active		315	ns				36,37
t40c	IOCHRDY Inactive Pulse Width	125		ns				36,37
	ISA REFRESH TIMINGS							
	REFRESH#							
t62a	REFRESH# Active Setup to MEMR# Active	120		ns				39,40
t62b	REFRESH# Active Hold from MEMR# Inactive	31	218	ns				8,40
t62c	REFRESH# Driven Active to SA[15:0] Valid	11		ns				8,40
t62d	REFRESH# Active Hold from SA[15:0] Invalid	11		ns				8,40
	AEN							
t63a	AEN Driven Active to MEMR# Active	11		ns				39,40
t63b	AEN Hold from MEMR# Inactive	11		ns				39,40
	SA[15:0]							
t64a	SA[15:0] Valid Setup to MEMR# Active	81		ns				39,40
t64b	SA[15:0] Valid Hold from MEMR# Inactive	36		ns				39,40
t64c	SA[15:0] Valid Float from MEMR# Inactive	45	120	ns			8	40

ISA Bus and X-Bus AC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C) (Continued)

Symbol	Parameter	Min	Max	Units	Type	Size	Notes	Fig
	MEMR#, SMEMR#							
t65a	MEMR# Active Pulse Width	225		ns				39,40
t65b	MEMR# Tri-state from MEMR# Inactive	45	120	ns				39,40
t65c	MEMR# Driven Inactive from IOCHRDY Active	120		ns				39,40
t65d	SMEMR# Propagation Delay from MEMR#		5	ns				39
	IOCHRDY							
t66a	IOCHRDY Inactive from MEMR# Active		76	ns				39,40
t66b	IOCHRDY Valid from MEMR# Active		76	ns				39,40
t66c	IOCHRDY Active to Inactive	120		ns				39,40
	IB Driving Bus from REFRESH#							
t67a	IB Drives Control and Address from REFRESH# Active	5		ns			8	40
	IB AND ISA MASTER ACCESSES TO THE X-BUS							
	BIOSCS#, KBCCS#, and RTCCS#							
t68a	CS# Driven Active from SA[19:0], LA[23:17] Valid		25	ns				41
t68b	CS# Driven Inactive from SA[16:0], LA[23:17] Invalid		25	ns				41
	XBUSTR# and XBUSOE#							
t69a	XBUSTR# Active from IOR#, MEMR# Active		17	ns				41
t69b	XBUSOE# Active from IOx#, MEMx# Active		29	ns				41
t69c	XBUSTR# Active Setup to XBUSOE# Active	3	12	ns				41
t69d	XBUSOE# Inactive from IOx#, MEMx# Inactive	35	60	ns			9	41
t69e	XBUSTR# Inactive from IOR#, MEMR# Inactive	45	100	ns			9	41
t69f	XBUSOE# Setup to XBUSTR# Inactive	10	45	ns			9	41
t69g	XBUSOE# Inactive from SA[16:0] and LA[23:17]		25	ns			10	41
t69h	XBUSTR# Inactive from IOR#, MEMR# Inactive	15	60	ns			10	41

ISA Bus and X-Bus AC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C) (Continued)

Symbol	Parameter	Min	Max	Units	Type	Size	Notes	Fig
	DMA ACCESSES TO X-BUS							
	XBUSTR#							
t70a	XBUSTR# Active from DACKx# Active		25	ns			12,14	42
t70b	XBUSTR# Inactive from DACKx# Inactive	10	65	ns			12	42
	MiSCELLANEOUS X-BUS TIMINGS							
	Mouse Timing Support							
t71a	IRQ12/M and IRQ1 Minimum Active Pulse Width (for Mouse Function and Keyboard)	180		ns				43
	Coprocessor Error Support							
t73a	IGNNE# Active from IOW# Active from Port F0h Access		220	ns				43
t73b	IGNNE# Inactive from FERR# Inactive		150	ns				43
	Real Time Clock Timing (RTCALE)							
t75a	RTCALE Pulse Width	200	300	ns				44
t75b	RTCALE Active from IOW# Active		70	ns				44
	Speaker Timing							
t76a	SPKR Valid Delay from OSC Rising		200	ns				45

NOTES:
1. No-wait-state (ZEROWS#) asserted.
2. This applies to the byte lane that the data has been swapped to.
3. Data is tri-stated from the standard memory commands (SMEMR# or SMEMW#), when they are generated.
4. This specification includes both the time the IB drives IOCHRDY active and the time it takes the IB to float IOCHRDY.
5. This applies to the last cycle of a demand mode DMA transfer.
6. Output from IB.
7. 36 ns has been added to the ISA spec to meet ZEROWS# setup requirements.
8. This applies to ISA Master initiated refresh only.
9. IB as a master cycles only.
10. ISA master cycles only.
11. This applies to the IB cycles that IOCHRDY is not driven low.
12. This applies to all DACK# signals.
13. 56 ns has been added to the ISA spec to meet MEMCS16# setup requirements. ISA devices are not suppose to use the SA address as part of their MEMCS16# decode. However, some devices do use SA as part of MEMCS16# decode.
14. X-Bus read
15. For back-to-back "sub cycles" generated as a result of byte assembly or disassembly, this spec is 34 ns.

6.3.5 AC TEST LOADS

Table 23. AC Test Loads

Capacitive Load	Signals
200 pF	REFRESH#, TC, SD[15:0], SA[19:0], SBHE#, LA[23:17], I0CS16#, MASTER#, MEMCS16#, MEMR#, MEMW#, SMEMR#, SMEMW#, IOR#, IOW#, AEN, BALE, IOCHRDY, ZEROWS#, RSTDRV, SYSCLK
120 pF	DACK#[7:5,3:0]
50 pF	SPKR, INTR, NMI, BIOSCS#, KBCCS#, RTCCS#, RTCALE, XBUSTR#, XBUSOE#, IGNNE#

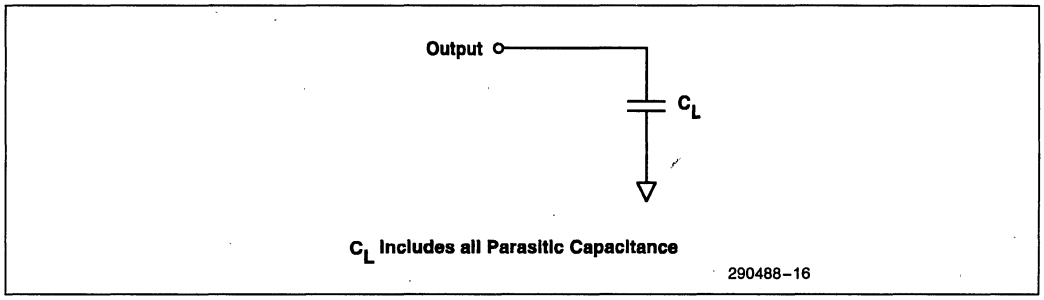


Figure 20. Test Load

6.3.6 AC TIMING WAVEFORMS

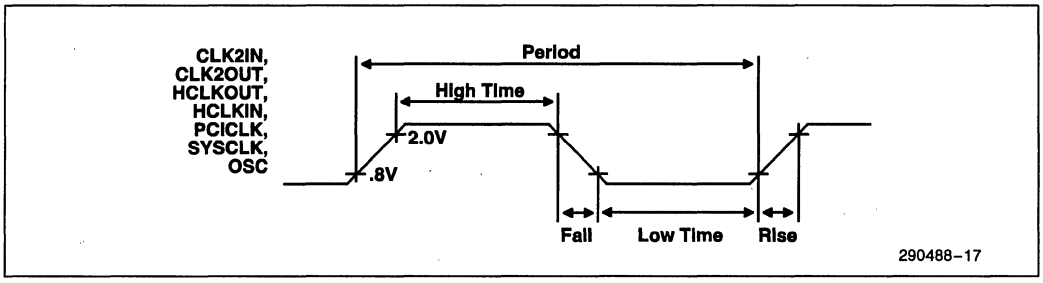


Figure 21. Clock Timing

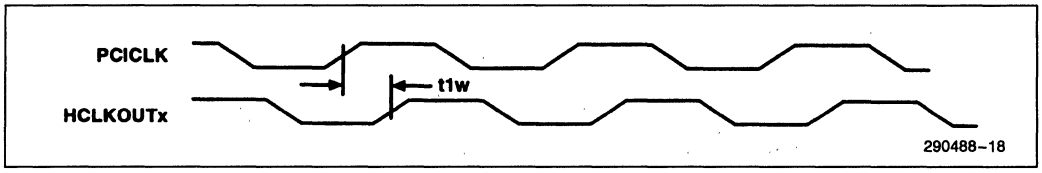


Figure 22. PCICLK-to-HCLKOUT Skew Timing

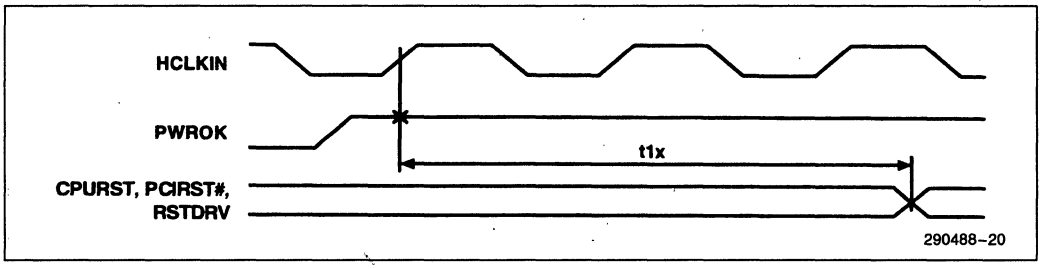


Figure 23. Reset Inactive after PWROK

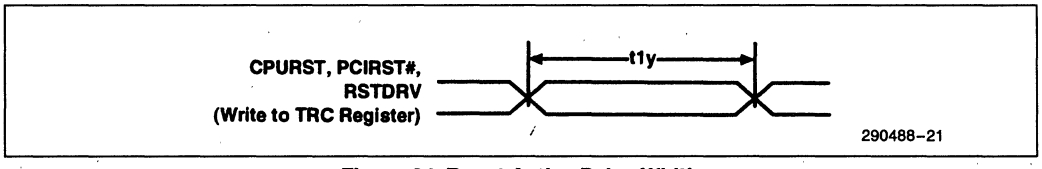


Figure 24. Reset Active Pulse Width

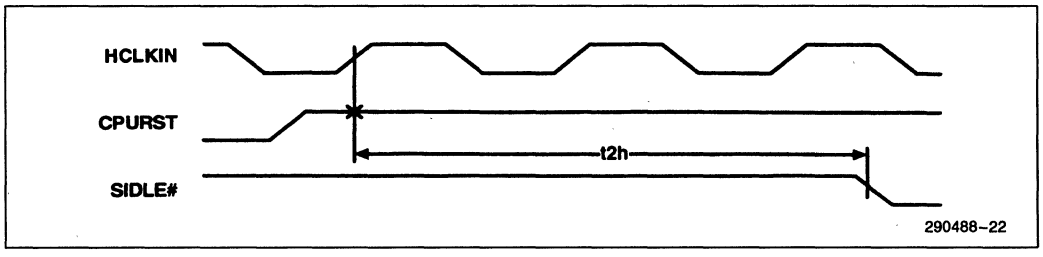


Figure 25. SIDLE# Active after CPURST

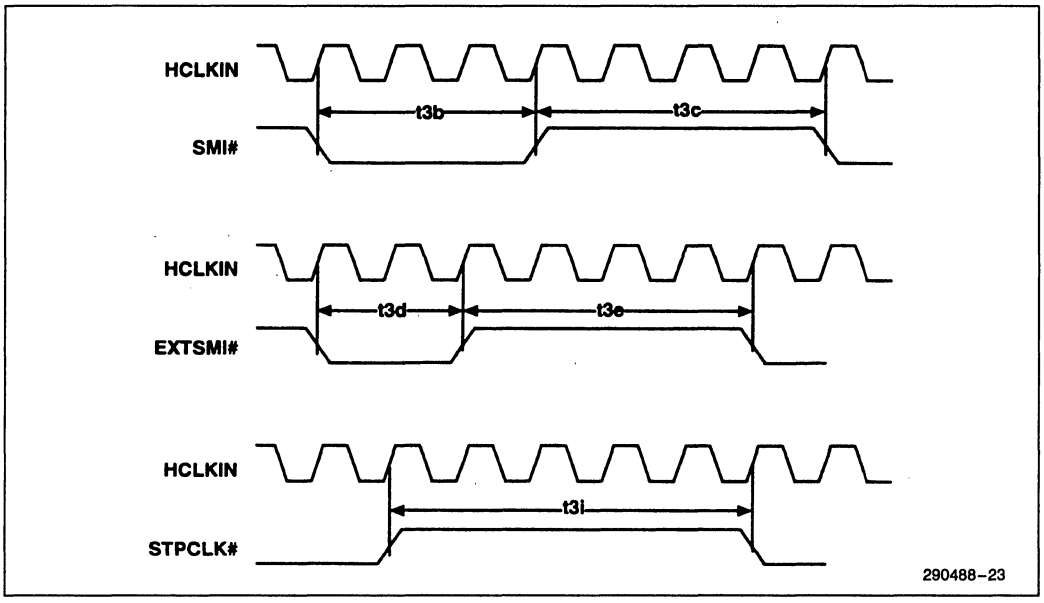


Figure 26. SMI#, EXTSMI#, and STPCLK# Timing

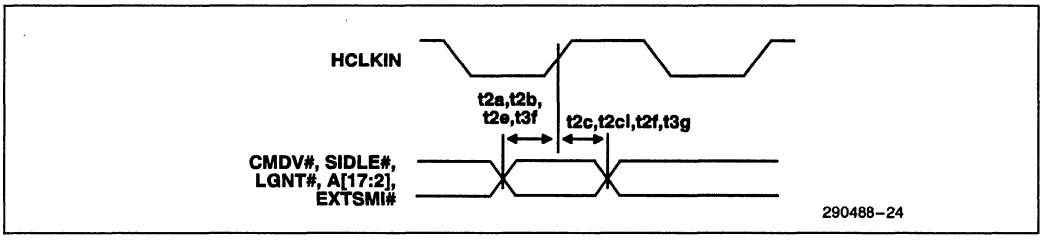


Figure 27. Input to HCLKIN Setup/Hold Times

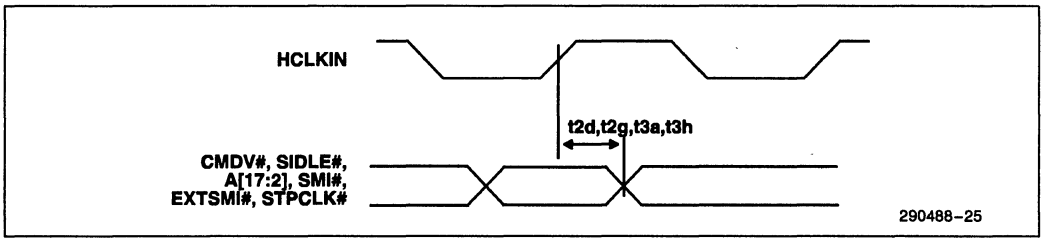


Figure 28. HCLKIN to Output Valid Delay

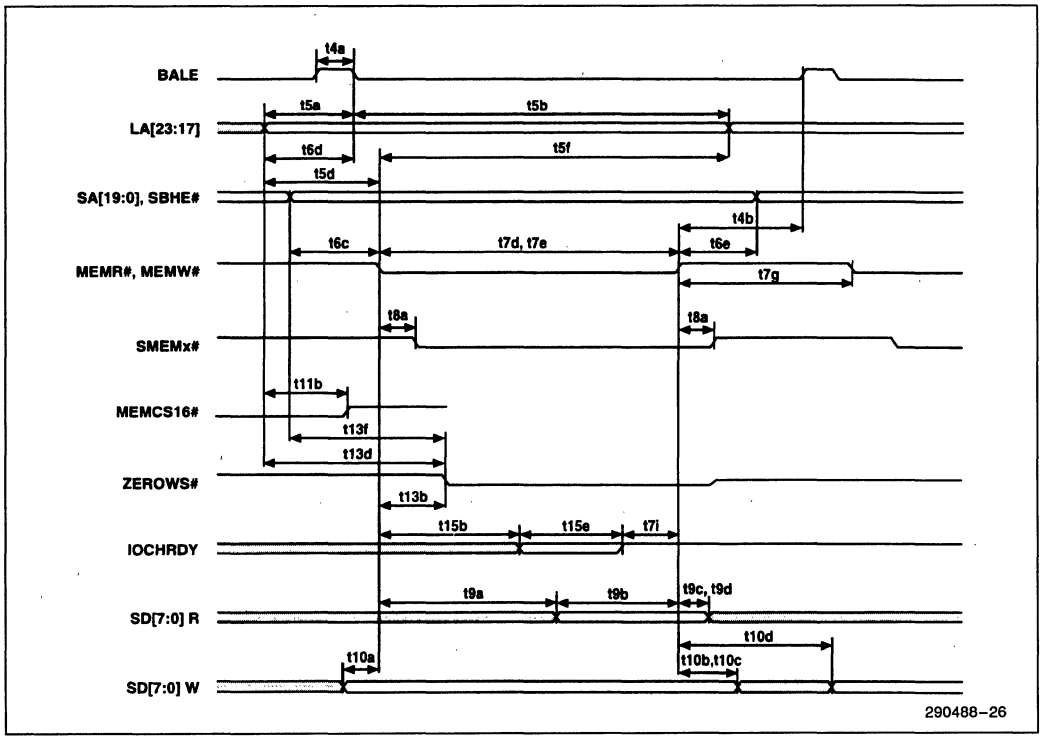


Figure 29. 8-Bit ISA Memory Slave Timing (IB as Master)

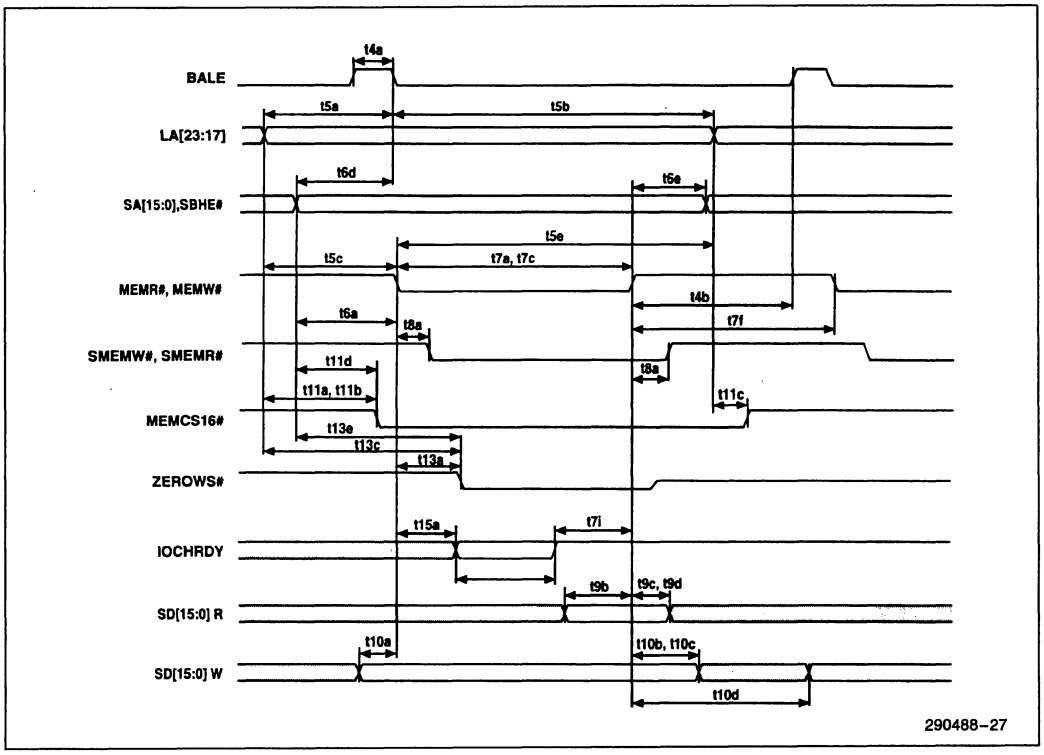


Figure 30. 16-Bit ISA Memory Slave Timing (IB as Master)

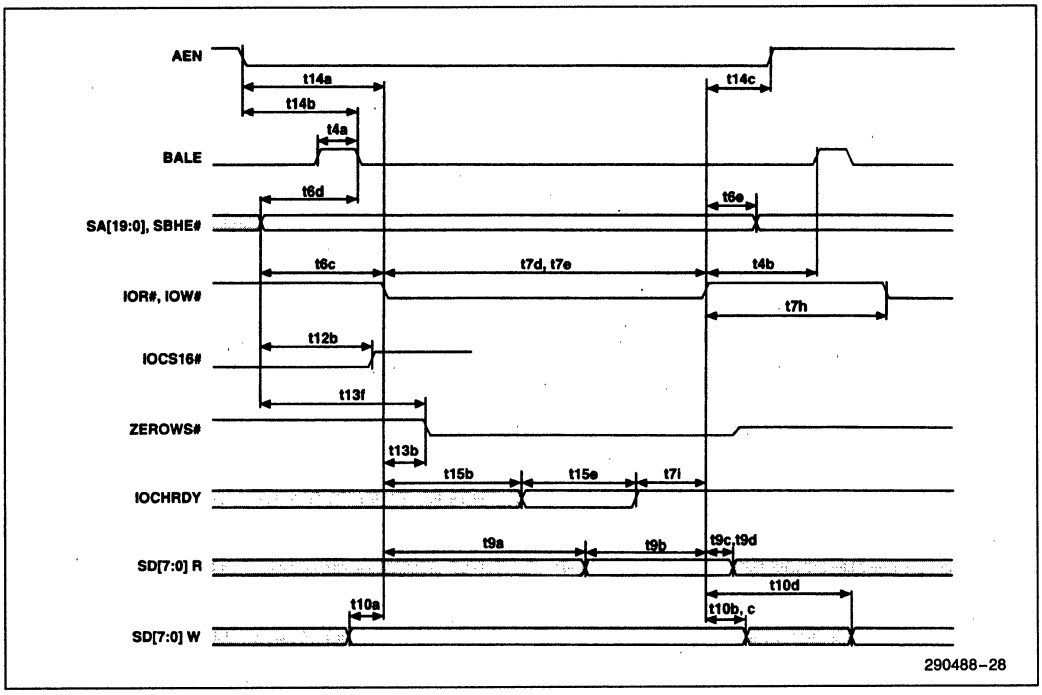


Figure 31. 8-Bit ISA I/O Slave Timing (IB as Master)

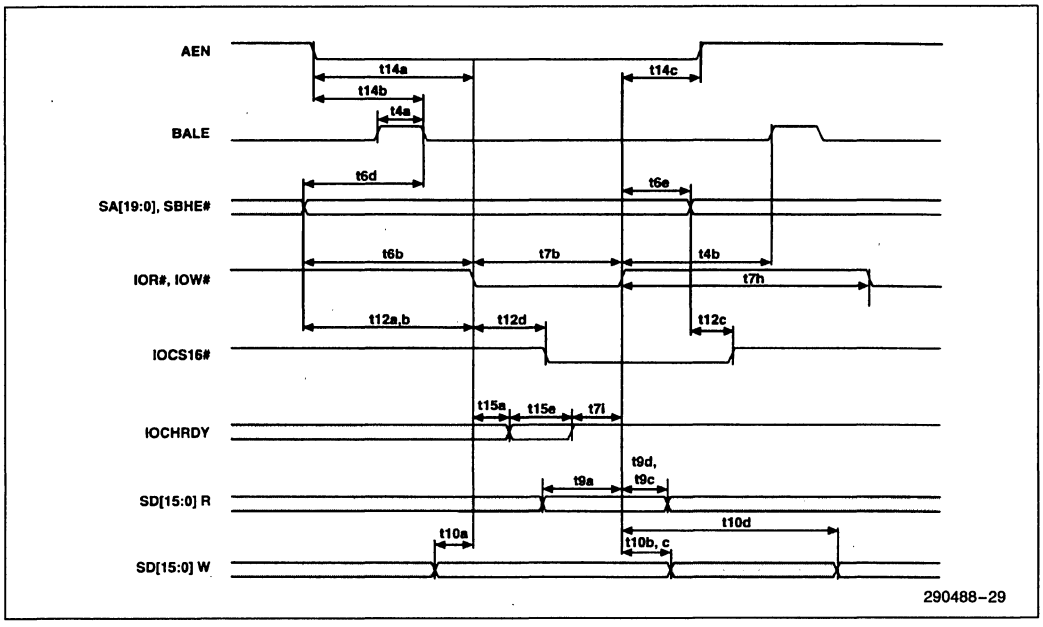


Figure 32. 16-Bit I/O Slave Timing (IB as Master)

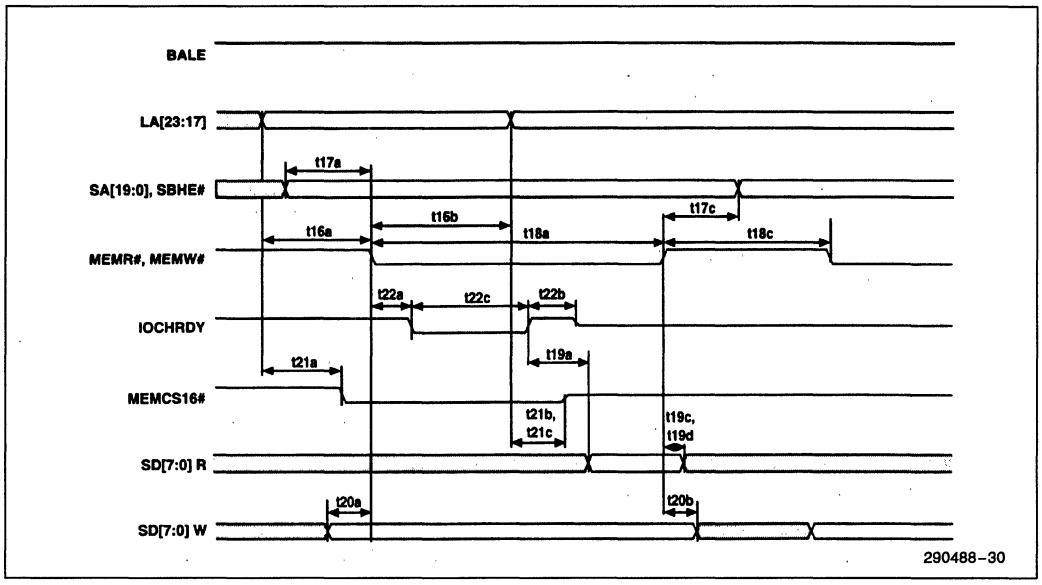


Figure 33. ISA Master Accessing PCI Memory Timing

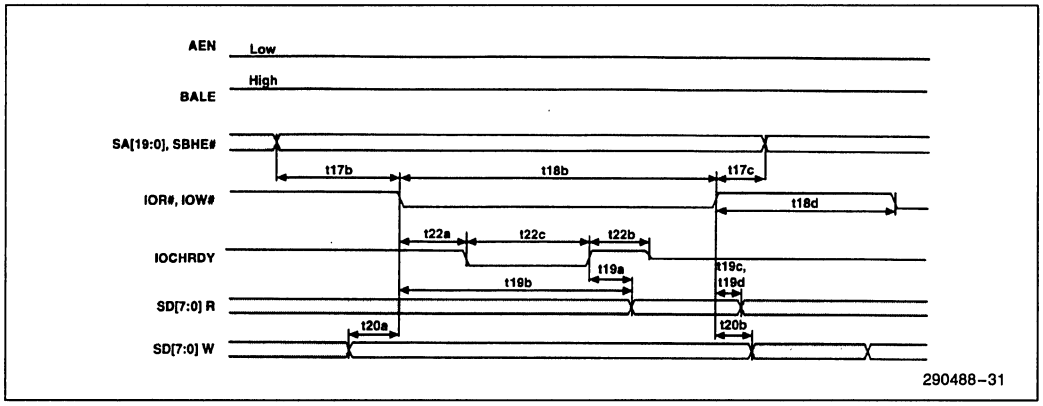


Figure 34. ISA Master Accessing IB Register Timing

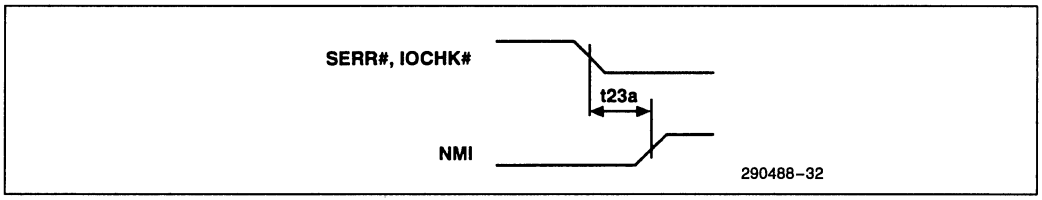


Figure 35. NMI Timing

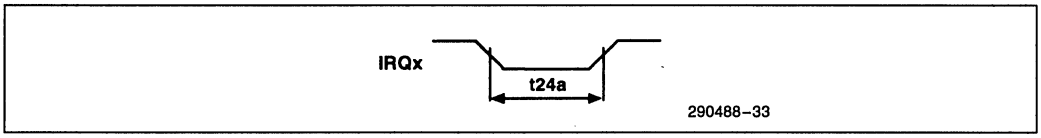


Figure 36. Interrupt Timing

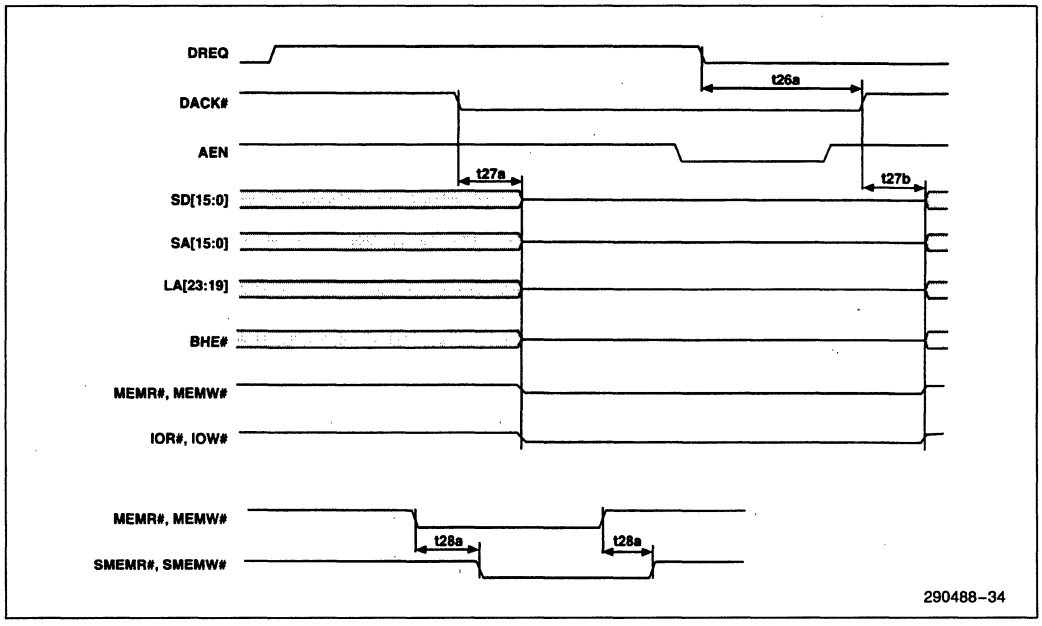


Figure 37. ISA Master Miscellaneous Timing

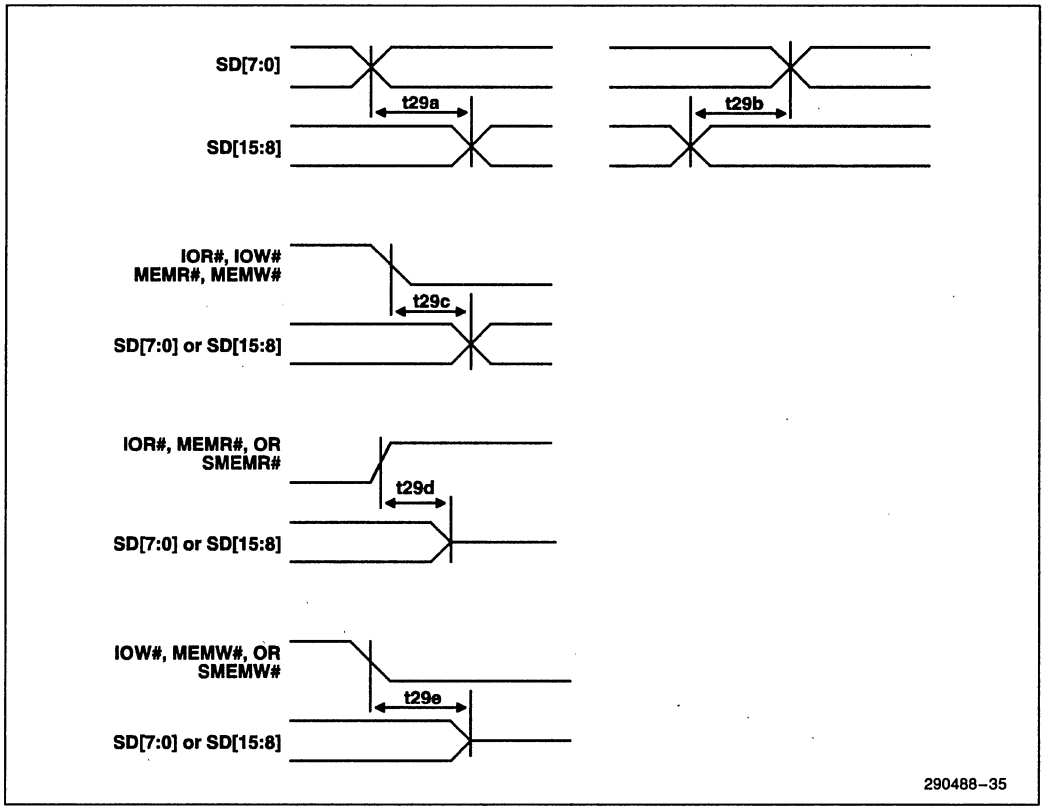


Figure 38. ISA Master Data Swap Timing

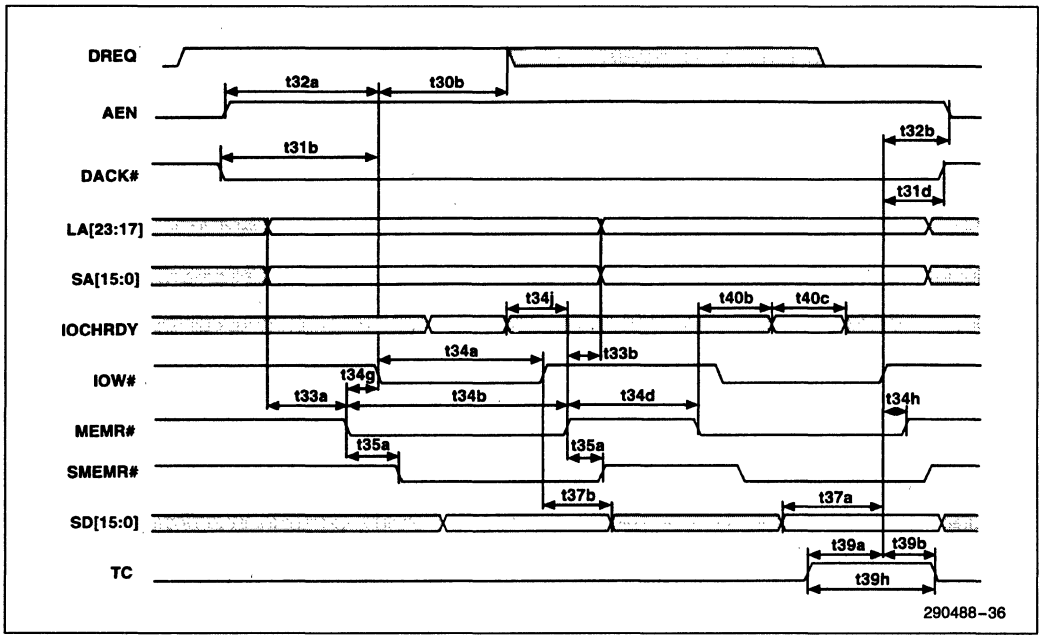


Figure 39. DMA Compatible Timing (Memory Read)

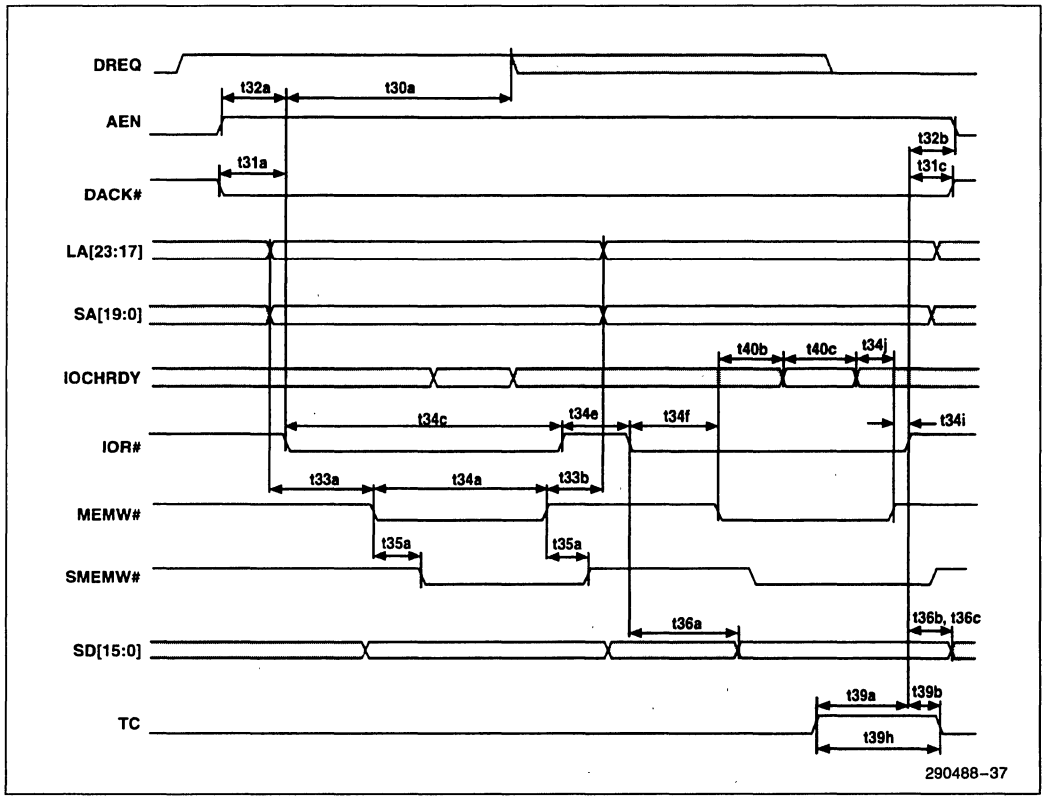


Figure 40. DMA Compatible Timing (Memory Write)

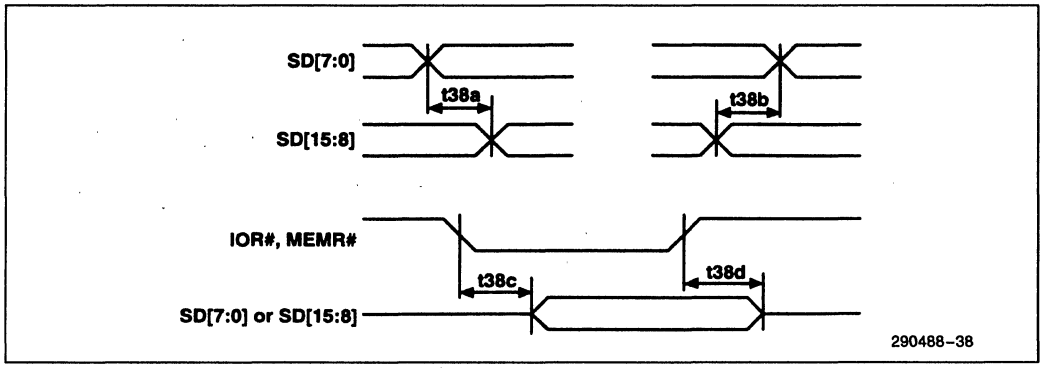


Figure 41. DMA Compatible Timing (Data Swap)

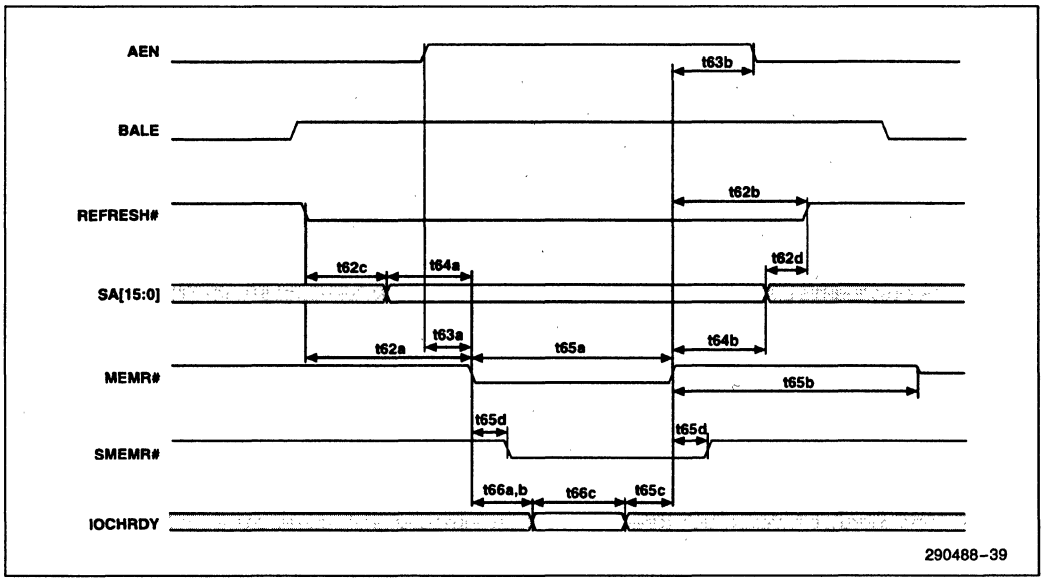


Figure 42. IB-Initiated Refresh Timing

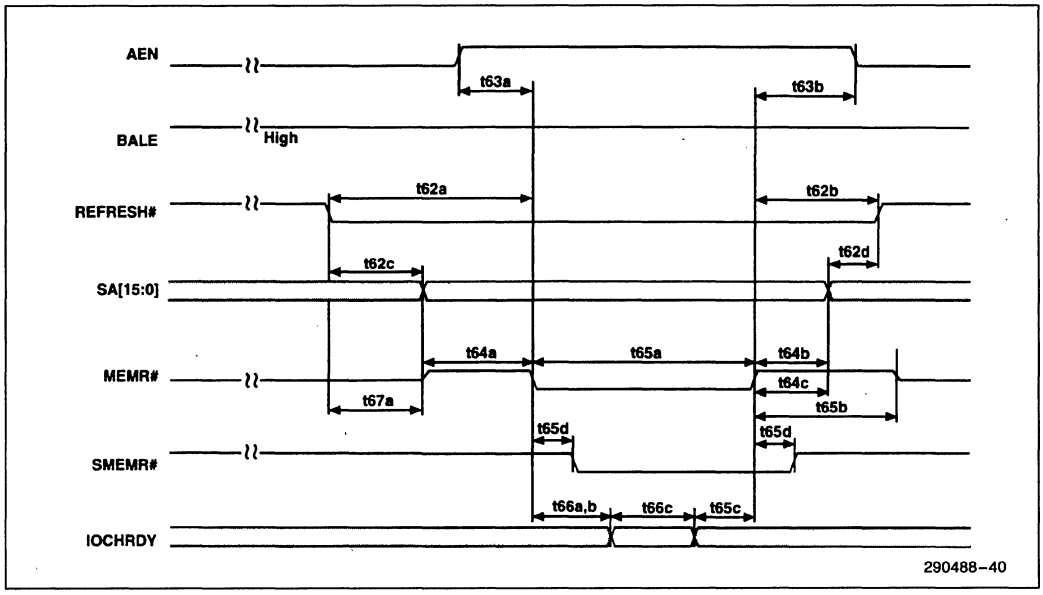


Figure 43. ISA Master-Initiated Refresh Timing

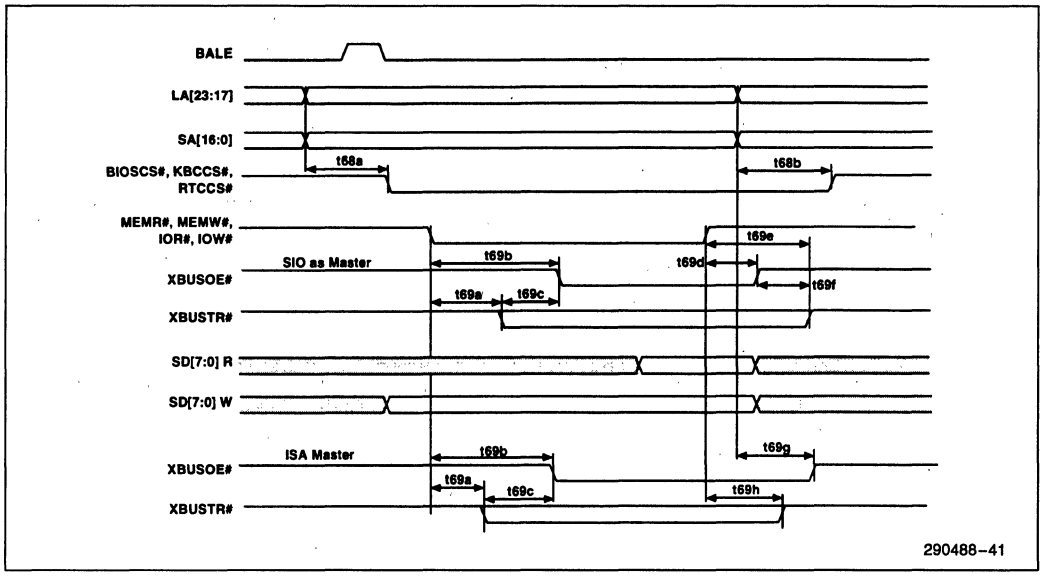


Figure 44. IB and ISA Master Access to X-Bus Timing

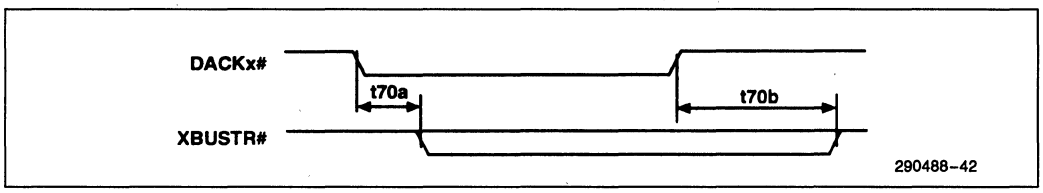


Figure 45. DMA Access to X-Bus Timing

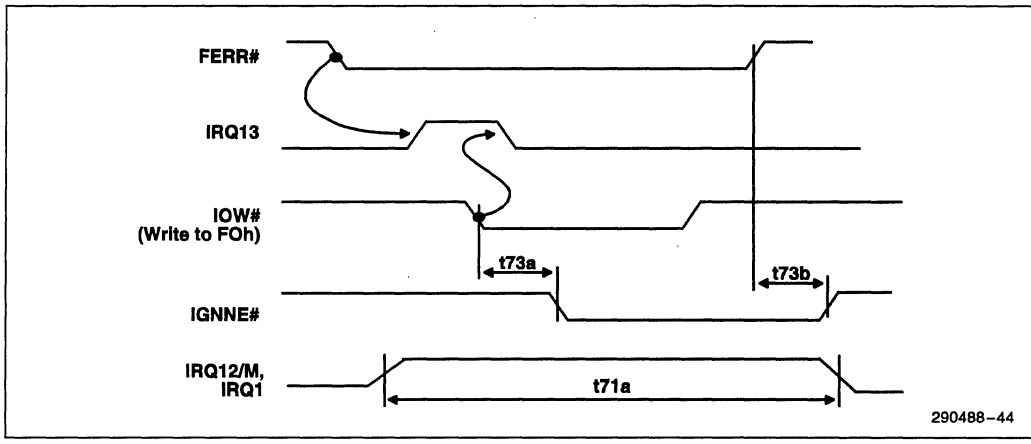


Figure 46. Coprocessor Error and Mouse Support Timing

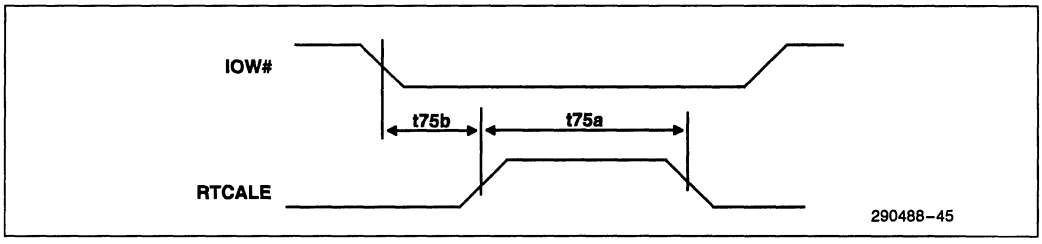


Figure 47. Real Time Clock Timing (RTCALE Generation)

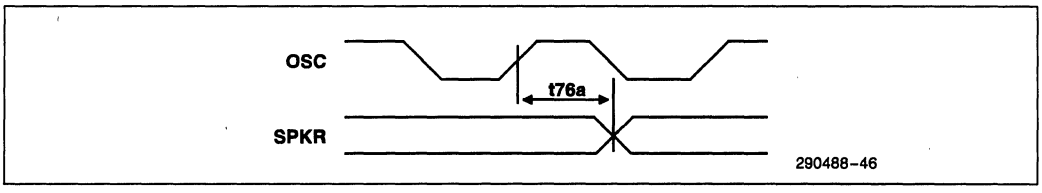


Figure 48. Speaker Timing

6.4 PSC AC Characteristics

This section provides the AC parameters and timing diagrams for the 82425EX PSC.

6.4.1 HOST CLOCK TIMING

AC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C)

Symbol	Parameter	Min	Max	Units	Notes	Fig
t1a	HCLKIN Period Stability		0.1%			
t1b	HCLKIN Period	30.0	40.0	ns		49
t1c	HCLKIN High Time	10.0		ns		49
t1d	HCLKIN Low Time	10.0		ns		49
t1e	HCLKIN Rise Time		2.0	ns		49
t1f	HCLKIN Fall Time		2.0	ns		49
t2a	CLK2IN Period Stability		0.1%			
t2b	CLK2IN Period	15	20.0	ns	Note 1	47

6.4.2 CPU INTERFACE TIMINGS

AC Characteristics (V_{DD} = 5V ±5%, T_{CASE} = 0 to 85 °C)

Symbol	Parameter	Min	Max	Units	Notes	Fig
t10a	HITM#, PCD, HLDA, BE[3:0], SMIACT#, SMI# Setup Time to HCLKIN Rising	10.0		ns		51
t10a1	ADS#, BLAST# Setup Time to HCLKIN Rising	12.0		ns		51
t10b	W/R#, M/IO#, D/C# Setup Time to HCLKIN Rising	14.0		ns		51
t10c	ADS#, HITM#, W/R#, M/IO#, D/C#, PCD, HLDA, BLAST#, BE[3:0]#, SMIACT#, SMI#, HLDA Hold Time from HCLKIN Rising	2.0		ns		51
t11a	HD[31:0], HDP[3:0] Setup Time to HCLKIN Rising	10.0		ns		51
t11b	HD[31:0], HDP[3:0] Hold Time from HCLKIN Rising	2.0		ns		51
t11c	HD[31:0], HDP[3:0] Output Enable from HCLKIN Rising	0.0	12.0	ns		54

AC Characteristics (V_{DD} = 5V ±5%, T_{CASE} = 0 to 85 °C) (Continued)

Symbol	Parameter	Min	Max	Units	Notes	Fig
t11d	HD[31:0], HDP[3:0] Valid Delay from HCLK Rising	3.0	14.0	ns		51
t11e	HD[31:0], HDP[3:0] Float Delay from HCLKIN Rising	0.0	12.0	ns		53
t12a	A[31,26:2] Setup Time to HCLKIN Rising	12.0		ns		51
t12b	A[31,26:2] Hold Time from HCLKIN Rising	2.0		ns		51
t12c	A[31,26:2] Output Enable from HCLKIN Rising	0.0	14.0	ns		54
t12d	A[31,26:2], BE[3:0]#, W/R# Valid Delay from HCLKIN Rising	3.0	14.0	ns		51
t12e	A[31,26:2], BE[3:0]#, W/R# Float Delay from HCLKIN Rising	0.0	12.0	ns		53
t13a	RDY#, BRDY# Rising Edge Valid Delay from HCLKIN Rising	3.0	16.0	ns		51
t13b	RDY#, BRDY# Falling Edge Valid Delay from HCLKIN Rising	3.0	16.0	ns		51
t14	AHOLD Valid Delay from HCLKIN Rising	3.0	11.0	ns		51
t15	EADS#, CPURST, HOLD Valid Delay from HCLKIN Rising	3.0	13.0	ns		52
t15a	KEN# Valid Delay from HCLKIN Rising	3.0	22.0	ns		51
t16a	INIT/SRESET High Pulse Width	16		HCLK		56
t16b	INIT/SRESET Valid Delay from HCLKIN Rising	3.0	13.0	ns		51
t17a	CPURST Setup Time to HCLKIN Rising	10.0		ns	Note 1	52
t17b	CPURST Hold Time to HCLKIN Rising	2.0		ns	Note 1	52
t17c	CPURST Pulse Width	3		HCLK	Note 2	56
t18a	LDEV# Setup Time to HCLKIN Rising	7.0				52
t18b	LDEV# Hold Time to HCLKIN Rising	2.0				52
t18c	LRDY# Early Setup Time to HCLKIN Rising	15.0				52
t18d	LRDY# Late Setup Time to HCLKIN Rising	7.0				52
t18e	LRDY# Hold Time to HCLKIN Rising	2.0				52
t18f	LRDY# to RDY# Propagation Delay	0.0	11.0			50

NOTES:
1. Synchronous Reset
2. Asynchronous Reset

6.4.3 SECOND LEVEL CACHE TIMING

AC Characteristics (V_{DD} = 5V ±5%, T_{CASE} = 0 to 85 °C)

Symbol	Parameter	Min	Max	Units	Notes	Fig
t20a	CI3E/CI3O2 Propagation Delay from A3	0.0	9.0	ns		57
t20b	CI3E/CI3O2 Valid Delay from HCLKIN Rising	3.0	13.0	ns		51
t21a	COE[1:0]# Falling Edge Valid Delay from HCLKIN Rising	3.0	13.0	ns		51
t21b	COE[1:0]# Rising Edge Valid Delay from HCLKIN Rising	3.0	13.0	ns		51
t22a	CWE[1:0]# Falling Edge Valid Delay from HCLKIN Rising	3.0	12.0	ns		51
t22b	CWE[1:0]# Rising Edge Valid Delay from HCLKIN Rising	3.0	13.0	ns		51
t22c	CWE[1:0]# Low Pulse Width	14.0		ns		56
t23	TWE# Valid Delay from HCLKIN Rising	3.0	14.0	ns		51
t24a	TAG[8:0] Valid Delay from HCLK	3.0	17.0	ns		
t24b	TAG[8:0] Setup Time to HCLKIN Rising	4.0		ns		52
t24c	TAG[8:0] Hold Time to HCLKIN Rising	7.0		ns		52

6.4.4 DRAM INTERFACE TIMING

AC Characteristics (V_{DD} = 5V ±5%, T_{CASE} = 0 to 85 °C)

Symbol	Parameter	Min	Max	Units	Notes	Fig
t30a	RAS[4:0]# Valid Delay from HCLKIN Rising	3.0	13.0	ns		51
t30b	RAS[4:0]# Pulse Width High	4		HCLK −5 ns		56
t31a	CAS[7:0]# Valid Delay from HCLKIN Rising	3.0	11.0	ns		51
t31b	CAS[7:0]# Pulse Width High	1		HCLK −2 ns		56
t32	WE# Valid Delay from HCLKIN Rising	3.0	17.0	ns		51
t33a	MA[10:0] Propagation Delay from A[26:5]	0.0	10.0	ns		57
t33b	MA[10:0] Row to Column Switching Delay	0.0	19.0	ns		57
t33c	MA[10:0] Valid Delay from HCLKIN Rising	3.0	15.0	ns		51

6.4.5 PCI TIMING

AC Characteristics (V_{DD} = 5V ±5%, T_{CASE} = 0 to 85 °C)

Symbol	Parameter	Min	Max	Units	Notes	Fig
	PCI Clock Timing					
t41a	Period	30.0		ns		49
t41b	PCLKIN High Time	8.0		ns		49
t41c	PCLKIN Low Time	8.0		ns		49
t41d	PCLKIN Rise Time		3.0	ns		49
t41e	PCLKIN Fall Time		3.0	ns		496
	PCI Interface Timing					
t50a	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, LOCK#, PAR, SERR#, DEVSEL# Valid Delay from PCLKIN Rising	2.0	11.0	ns		51
t50b	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, LOCK#, PAR, SERR#, DEVSEL# Output Enable Delay from PCLKIN Rising	2.0		ns		
t50c	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, LOCK#, PAR, SERR#, DEVSEL# Float Delay from PCLKIN Rising		28.0	ns		
t50d	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, LOCK#, PAR, SERR#, DEVSEL# Setup Time to PCLKIN Rising	7.0		ns		52
t50e	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, LOCK#, PAR, SERR#, DEVSEL# Hold Time from PCLKIN Rising	0.0		ns		52
t51a	AD[31:0] Setup Time to PCLKIN Rising	7.0		ns		52
t51b	AD[31:0] Hold Time to PCLKIN Rising	0.0		ns		52
t51c	AD[31:0] Valid Delay from PCICLK	2.0	11.0	ns		57
t52a	PREQ0, PREQ1/LDEV# Setup Time to PCLKIN	12.0		ns		52
t52b	PREQ0, PREQ1/LDEV# Hold Time to PCLKIN	2.0		ns		52
t52c	PGNT1, PGNT0 Valid Delay from PCLKIN	2.0	12.0	ns		51
t53a	PWRGOOD Setup Time to HCLKIN Rising	7.0		ns		52
t53b	PWRGOOD Hold Time to HCLKIN Rising	2.0		ns		52

6.4.6 PSC/IB LINK INTERFACE TIMING

AC Characteristics (V_{DD} = 5V ±5%, T_{CASE} = 0 to 85 °C)

Symbol	Parameter	Min	Max	Units	Notes	Fig
t60a	CMDV#, SIDLE# Valid Delay from HCLKIN Rising	3.0	13.0	ns		51
t60b	CMDV#, SIDLE# Setup Time to HCLKIN Rising	11.0		ns		52
t60c	CMDV#, SIDLE# Hold Time to HCLKIN Rising	1.0		ns		52
t61a	LREQ Setup Time to HCLKIN Rising	10.0		ns		52
t61b	LREQ Hold Time to HCLKIN Rising	1.0		ns		52
t62	LGNT Valid Delay from HCLKIN Rising	3.0	13.0	ns		51

6.4.7 PCI BUS IDE TIMING

AC Characteristics (V_{DD} = 5V ±5%, T_{CASE} = 0 to 85 °C)

Symbol	Parameter	Min	Max	Units	Notes	Fig
t70	LBIDE#, IDE1CS#, IDE3CS#, DIR, IOR#, IOW# Valid Delay from PCLKIN Rising	2.0	12.0	ns		51
t71a	IORDY Falling Setup Time to HCLKIN Rising	18.0		ns		52
t71a1	IORDY Rising Setup Time to HCLKIN Rising	8.0		ns		52
t71b	IORDY Hold Time to HCLKIN Rising	2.0		ns		52
t72	IDEA[2:0] Valid Delay from PCLKIN Rising	2.0	12.0	ns		51
t73a	IDED[15:0] Valid Delay from PCLKIN Rising	2.0	12.0	ns		51
t73b	IDED[15:0] Setup Time to HCLKIN Rising	8.0		ns		52
t73c	IDED[15:0] Hold Time to HCLKIN Rising	2.0		ns		52

6.4.8 AC TEST LOADS

Table 24. AC Test Loads

Capacitive Load	Pin
0 pF	RAS[3:0]#, CAS[7:0]#, WE#, MA[10:0]
0 pF	HD[31:0], HDP[3:0], A[31:30,26:2], RDY#, BRDY#, BOFF#, AHOLD, EADS#, INV, KEN#, CPURST, HOLD, A20M#, CI3E, CI3O2, COE[1:0]#, CWE[1:0]#, TWE#, TAG[8:0], CMDV#, SIDLE#, LGNT, PGNT1, PGNT0, LBIDE#, IDE1CS#, IDE3CS#, DIR, IOR#, IOW#

6.4.9 MISCELLANEOUS CLOCK TIMINGS

Symbol	Parameter	Min	Max	Units	Notes	Fig
t3a	PCICLKIN to HCLKIN Skew	0	0.5	ns	Measured at 1.5V to 2.5V	55

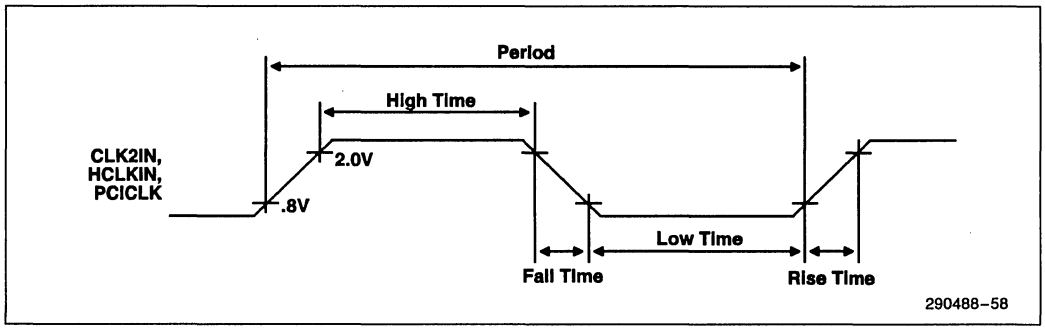


Figure 49. Clock Timing

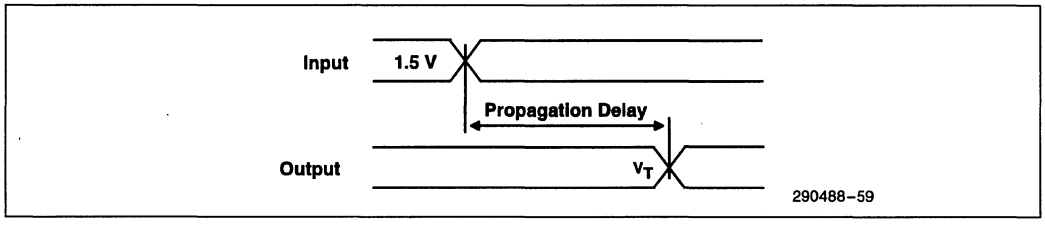


Figure 50. Propagation Delay

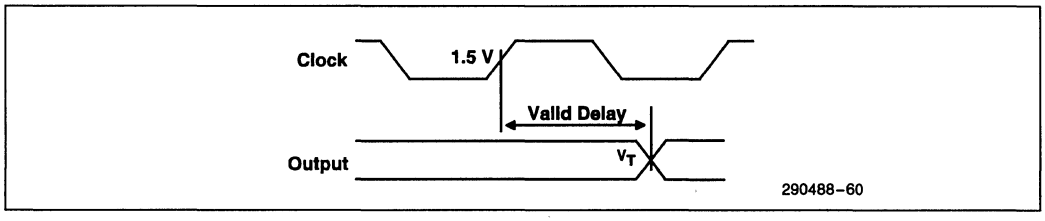


Figure 51. Valid Delay from Rising Clock Edge

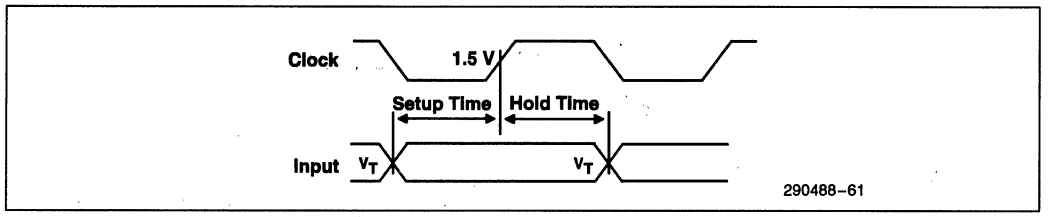


Figure 52. Setup and Hold Times

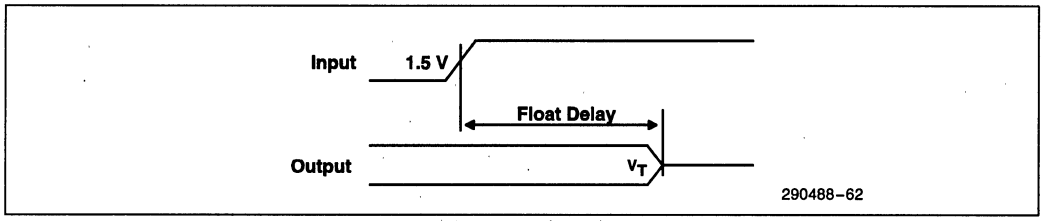


Figure 53. Float Delay

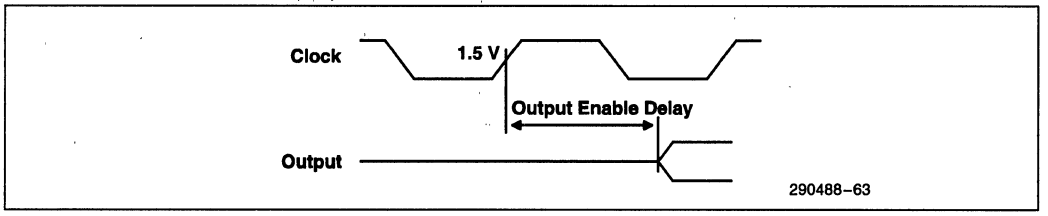
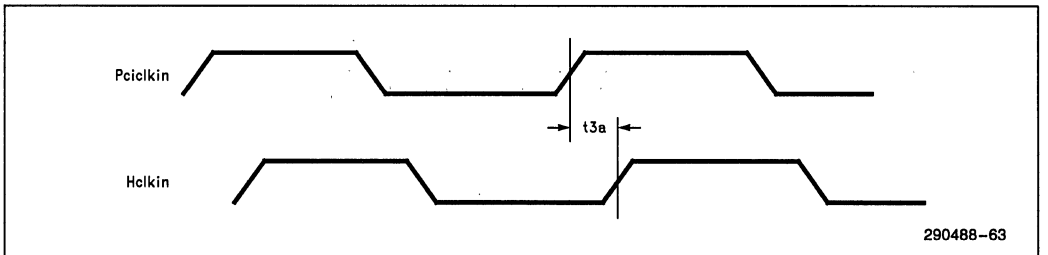


Figure 54. Output Enable Delay



NOTE:
The skew between the HCLK input to the PSC and the PCICLK input to the PSC is measured from 1.5V on the rising edge of PCICLKIN to 2.5V on the rising edge of HCLKIN, measured at the PSC.

Figure 55. PCICLKIN to HCLKIN Skew

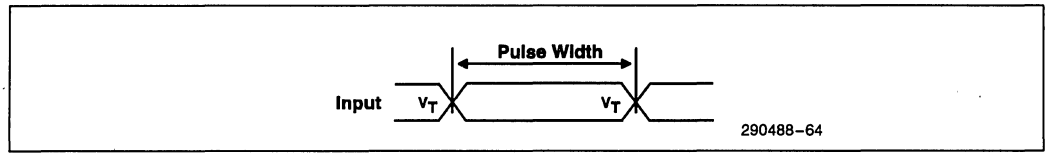


Figure 56. Pulse Width

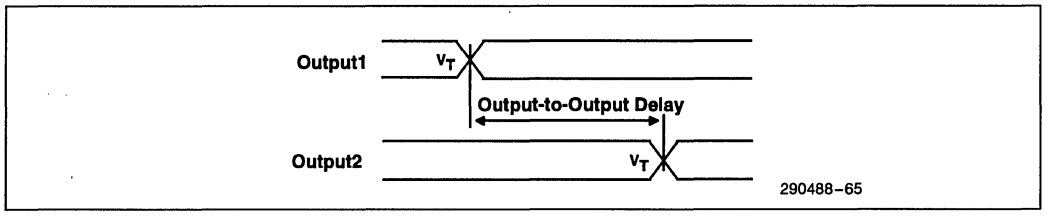


Figure 57. Output-to-Output Delay

7.0 IB AND PSC PACKAGE INFORMATION

Figure 58 shows the package information for the 82426EX IB and Figure 59 shows the package information for the 82425EX PSC.

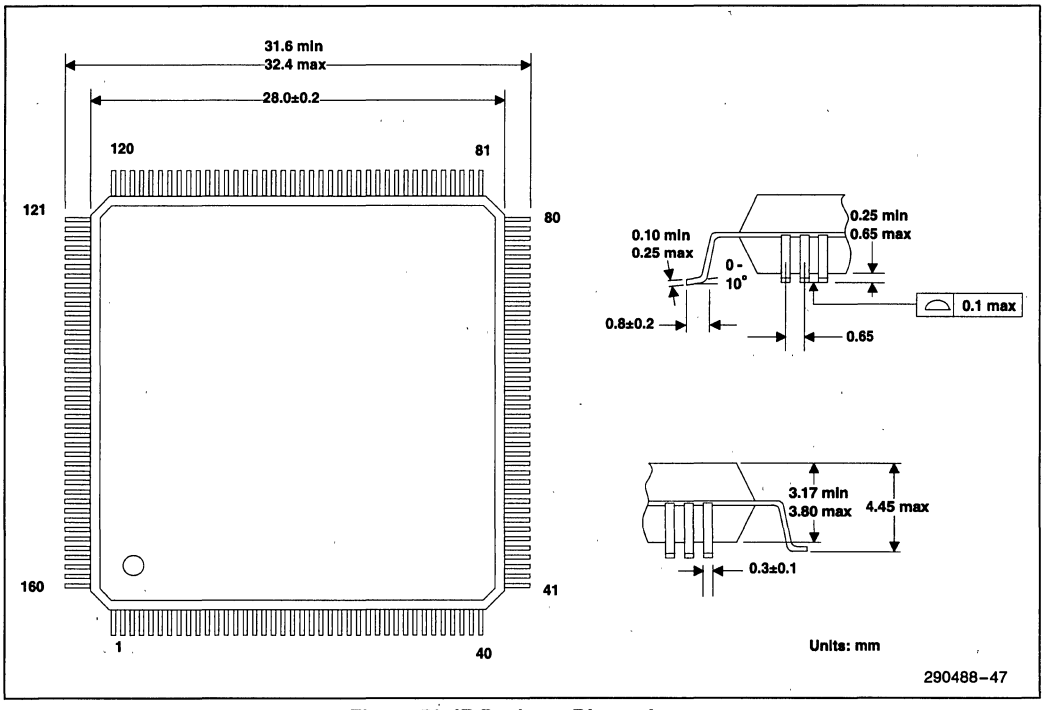


Figure 58. IB Package Dimensions

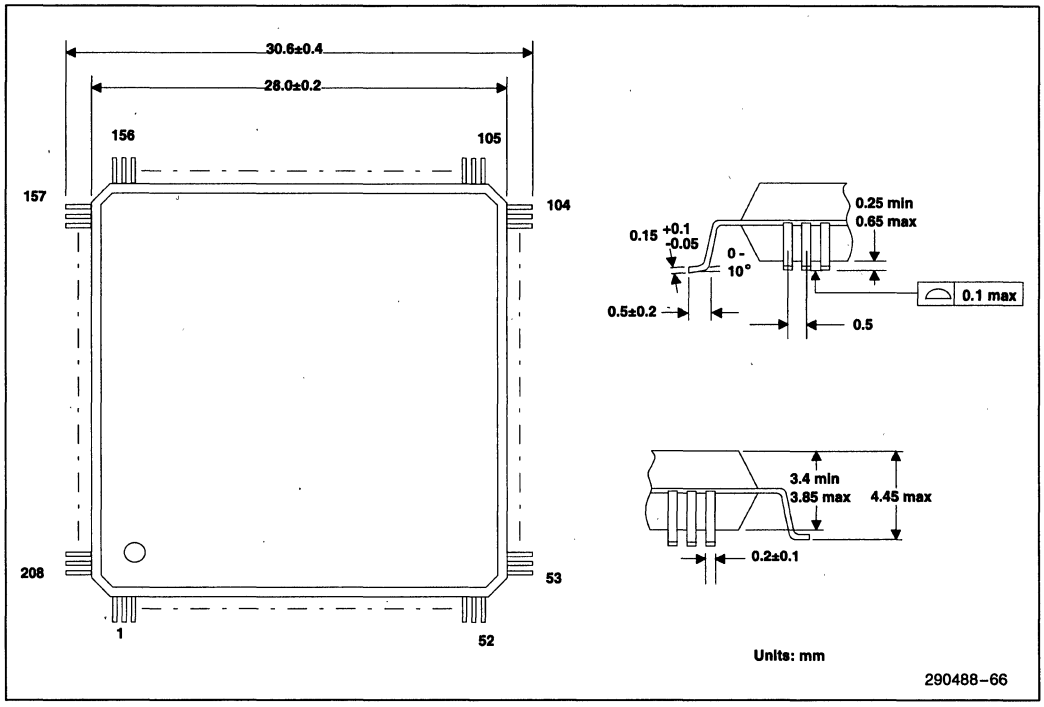


Figure 59. PSC Package Dimensions

7.1 Thermal Characteristics

The 82420EX PCIset is designed for operation at case temperatures between 0°C and 85°C. The thermal resistances of the packages are given in Table 25 and Table 26.

Table 25. PSC Package Thermal Resistance

Parameter	Air Flow (Linear Feet per Minute)		
	0	200	400
θ Junction to Case (°C/Watt)	16	16	16
θ Case to Ambient (°C/Watt)	34	27	23

Table 26. IB Package Thermal Resistance

Parameter	Air Flow (Linear Feet per Minute)		
	0	200	400
θ Junction to Case (°C/Watt)	16	16	16
θ Case to Ambient (°C/Watt)	34	27	23

8.0 TESTABILITY

8.1 PSC Testability

8.1.1 PSC TRI-STATE CONTROL

The PSC can be forced to tri-state all of its output drivers. The LOCK# must be connected to Vcc through a pull-up resistor for normal operation. The PSC will latch the values of LOCK# on the falling edge of CPURST. If these signals have been driven to a logic "0", the PSC will tri-state all of its drivers on the next rising edge of HCLKIN. The PSC will continue to tri-state all drivers until the rising edge of HCKLIN after LOCK# is forced to a logic "1".

8.1.2 PSC NAND TREE

A NAND Tree is provided in the PSC for Automated Test Equipment (ATE) board level testing. The NAND tree allows the tester to set the connectivity of each of the PSC signal pins. While in NAND tree mode, all PSC drivers except A30 are tri-stated. The NAND tree output is driven on pin A30.

NAND tree mode is entered similar to tri-state mode. During CPURST, PREQ0#, like LOCK#, is driven low. NAND tree mode is entered on the rising edge of HCLKIN after CPURST goes inactive.

Table 27 shows the sequence of the NAND tree in the PSC. Non-inverting inputs are driven directly into the input of a NAND gate in the NAND tree.

Table 27. PSC NAND Tree Structure

Tree Output#	Pin #	Pin Name	Type	Comments
	111	A30	NI	A30 is the test mode output
1	39	HCLKIN	NI	End of NAND tree, goes to A30 in test mode
2	208	AD0	NI	
3	207	AD1	NI	
4	206	AD2	NI	
5	205	AD3	NI	
6	204	AD4	NI	
7	203	AD5	NI	
8	202	AD6	NI	
9	199	AD7	NI	
10	198	C/BE0#	NI	
11	197	AD8	NI	
12	196	AD9	NI	
13	195	AD10	NI	
14	194	AD11	NI	
15	192	AD12	NI	
16	191	AD13	NI	
17	190	AD14	NI	
18	189	AD15	NI	
19	188	C/BE1#	NI	
20	186	PAR	NI	
21	184	STOP#	NI	
22	183	DEVSEL#	NI	
23	182	KBDRST#	INV	
24	181	TRDY#	NI	
25	180	IRDY#	NI	
26	179	FRAME#	NI	
27	177	C/BE2#	NI	
28	176	AD16	NI	

Table 27. PSC NAND Tree Structure (Continued)

Tree Output #	Pin #	Pin Name	Type	Comments
29	175	AD17	NI	
30	174	AD18	NI	
31	173	AD19	NI	
32	171	AD20	NI	
33	170	AD21	NI	
34	169	AD22	NI	
35	168	AD23	NI	
36	167	C/BE3#	NI	
37	166	AD24	NI	
38	163	AD25	NI	
39	162	AD26	NI	
40	161	AD27	NI	
41	160	AD28	NI	
42	159	AD29	NI	
43	158	AD30	NI	
44	157	AD31	NI	
45	156	PGNT0#	NI	
46	154	PGNT1#/HRDY#	NI	
47	152	PREQ1#/HDEV#	NI	
48	151	WE#	NI	
49	150	CAS7#	NI	
50	149	CAS3#	INV	
51	148	CAS3#	INV	
52	146	CAS2#	NI	
53	145	CAS5#	NI	
54	144	CAS1#	NI	
55	143	CAS4#	NI	
56	141	CAS0#	NI	

Table 27. PSC NAND Tree Structure (Continued)

Tree Output #	Pin #	Pin Name	Type	Comments
57	140	RAS4#	NI	
58	139	RAS3#	NI	
59	138	RAS2#	NI	
60	137	RAS1#	NI	
61	136	RAS0#	NI	
62	135	COE0#	NI	
63	134	COE1#	NI	
64	133	CI3O2	NI	
65	132	CI3E	NI	
66	131	SMIACT#	INV	
67	130	LBIDE#	NI	
68	129	MA10	NI	
69	127	MA9	NI	
70	126	MA8	NI	
71	125	MA7	NI	
72	124	MA6	NI	
73	123	MA5	NI	
74	122	MA4	NI	
75	121	MA3	NI	
76	120	MA2	NI	
77	118	MA1	NI	
78	117	MA0	NI	
79	116	CLK2IN	NI	
80	115	CWE1#	NI	
81	113	CWEO#	NI	
82	112	AHOLD	NI	
83	110	HDP3	NI	
84	109	HD23	NI	

Table 27. PSC NAND Tree Structure (Continued)

Tree Output #	Pin #	Pin Name	Type	Comments
85	108	HD20	NI	
86	107	HD22	NI	
87	106	HD19	NI	
88	105	HD21	NI	
89	104	HD11	NI	
90	103	HD18	NI	
91	102	HD9	NI	
92	101	HD13	NI	
93	100	HD17	NI	
94	99	HD10	NI	
95	98	HDP1	NI	
96	95	HD8	NI	
97	94	HD15	NI	
98	93	HD12	NI	
99	92	HD24	NI	
100	91	HD25	NI	
101	90	HD3	NI	
102	89	HDP2	NI	
103	88	HD27	NI	
104	87	HD5	NI	
105	86	HD16	NI	
106	85	HD26	NI	
107	83	HD14	NI	
108	82	HD29	NI	
109	81	HD31	NI	
110	80	HD6	NI	
111	79	HD7	NI	
112	78	HD28	NI	

Table 27. PSC NAND Tree Structure (Continued)

Tree Output#	Pin #	Pin Name	Type	Comments
113	77	HD4	NI	
114	76	HD30	INV	
115	75	HD2	NI	
116	74	HD1	NI	
117	73	HDP0	NI	
118	72	HD0	NI	
119	71	EADS#	NI	
120	70	HOLD	NI	
121	69	RDY#	NI	
122	68	KEN#	NI	
123	67	BRDY#	NI	
124	66	PCD/CACHE#	NI	
125	65	D/C#	NI	
126	64	W/R#	NI	
127	63	MI/O#	NI	
128	62	HLDA	NI	
129	59	BLAST#	INV	
130	58	ADS#	INV	
131	57	HITM#	NI	
132	56	BE3#	NI	
133	55	BE2#	NI	
134	54	BE1#	NI	
135	53	BE0#	NI	
136	52	TWE#	NI	
137	51	TAG7	NI	
138	50	TAG6	NI	
139	49	TAG5	NI	
140	48	TAG4	NI	

Table 27. PSC NAND Tree Structure (Continued)

Tree Output #	Pin #	Pin Name	Type	Comments
141	47	TAG3	NI	
142	46	TAG2	NI	
143	45	TAG1	NI	
144	44	TAG0	NI	
145	42	TAG8	NI	
146	41	SRESET/INIT	NI	
147	37	SMI#	INV	
148	36	A4	NI	
149	35	A6	NI	
150	34	A3	NI	
151	33	A2	NI	
152	32	A10	NI	
153	31	A8	NI	
154	30	A7	NI	
155	29	A11	NI	
156	28	A5	NI	
157	26	A9	INV	
158	25	A13	NI	
159	24	A16	NI	
160	23	A20	NI	
161	22	A12	NI	
162	21	A15	NI	
163	20	A22	NI	
164	19	A24	NI	
165	18	A14	NI	
166	17	A18	NI	
167	16	A21	NI	
168	14	A19	NI	

Table 27. PSC NAND Tree Structure (Continued)

Tree Output #	Pin #	Pin Name	Type	Comments
169	13	A23	NI	
170	12	A17	NI	
171	11	A26	NI	
172	9	A25	NI	
173	8	A31	NI	
174	7	PCICLKIN	NI	
175	6	LGNT#	NI	
176	5	LREQ#	NI	
177	4	SIDLE#	NI	
178	3	CMDV#	NI	
179	2	SERR#	NI	Cell furthest from NAND Tree output

NOTES:
NI = Non-Inverting
INV = Inverting

8.1.3 PSC NAND TREE DIAGRAM

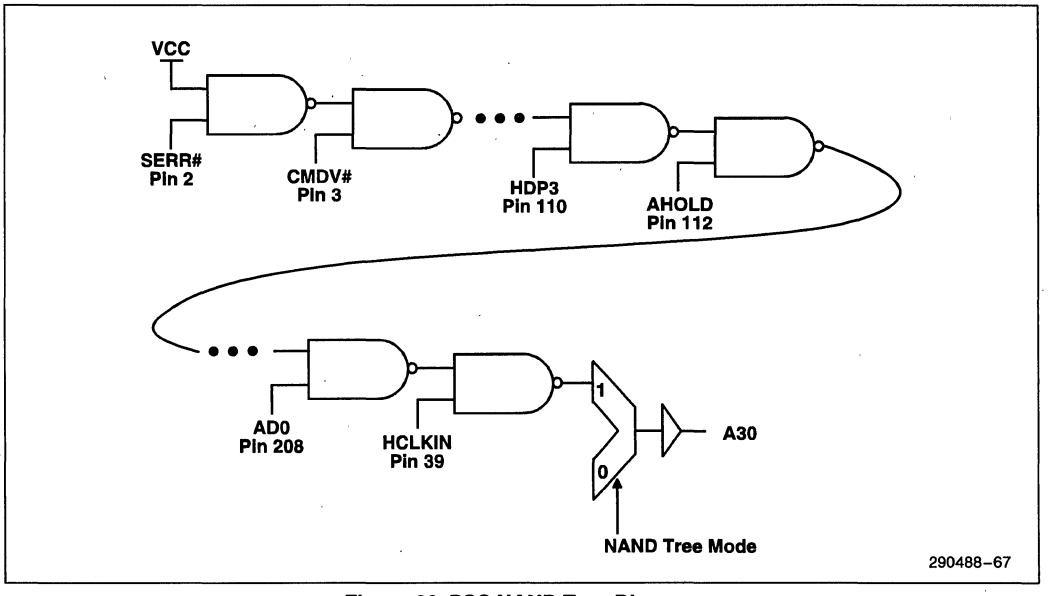


Figure 60. PSC NAND Tree Diagram

8.2 IB Testability

The TESTIN# pin is used to test the IB. During normal operations, the TESTIN# pin must be pulled high through an external pull-up.

8.2.1 IB TRI-STATE

The TESTIN# pin and IRQ3 are used to provide a high impedence tri-state test mode. When the following input combination occurs, all outputs and bidirectional pins are tri-stated, including SPKR:

TESTIN# = 0
IRQ3 = 1
IRQ5 = 0
IRQ6 = 1

The IB must be reset after the bidirectional and output pins have been tri-stated in this manner.

8.2.2 IB NAND TREE

A NAND Tree is provided primarily for VIL/VIH testing. The NAND Tree is also useful for ATE at board level testing. The NAND Tree allows for the tester to test the solder connections for each individual signal pin.

The TESTIN# pin along with IRQ5 and IRQ6 activate the NAND Tree. All outputs and bidirectional pins, except SPKR, are tri-stated when the following input combinations occur:

TESTIN# = 0 and IRQ5 = 1
or
TESTIN# = 0 and IRQ6 = 0

The output pulse train is observed at the SPKR test output, which is not tri-stated while in NAND Tree mode.

The sequence of the ATE test is as follows:

1. Drive TESTIN# low.
2. Drive each input and bidirectional pin noted in Section 8.2.3 high, except for SPKR.
3. Starting with pin 1, SYSCLK, individually drive each pin low. Expect SPKR to toggle with each pin.
4. Turn off tester drivers before driving TESTIN# high.
5. Reset the IB prior to proceeding with further testing.

8.2.3 IB NAND TREE CELL ORDER

NAND Tree cell order is dependent on pin placement. The IB NAND Tree follows pin order around the part from pin 1 to pin 158.

Table 28. IB NAND Tree Structure

Tree Output #	Pin #	Pin Name	Type	Comments
	69	SPKR	NI	Test Mode Output
1	158	SA12	NI	End of NAND Tree, goes to SPKR in test mode
2	157	REFRESH#	NI	
3	156	SA13	NI	
4	155	DREQ1	NI	
5	154	SA14	NI	
6	153	DACK1#	NI	
7	152	SA15	NI	
8	151	DREQ3	NI	
9	149	SA16	NI	
10	148	DACK3#	NI	
11	147	SA17	NI	
12	146	IOR#	NI	
13	145	SA18	NI	
14	144	IOW#	NI	
15	143	SA19	NI	
16	142	SMEMR#	NI	
17	139	AEN	NI	
18	138	SMEMW#	NI	
19	137	IOCHRDY	NI	
20	136	SD0	NI	
21	135	SD1	NI	
22	134	ZEROWS#	NI	
23	133	SD2	NI	
24	132	SD3	NI	
25	131	DREQ2	NI	
26	129	SD4	NI	
27	128	SD5	NI	
28	127	IRQ9	NI	

Table 28. IB NAND Tree Structure (Continued)

Tree Output #	Pin #	Pin Name	Type	Comments
29	126	SD6	NI	
30	125	SD7	NI	
31	124	RSTDRV	NI	
32	123	IOCHK#	NI	
33	122	KBCCS#	NI	
34	119	IRQ1	NI	
35	118	PIRQ0#	NI	
36	117	PIRQ1#	NI	
37	116	SERR#	NI	
	115	TESTIN#	NI	Not part of NAND tree
38	114	PWROK	NI	
39	113	PCICLK2	NI	
40	112	PCICLK1	NI	
41	111	PCIRST#	NI	
42	109	CLK2OUT	NI	
43	108	CLK2IN	NI	
44	107	CMDV#	NI	
45	106	SIDLE#	NI	
46	105	LREQ#	NI	
47	104	LGNT#	NI	
48	103	A17	NI	
49	102	A14	NI	
50	101	A15	NI	
51	99	OSC	NI	
52	98	A12	NI	
53	97	A16	NI	
54	96	A13	NI	
55	95	A9	NI	
56	94	A5	NI	

Table 28. IB NAND Tree Structure (Continued)

Tree Output #	Pin #	Pin Name	Type	Comments
57	93	A11	NI	
58	92	A7	NI	
59	91	A8	NI	
60	89	A10	NI	
61	88	A6	NI	
62	87	A4	NI	
63	86	A2	NI	
64	85	A3	NI	
65	84	SMI#	NI	
66	83	HCLKIN	NI	
67	82	HCLKOUT1	NI	
68	81	HCLKOUT2	NI	
69	79	CPURST	NI	
70	78	SRESET	NI	
71	77	NMI	NI	
72	76	IGNNE#	NI	
73	75	FERR#	NI	
74	73	INTR	NI	
75	72	STPCLK#	NI	
76	71	EXTSMI#	NI	
77	68	RTCALE	NI	
78	67	RTCCS#	NI	
79	66	BIOSCS#	NI	
80	65	XBUSOE#	NI	
81	64	XBUSTR#	NI	
82	63	IRQ8#	NI	
83	62	SD15	NI	
84	61	SD14	NI	

Table 28. IB NAND Tree Structure (Continued)

Tree Output #	Pin #	Pin Name	Type	Comments
85	59	SD13	NI	
86	58	SD12	NI	
87	57	DREQ7	NI	
88	56	SD11	NI	
89	55	DACK7#	NI	
90	54	SD10	NI	
91	53	DREQ6	NI	
92	52	SD9	NI	
93	51	DACK6#	NI	
94	49	SD8	NI	
95	48	DREQ5	NI	
96	47	MEMW#	NI	
97	45	DACK5#	NI	
98	44	MEMR#	NI	
99	43	DREQ0	NI	
100	42	LA17	NI	
101	41	DACK0#	NI	
102	40	LA18	NI	
103	39	IRQ14	NI	
104	37	LA19	NI	
105	36	IRQ15	NI	
106	35	LA20	NI	
107	34	IRQ12/M	NI	
108	33	LA21	NI	
109	32	IRQ11	NI	
110	31	LA22	NI	
111	30	IRQ10	NI	
112	29	LA23	NI	

Table 28. IB NAND Tree Structure (Continued)

Tree Output#	Pin #	Pin Name	Type	Comments
113	28	IOCS16#	NI	
114	26	SBHE#	NI	
115	25	MEMCS16#	NI	
116	24	SA0	NI	
117	23	SA1	NI	
118	22	SA2	NI	
119	21	SA3	NI	
120	18	BALE	NI	
121	17	SA4	NI	
122	16	TC	NI	
123	15	SA5	NI	
124	14	DACK2#	NI	
125	13	SA6	NI	
126	12	IRQ3	NI	
127	11	SA7	NI	
128	9	IRQ4	NI	
129	8	SA8	NI	
130	7	IRQ5	NI	
131	6	SA9	NI	
132	5	IRQ6	NI	
133	4	SA10	NI	
134	3	IRQ7	NI	
135	2	SA11	NI	
136	1	SYSCLK	NI	Cell furthest from NAND Tree output

8.2.4 IB NAND TREE DIAGRAM

Figure 61 shows the NAND Tree diagram. The only function pin not included in the pin order is SPKR, which is used as the Test Output at the end of the tree.

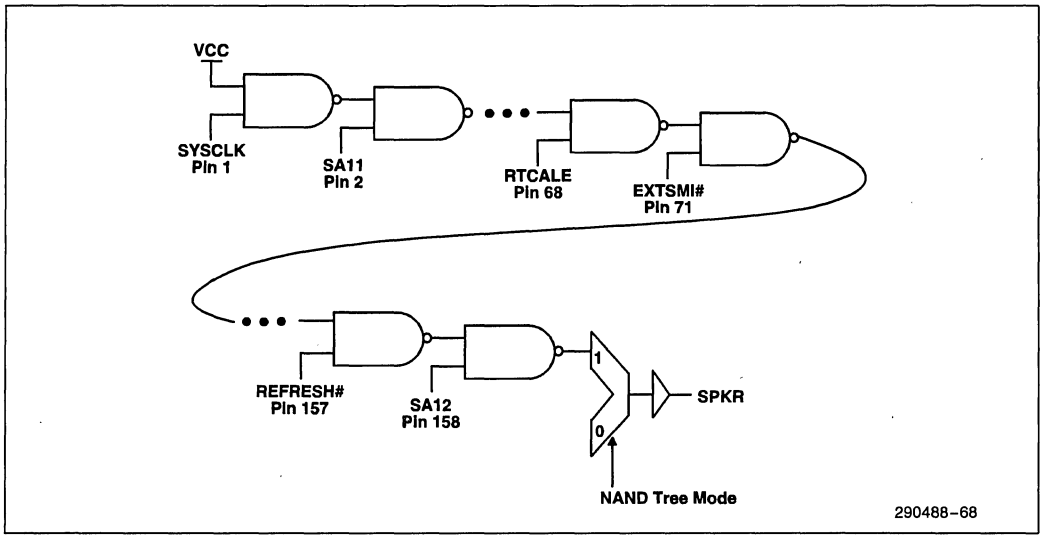


Figure 61. IB NAND Tree Diagram

9.0 REVISION HISTORY

Revision -003 of the 82420EX PCIset data sheet contains updates and improvements to the original version. A revision summary of changes is listed below.

The sections significantly revised since revision -001 are:

Global	The 82420EX PCIset supports host bus operations of 25 and 33 MHz. All references to 50 MHz support have been removed.
Sections 1.1, 1.2, 2.1.1	PGNT1#/HRDY# and M/IO# signal type was incorrectly published. M/IO# and PGNT1#/HRDY# are I/O pins. EXTSMI# and PWROK signal type was incorrectly published. EXTSMI# and PWROK are IS pins. SERR# was added to Table 3.
Section 2.2.4	Pin description for the OSC pin has been added.
Section 3.1	Data returned during accesses to a reserved register location was incorrectly published. Reads return all 0's during accesses to a reserved register location.
Section 3.3.5	More information is provided on the contents of the Revision Identification Register.
Section 3.3.6	The byte merging feature description has been updated.
Section 3.3.10	Part Revision Register description has been updated.
Section 3.3.20	A warning has been added concerning O/S which size system memory directly without the use of system BIOS.

Section 3.3.21	Bits[2:0] = 000 and bits[2:0] = 001 are now reserved configurations in the SMRAM Control Register.
Section 3.3.22	Changes have been made to the Fast Off Timer Count Granularity.
Section 3.3.25	Changes have been made to the Fast Off Timer count down value.
Section 3.3.26	Bits[15:8] were incorrectly omitted from the SMI Request Register. Bits[15:8] are shown as reserved.
Section 3.3.27	Value of the duration of the STPCLK# asserted period has changed.
Section 3.3.28	Value of the duration of the STPCLK# negated period has changed.
Section 4.3.4	Further information has been added to the keyboard controller circuit description.
Section 4.12.1	Table 19 has been updated. New notes have been added to Table 19.
Section 4.13.6	Table 21 has been updated.
Section 5.0	Section 5.0 has been removed. The 82420EX PCIset electrical characteristics will be published as a separate document. See your Intel representative for a copy of the document with the 82420EX PCIset electrical characteristics.
Section 6.0	Section 6.0 is now Section 5.0. Figure 58 and Figure 59 have changed.
Section 6.1	Section 6.1 is now Section 5.1. Table 25 and Table 26 have changed.

The sections significantly revised since revision -002 are:

Section 3.3.22	The description of bit 1 in the SMI Control Register has changed. Software can only set bit 1 to a 0 by writing a 0 to it.
Section 5.0	Section 5.0 has been added. This is a new section titled Design Considerations.
Section 6.0	Section 6.0 has been added. This section includes the AC, DC and mechanical specifications and timings. The following IB specifications have changed since the electrical characteristics were last published in revision -001: t1c (min), t1f (min). The following PSC specifications have changed since the electrical characteristics were last published in revision -001: t1c (min), t1d (min), t41b (min), t41c (min).

The sections significantly revised since revision -003 are:

1. References to programmable access timings to support 50 MHz operation of the PSC are removed.
2. Section 4.15.1, Clock, Layout/Loading Recommendation, is added.
3. Section 6.3.4, ISA Bus and X-Bus Timings: t6a, t6d, t6e, t16a, t27a, are changed.
4. Section 6.4.9, Miscellaneous Clock Timings, is added.

82378ZB SYSTEM I/O (SIO) AND 82379AB SYSTEM I/O APIC (SIO.A)

- Provides the Bridge between the PCI Bus and ISA Bus
- 100% PCI and ISA Compatible
  - PCI and ISA Master/Slave Interface
  - Directly Drives 10 PCI Loads and 6 ISA Slots
  - PCI at 25 MHz and 33 MHz
  - ISA from 6 MHz to 8.33 MHz
- Enhanced DMA Functions
  - Scatter/Gather (82378ZB)
  - Fast DMA Type A, B and F (82378ZB)
  - Compatible DMA Transfers
  - 32-bit Addressability(82378ZB)
  - 27-bit Addressability(82379AB)
  - Seven Independently Programmable Channels
  - Functionality of Two 82C37A DMA Controllers
- Data Buffers to Improve Performance
  - 8-Byte DMA/ISA Master Line Buffer
  - 32-bit Posted Memory Write Buffer to ISA
- Integrated 16-bit BIOS Timer
- Non-Maskable Interrupts (NMI)
  - PCI System Errors
  - ISA Parity Errors
- Arbitration for ISA Devices
  - ISA Masters
  - DMA and Refresh
- Four Dedicated PCI Interrupts
  - Level Sensitive
  - Mapped to Any Unused Interrupt
- Arbitration for PCI Devices
  - Six PCI Masters Supported
  - Fixed, Rotating, or a Combination
- Utility Bus (X-Bus) Peripheral Support
  - Provides Chip Select Decode
  - Controls Lower X-Bus Data Byte Transceiver
- Functionality of One 82C54 Timer
  - System Timer
  - Refresh Request
  - Speaker Tone Output
- Functionality of Two 82C59 Interrupt Controllers
  - 14 Interrupts Supported
  - Edge/Level Selectable Interrupts
- I/O APIC (Advanced Programmable Interrupt Controller (82379AB)
  - Support for Multi-Processor Systems
- System Power Management
  - Programmable System Management Interrupt (SMI)Hardware Events, Software Events, EXTSMI#
  - Programmable CPU Clock Control (STPCLK#)
  - Fast-On/Off Mode
- 208 Pin QFP Package

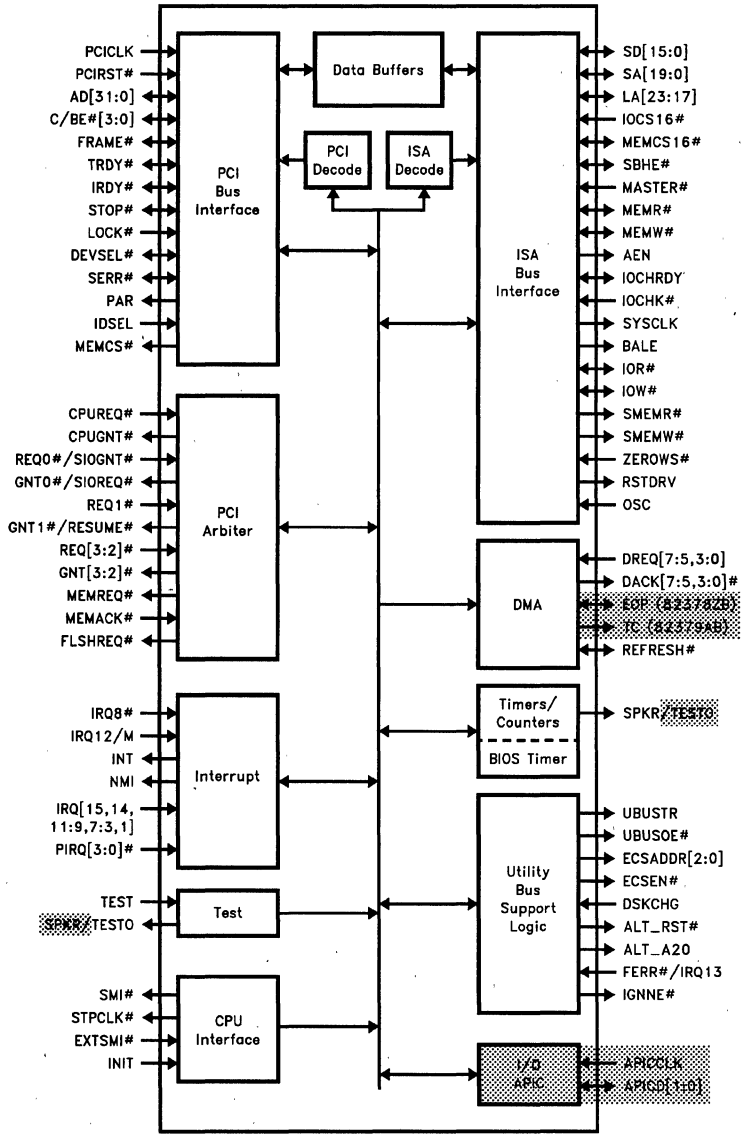
1

The 82378ZB System I/O (SIO) and 82379AB System I/O—APIC (SIO.A) components are PCI-to-ISA Bus Bridge devices. These devices integrate many of the common I/O functions found in today's ISA-based PC systems—a seven channel DMA controller, two 82C59 interrupt controllers, an 8254 timer/counter, a BIOS timer, Intel SMM power management support, and logic for NMI generation. In addition, the SIO and SIO.A each support a total of 6 PCI Masters, and 4 PCI Interrupts. Decode is provided for peripheral devices such as the Flash BIOS, Real Time Clock, Keyboard/Mouse Controller, Floppy Controller, two Serial Ports, one Parallel Port, and IDE Hard Disk Drive. For both the SIO and SIO.A, each DMA channel supports compatibility transfers. The SIO also supports types A, B, and F transfers and scatter/gather. In addtion to the standard ISA-compatible interrupt controller that is in both the SIO and SIO.A, the SIO.A contains an Advance Programmable Interrupt Controller (IO APIC) for use in multi-processing systems.

This document describes both the 82378ZB (SIO) and 82379AB (SIO.A) components. Unshaded areas describe the 82378ZB. Shaded areas, like this one, describe differences between the 82379AB and 82378ZB.

The complete document for this product is available from Intel's Literature Center at 1-800-548-4725.

November 1995
Order Number: 290571-001



290571-1

NOTE:
The I/O APIC signals are only on the 82379AB. Also, SPKR/TESTO functions are multiplexed on the same singal pin on the 82379AB and are separate signals on the 82378ZB.

82378ZB and 82379AB Component Block Diagram

PRELIMINARY

1

82350
EISA Chip Set

January 1994

Order Number: 290220-004

82350 EISA CHIP SET

CONTENTS

EISA TERMINOLOGY

ISA BUS— The bus used in Industry Standard Architecture compatible computers. In the context of an EISA system, it refers to the ISA subset of the EISA bus.

EISA BUS— Extended ISA bus, a superset of the ISA bus. It includes all ISA bus features, along with extensions to enhance performance and capabilities.

HOST CPU— The main system processor, located on a separate Host Bus. This uses the EBC and other system board facilities to interface to the EISA bus.

CPU CYCLE— 386 CPU and/or the 82385 subsystem, or 80486 CPU is the master running the cycle.

EISA MASTER— A 16-bit or 32-bit bus master that uses the EISA signal set to generate memory or I/O cycles. The bus controller will convert the EISA control signals to ISA signals, when necessary.

ISA MASTER—A 16-bit bus master that uses the ISA subset of the EISA bus for generation of memory or I/O cycles. This device must understand 8-bit or 16-bit ISA slaves, and route data to the appropriate byte lanes. It is not required to handle any of the signals associated with the extended portion of the EISA bus.

EISA SLAVE— An 8-bit, 16-bit or 32-bit memory or I/O slave device that uses the extended signal set of the EISA bus to accept cycles from various masters. It returns information about its type and width using extended and ISA signals.

ISA SLAVE— A 16-bit or 8-bit slave that uses the ISA subset of the EISA bus to accept cycles from various masters. It returns ISA signals to indicate its type and width.

DMA SLAVE— An I/O device that uses the DMA signals (DREQ, DACK#) of the system board ISP to perform a direct memory access.

ISACMD— The ISA command signals (IORC#, IOWC#, MRDC#, MWTC#)

ASSEMBLY/DISASSEMBLY— This occurs when the master/slave data bus size are mismatched. The EBC runs multiple cycles to route bytes to the appropriate byte lanes (byte swapping). For example, if the 32-bit CPU is accessing an 8-bit slave, the EBC will need to run four cycles to the 8-bit slave and route the bytes to appropriate byte lanes

CYCLE TRANSLATION— This is performed by the EBC when the master and slave are on different busses (Host/EISA/ISA). The EBC will translate the master protocol to the slave protocol (Host master accessing EISA slave).

EISA System Introduction

Extended Industry Standard Architecture (EISA) is a high performance 32-bit architecture based upon the Industry Standard Architecture (ISA) (PC AT*). The wide acceptance of the 32-bit 386 microprocessor family has led to this interest in extending ISA to 32-bits. EISA's advanced capabilities and 32-bit architecture can unleash the full potential of the 386 and i486™ CPUs.

The EISA consortium has defined the EISA bus in response to the demand for a 32-bit high performance ISA compatible system. The open industry standard allows for industry wide participation, compatibility, and differentiation.

EISA brings advances in performance and convenience to the user. It provides 32-bit memory addressing and data transfers for CPU, DMA and bus masters allowing 33 Mbyte/second transfer rate for DMA and bus masters on the EISA bus. EISA provides a specification for auto-configuration of add-in cards that will eliminate the need for jumpers and switches on EISA cards. Interrupts are shareable and programmable. Figure 1 and 2 show the types of busses in an EISA system. A new bus-arbitration makes possible a new generation of intelligent bus master add-in cards that bring advanced applications to PCs.

Since the EISA system is 100% compatible with the ISA 8-bit and 16-bit expansion boards and software, ISA cards can be plugged into the EISA connector slots. The EISA slots can be defined as ISA or EISA for ease of compatibility during configuration. The EISA connector is a superset of the ISA connector maintaining full compatibility with ISA expansion cards and software. Simultaneous use of EISA and ISA add-in boards is available with automatic system and expansion board configuration.

82350 EISA Chip Set Highlights

The Intel 82350 EISA chip set is the industry's first 100% EISA/ISA compatible chip set. The 82350

Intel486 is a trademark of Intel Corporation.

*PC AT is a trademark of International Business Machine Corporation.

EISA chip set supports the 33 MHz and 25 MHz 386 CPU or i486 CPU, 82385 Cache Controller, and optional 80387 numerics coprocessor. The EISA chip set includes three chips:

82352DT	EISA Bus Buffers (EBB) (Optional)
82357	Integrated System Peripheral (ISP)
82358	EISA Bus Controller (EBC)

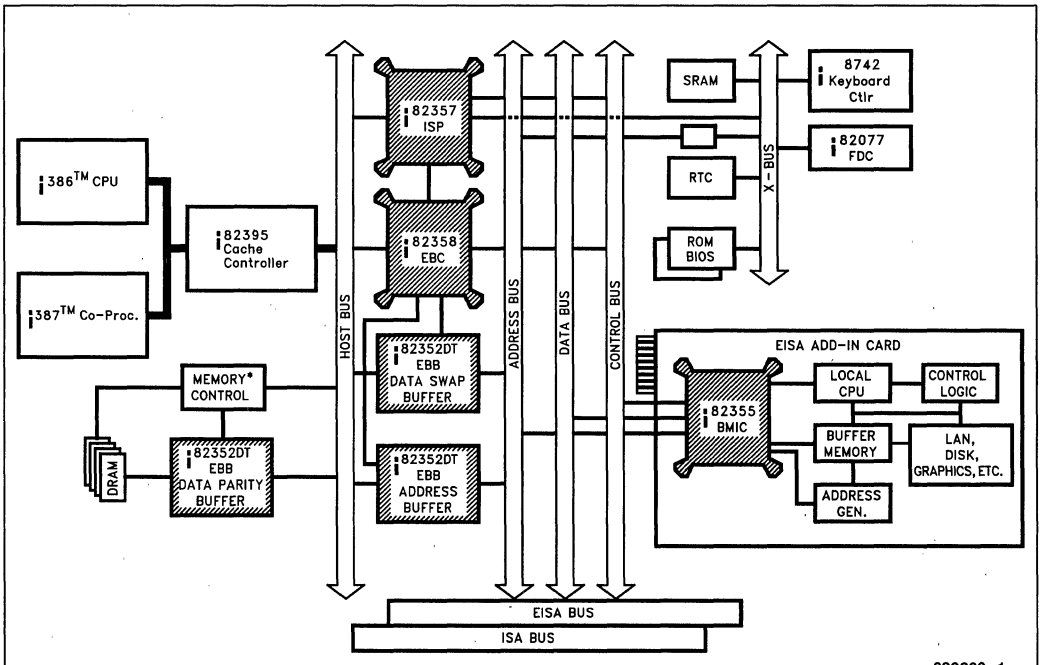
Information on the 82352DT EBB device is located in a separate data sheet.

The ISP performs the DMA functions of the system and is fully compatible with ISA functions. It integrates seven 32-bit DMA channels, five 16-bit timer/counters, two eight channel interrupt controllers, and provides for multiple NMI control and generation. It provides refresh address generation and keeps track of pending refresh requests when the bus is unavailable. The ISP supports multiple EISA bus masters while offering intelligent system arbiter services which grant the bus on a rotational basis.

The EBC is the EISA "engine". It is an intelligent bus controller that controls 8, 16 and 32-bit bus masters and slaves. It provides the state machine interface to Host, ISA and EISA busses and other IC's in the chip set. It offers a simple interface to the 386/i486 CPU and EISA bus. The EBC services as a bridge between the EISA and ISA devices. Data bus size mismatches are handled automatically by the EBC (including byte assembly and disassembly). It also guarantees cache operation on the Host, EISA, and ISA busses.

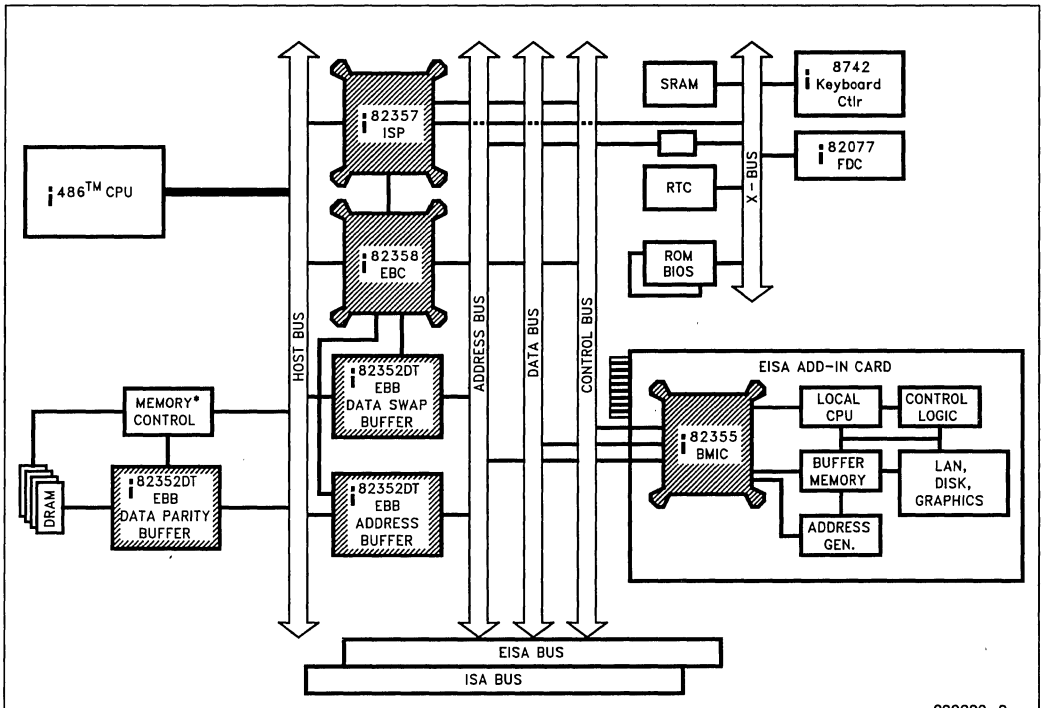
More information on EBC and ISP devices can be found in the data sheets in this document.

The 82355 Bus Master Interface Chip (BMIC) is a new device for add-in cards that takes advantage of the EISA bus master capabilities. Information on the 82355 BMIC is located in a separate data sheet.



NOTE:
*Memory Controller can be implemented as:
1) Discrete
2) ASIC (Custom)
3) ASIC (Standard)

Figure 1. Intel's 386 CPU System with 82350 EISA Chip Set



NOTE:
*Memory Controller can be implemented as:
1) Discrete
2) ASIC (Custom)
3) ASIC (Standard)

Figure 2. Intel's i486™ CPU System with 82350 EISA Chip Set

1

MECHANICAL DATA

PACKAGING INFORMATION

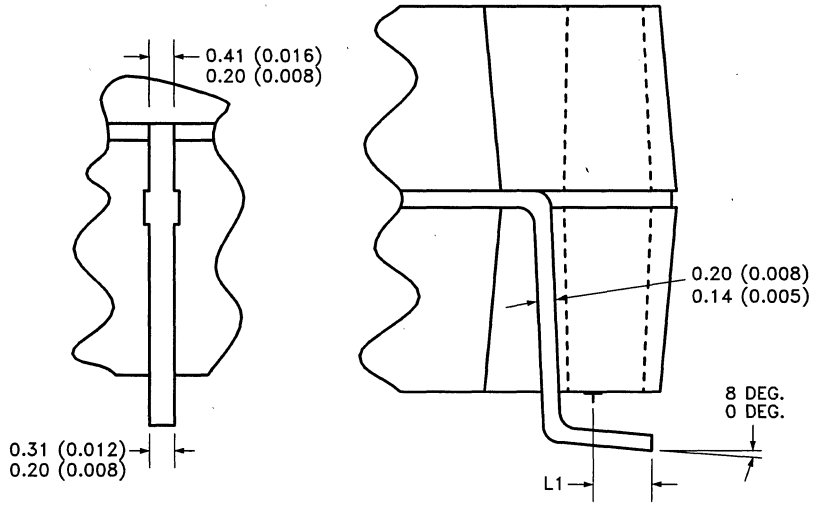
(See Packaging Spec. Order # 231369)

Introduction

The individual components of Intel's EISA Chip Set come in JEDEC standard Gull Wing packages (25 MIL pitch), with "bumpers" on the corners for ease of handling. Please refer to the accompanying table for the package associated with each device, and to the individual component specifications for pinouts. (Note that the individual pinouts are numbered consistently with the numbering scheme depicted in the accompanying figures).

TYPICAL LEAD

Case Outline Drawings
Plastic Fine Pitch Chip Carrier
0.84 mm Pitch

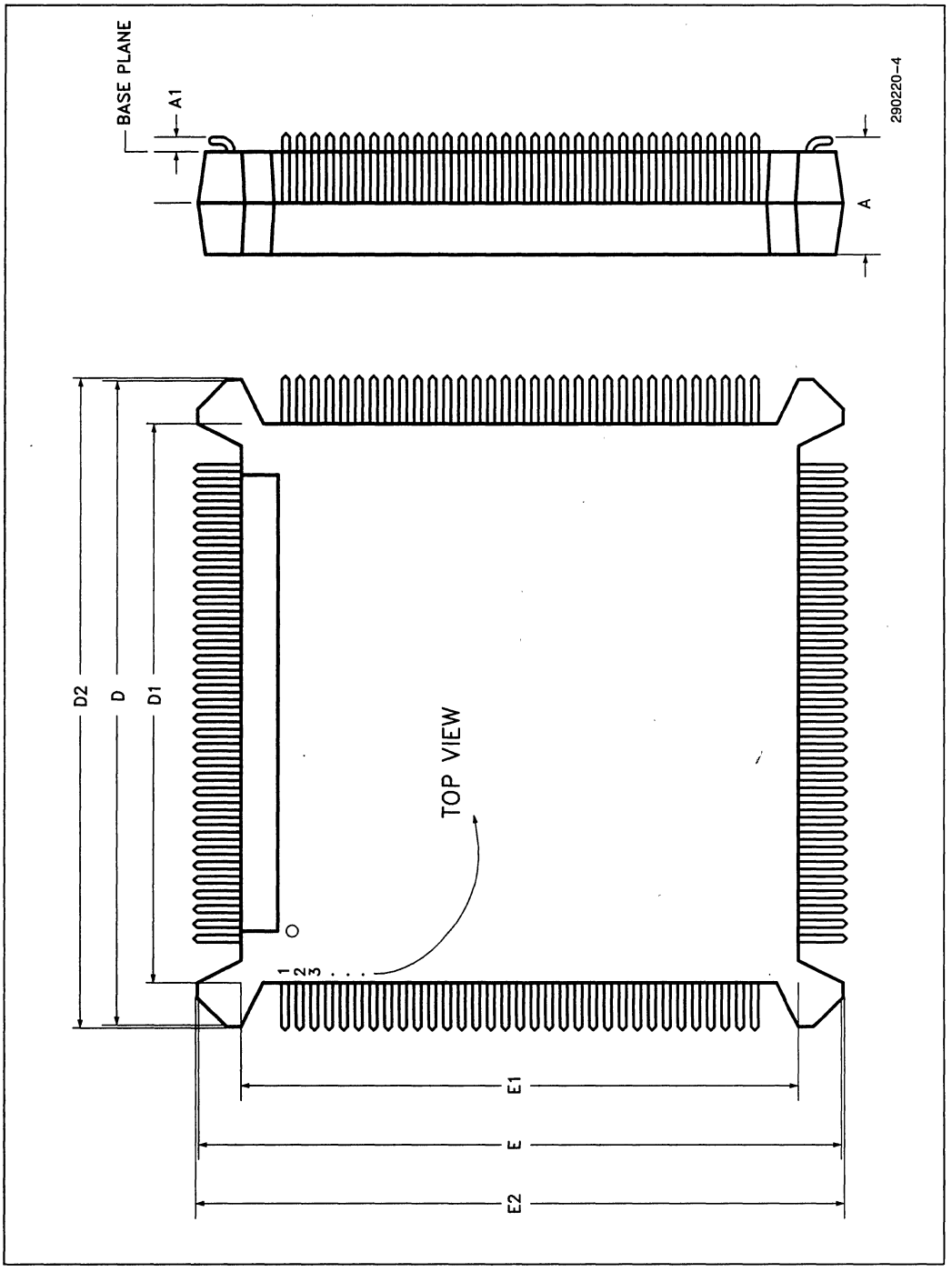


mm (inch)

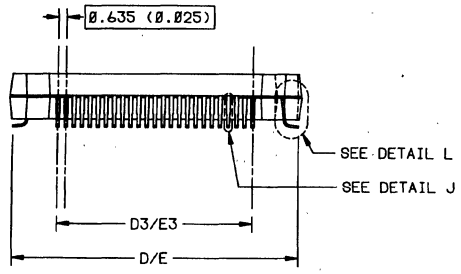
290220-3

Symbol	Description	Inch		mm	
		Min	Max	Min	Max
N	Lead Count	132		132	
A	Package Height	0.160	0.170	4.06	4.32
A1	Standoff	0.020	0.030	0.51	0.76
D, E	Terminal Dimension	1.075	1.085	27.31	27.56
D1, E1	Package Body	0.947	0.953	24.05	24.21
D2, E2	Bumper Distance	1.097	1.103	27.86	28.02
D3, E3	Lead Dimension	0.800 Ref		20.32 Ref	
L1	Foot Length	0.020	0.030	0.51	0.76

PRINCIPAL DIMENSIONS & DATUMS



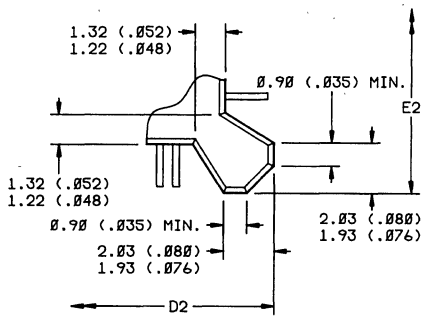
TERMINAL DETAILS



290220-5

mm (inch)

BUMPER DETAIL



290220-6

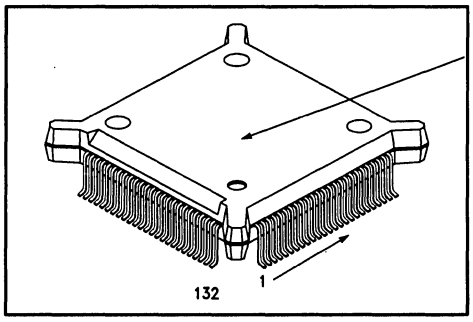
mm (inch)

Package Thermal Specification

The 82357 ISP and 82358 EBC are specified for operation when the case temperature is within the range of 0°C to 85°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 in the figure below.

PLASTIC QUAD FLAT PACK (PQFP)



290220-7

Table 2. 82357 ISP and 82358 DT EBC PQFP Package Thermal Characteristics

Thermal Resistance— °C/Watt							
Parameter	**Air Flow Rate (ft/min)**						
	0	**50**	**100**	**200**	**400**	**600**	**800**
θ Junction—Case	7	7	7	7	7	7	7
θ Case to Ambient	22	21	19.5	17.5	14.5	12	10

NOTES:
1. Table 2 applies to the PQFP device plugged into a socket or soldered directly into the board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$.

PROCESS NAME:
1.2μ CHMOS III P-well

I_{CC} AT HOT WITH NO RESISTIVE LOADS:
150 mA Max at 85°C.

82352DT
EISA BUS BUFFER (EBB)

- **Designed Specifically for EISA Bus Requirements**
- **Provides Three Modes of Operation**
  - **— Data Latch and Swap Functions Allow Swapping and Assembly of Data between the Host and EISA/ISA Buses on a Byte by Byte Basis (Mode 0)**
  - **— Provides a Buffered Path with Parity Generation/Check between the Host Data Bus and DRAM (Mode 1)**
  - **— Address Latch Functions Provide Latching between the Host and EISA/ISA Buses (LA and SA Addresses) (Mode 3)**
- **120-Pin Quad Flat Pack (QFP)**
- **Similar in Function to Discrete Implementation Using 74F543s/544, 74180s, and 74ALS245s**
- **Replaces 19 Discrete Components**
  - **— Three 82352DTs are Used Per 82350 EISA System**
- **The 82352DT Interfaces Easily to the System**
  - **— Buffer Control for the 32-Bit Mode W/O Parity and the EISA Address Mode is Provided by the 82358 (EISA Bus Controller)**

(See Packaging Specification Order Number 240800, Package Type S)

The 82352DT design allows it to replace the multiple address and data latch-buffer/driver ICs used in EISA applications. The EBB provides three modes of operation: a 32-bit mode without parity to replace the EISA data swap buffers, a 32-bit mode with parity to replace the EISA DRAM data parity buffers, and an EISA address mode to replace the host to EISA/ISA address buffers. Mode 2 on the EBB is reserved. The same chip is strapped in three different ways to obtain the three configurations.

82352DT is manufactured and tested for Intel by LSI Logic in accordance with their internal standards.

The complete document for this product is available from Intel's Literature Center at 1-800-548-4725.

October 1993
Order Number: 290254-007

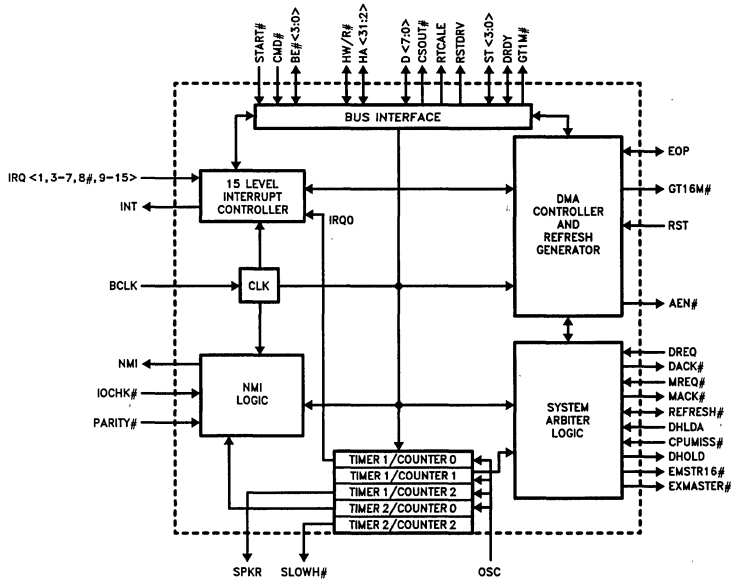
82357
INTEGRATED SYSTEM PERIPHERAL (ISP)

- **Provides Enhanced DMA Functions**
  - **— ISA/EISA DMA Compatible Cycles**
  - **— All Transfers are Fly-By Transfers**
  - **— 32-Bit Addressability**
  - **— Seven Independently Programmable Channels**
  - **— Provides Timing Control for 8-, 16-, and 32-Bit DMA Data Transfers**
  - **— Provides Timing Control for Compatible, Type "A", Type "B", and Type "C" (Burst) Cycle Types**
  - **— 33 Mbytes/sec Maximum Data Transfer Rate**
  - **— Provides Refresh Address Generation**
  - **— Supports Data Communication Devices and Other Devices That Work from a Ring Buffer in Memory**
  - **— Incorporates the Functionality of Two 82C37A DMA Controllers**
- **Provides High Performance Arbitration**
  - **— For CPU, EISA/ISA Bus Masters, DMA Channels, and Refresh**
- **Incorporates the Functionality of Two 82C59A Interrupt Controllers**
  - **— 14 Independently Programmable Channels for Level-or-Edge Triggered Interrupts**
- **Five Programmable 16-Bit Counter/ Timers**
  - **— Generates Refresh Request Signal**
  - **— System Timer Interrupt**
  - **— Speaker Tone Output**
  - **— Fail-Safe Timer**
  - **— Periodic CPU Speed Control**
  - **— 82C54 Programmable Interval Timer Compatible**
- **Provides Logic for Generation/Control of Non-Maskable Interrupts**
  - **— Parity Errors for System and Expansion Board Memory**
  - **— 8 μs and 32 μs Bus Timeout**
  - **— Immediate NMI Interrupt via Software Control**
  - **— Fail-Safe Timer**
- **132-Pin PQFP Package**

  (See Packaging Specifications: Order Number 240800, Package Type NG)

1

82357 Internal Block Diagram



290253-78

Intel486 is a trademark of Intel Corporation.

The complete document for this product is available from Intel's Literature Center at 1-800-548-4725.

March 1994
Order Number: 290253-006

82358DT
EISA BUS CONTROLLER

- **Supports 82350 and 82350DT Chip Set Based Systems**
  - **— Mode Selectable for Either 82350 or 82350DT Based Systems**
  - **— Mode Defaults to 82350 Based Systems**
- **Socket Compatible with the 82358 (EISA Bus Controller)**
- **Provides EISA/ISA Bus Cycle Compatibility**
  - **— EISA/ISA Standard Memory or I/O Cycles**
  - **— EISA/ISA Wait State Cycles**
  - **— ISA No Wait State Cycles**
  - **— EISA Burst Cycles**
- **Supports Intel386™ & Intel486™ Microprocessors**
- **Translates Host (CPU) and 82359 (DRAM Controller) Cycles to EISA/ISA Bus Cycles**
- **Generates ISA Signals for EISA Masters**
- **Generates EISA Signals for ISA Masters**
- **Supports 8-, 16-, or 32-bit DMA Cycles**
  - **— Type A, B, or C (Burst) Cycles**
  - **— Compatible Cycles**
- **Supports Host and EISA/ISA Refresh Cycles**
- **Generates Control Signals for Address and Data Buffers**
  - **— 82353 (ADP) and 82352 (EBB)**
- **Supports Byte Assembly/Disassembly for 8-, 16-, or 32-Bit Transfers**
- **Selectable Host (CPU) Posted Memory Write Support to EISA/ISA Bus**
- **Cache Controller (82385, 82395) Interface to Maximize Performance for 386 Based Systems**
- **Supports I/O Recovery Mechanism**
- **Generates CPU, 82385, and System Software Resets**
- **132-Pin PQFP Package**
- **Low Power CHMOS Technology**

(See Packaging Specification Order #240800, Package Type NG)

The 82358DT EISA Bus Controller is part of Intel's 82350 and 82350DT chip sets. There are five mode or function select pins which allow the 82358DT to be programmed for use in either 82350 or 82350DT based systems. The mode pins also provide support for posted memory write cycles to the EISA/ISA bus and Intel486™ burst support. The 82358DT defaults to 82350 mode and is 100% socket compatible with the 82358 (EBC).

The 82358DT interfaces the 386 and Intel486 microprocessors to the Extended Industry Standard Architecture (EISA) bus. It is used to facilitate bus cycles between the Host (CPU) bus and the EISA/ISA bus. In an 82350 system, the 82358DT interfaces to the cycle address and control signals of the Host bus. In an 82350DT system, the 82358DT interfaces to the cycle address and control signals of the 82359 DRAM controller. The 82358DT generates the appropriate data conversion and alignment control signals to implement an external byte assembly/disassembly mechanism for transferring data of different widths between the Host, EISA, and Industry Standard Architecture (ISA) buses. It also provides the cycle translation between the Host, EISA, and ISA buses.

The 82358DT is tightly coupled with the 82357 DMA controller (ISP) to run 8-. 16-, or 32-Bit EISA/ISA DMA transfers.

The 82358DT features hardware enforced I/O recovery logic to provide I/O recovery time between back- to-back I/O cycles.

The 82358DT provides special cache hardware interface signals to implement a high performance 386 based system with an 82385 or 82395 cache controller.

The 82358DT also provides resets to the Intel486, 80386, 82385, and other devices in the system to provide an integrated synchronous system reset.

The complete document for this product is available from Intel's Literature Center at 1-800-548-4725.

March 1994
Order Number: 290380-004

82420/82430
PCIset BRIDGE COMPONENT

82378ZB (SIO), 82379AB (SIO.A) FOR ISA BUSES

- Provides the Bridge between the PCI Bus and ISA Bus
- 100% PCI and ISA Compatible
- Enhanced DMA Functions (82378ZB Only)
- Integrated Data Buffers to Improve Performance
- Integrated 16-bit BIOS Timer
- Arbitration for PCI Devices
- Arbitration for ISA Devices
- Integrates the Functionality of One 82C54 Timer
- Integrates the Functionality of Two 82C59 Interrupt Controllers
- Non-Maskable Interrupts (NMI)
- Four Dedicated PCI Interrupts
- Complete Support for SL Enhanced Intel486™ CPU's
- Integrated Power Management Support
  - — System Management Interrupts
  - — Fast Off Timer
  - — STPCLK# Signal to Throttle CPU Clock
  - — APM Port
- Provides I/O APIC for Dual-Processor (DP) Support

82374EB/SB (ESC), 82375EB/SB (PCEB) FOR EISA BUSES

- Provides the Bridge between the PCI Bus and EISA Bus
- 100% PCI and EISA Compatible
- Data Buffers Improve Performance
- Data Buffer Management Ensures Data Coherency
- Burst Transfers on both the PCI and EISA Buses
- 32-Bit Data Paths
- PCI and EISA Address Decoding and Mapping
- Programmable Main Memory Address Decoding
- Integrated EISA Compatible Bus Controller
- Supports Eight EISA Slots
- Provides Enhanced DMA Controller
- Provides High Performance Arbitration
- Integrates Support Logic for X-Bus Peripheral and more
- Integrates the Functionality of Two 82C59 Interrupt Controllers and Two 82C54 Timers
- Generates Non-Maskable Interrupts
- Provides BIOS Interface

1

The 82420/82430 PCIset Bridge components provide a bridge between the PCI to either EISA or ISA buses. The 82378 provides the bridge between PCI bus and the ISA bus while the 82374 and 83275 together provide the bridge between the PCI bus and the EISA bus.

The SIO integrates many of the common I/O functions found in todays ISA based PC systems. The SIO incorporates the logic for a PCI interface (master and slave), ISA interface (master and slave), enhanced seven channel DMA controller and support for other decode logic. The 82379AB adds an APIC for dual-processing Pentium™ Processor systems.

November 1994
Order Number: 290509-002

The 82374 EISA System Component (ESC) and 83275 PCI-EISA Bridge (PCEB) together provide the EISA system compatible master/slave functions on both the PCI Local Bus and the EISA Bus and the common I/O functions found in today's EISA systems. The ESC incorporates the logic for an EISA (master and slave) interface, EISA bus controller, enhanced seven channel DMA controller with Scatter-Gather support, EISA arbitration, 14 channel interrupt controller, five programmable timer/counters and non-maskable control logic. The ESC also integrates support logic to decode peripheral devices such as the Flash BIOS, Real Time Clock, Keyboard/Mouse Controller, Floppy Controller, two Serial Ports, one Parallel Port, and IDE Hard Disk Drive. The PCEB provides the address and data paths, bus controls, and bus protocol translation for PCI-to-EISA and EISA-to-PCI transfers. Extensive data buffering in both directions increases system performance by maximizing PCI and EISA Bus efficiency and allowing concurrency on the two buses. The PCEB integrates central bus control functions, PCI parity generation, system error reporting, and programmable PCI and EISA memory and I/O address space mapping and decoding.

82374EB/82374SB EISA SYSTEM COMPONENT (ESC)

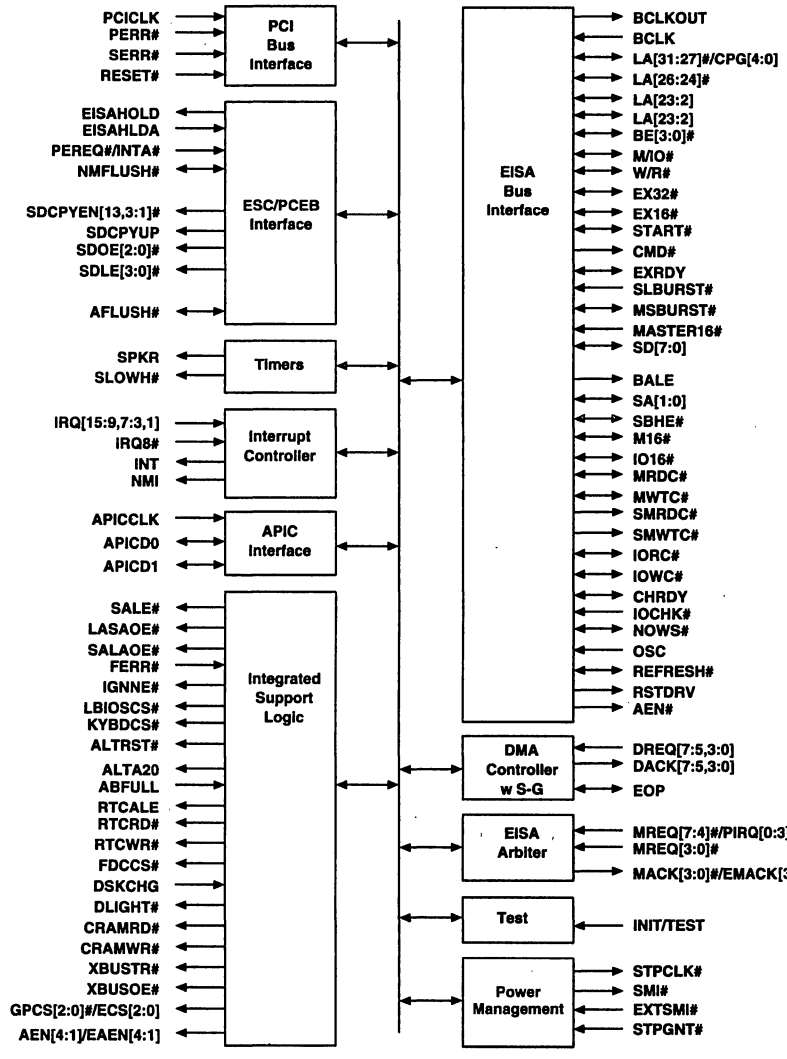
- Integrates EISA Compatible Bus Controller
  - Translates Cycles Between EISA and ISA Bus
  - Supports EISA Burst and Standard Cycles
  - Supports ISA Zero Wait-State Cycles
  - Supports Byte Assembly/ Disassembly for 8-, 16- and 32-Bit Transfers
  - Supports EISA Bus Frequency of up to 8.33 MHz
- Supports Eight EISA Slots
  - Directly Drives Address, Data and Control Signals for Eight Slots
  - Decodes Address for Eight Slot Specific AENs
- Provides Enhanced DMA Controller
  - Provides Scatter-Gather Function
  - Supports Type A, Type B, Type C (Burst), and Compatible DMA Transfer
  - Provides Seven Independently Programmable Channels
  - Integrates Two 82C37A Compatible DMA Controllers
- Integrates the Functionality of two 82C59 Interrupt Controllers and two 82C54 Timers
  - Provides 14 Programmable Channels for Edge or Level Interrupts
  - Provides 4 PCI Interrupts Routible to any of 11 Interrupt Channels
  - Supports Timer Function for Refresh Request, System Timer, Speaker Tone, Fail Safe Timer, and CPU Speed Control
- Advanced Programmable Interrupt Controller (APIC)
  - Multiprocessor Interrupt Management
  - Separate Bus For Interrupt Messages
- 5V CMOS Technology
- Provides High Performance Arbitration
  - Supports Eight EISA Masters and PCEB
  - Supports ISA Masters, DMA Channels, and Refresh
  - Provides Programmable Arbitration Scheme for Fixed, Rotating, or Combination Priority
- Integrates Support Logic for X-Bus Peripherals
  - Generates Chip Selects/Encoded Chip Selects for Floppy and Keyboard Controller, IDE, Parallel/ Serial Ports , and General Purpose Peripherals
  - Provides Interface for Real Time Clock
  - Generates Control Signals for X-Bus Data Transceiver
  - Integrates Port 92, Mouse Interrupt, and Coprocessor Error Reporting
- Generates Non-Maskable Interrupts (NMI)
  - PCI System Errors
  - PCI Parity Errors
  - EISA Bus Parity Errors
  - Fail Safe Timer
  - Bus Timeout
  - Via Software Control
- Provides BIOS Interface
  - Supports 512K Bytes of Flash or EPROM BIOS on the X-Bus
  - Allows BIOS on PCI
  - Supports Integrated VGA BIOS
- 82374SB System Power Management (Intel SMM Support)
  - Fast On/Off Support via SMI GenerationHardware Events, Software Events, EXTSMI#, Fast Off Timer, System Events
  - Programmable CPU Clock Control
  - Enables Energy Efficient Desktop Systems
- Only Available as Part of a Supported Kit
- 208-Pin QFP Package

November 1995
Order Number: 290476-004

This document describes both the 82374EB and 82374SB components. Unshaded areas describe the 82374EB. Shaded areas, like this one, describe the 82374SB operations that differ from the 82374EB.

The 82374EB/SB EISA System Component (ESC) provides all the EISA system compatible functions. The ESC with the PCEB provide all the functions to implement an EISA-to-PCI bridge and EISA I/O subsystem. The ESC integrates the common I/O functions found in today's EISA-based PC systems. The ESC incorporates the logic for an EISA (master and slave) interface, EISA bus controller, enhanced seven channel DMA controller with scatter-gather support, EISA arbitration, 14 channel interrupt controller, Advanced Programmable Interrupt Controller (APIC), five programmable timer/counters, and non-maskable-interrupt (NMI) control logic. The ESC also integrates support logic to decode peripheral devices such as the Flash BIOS, real time clock, keyboard/mouse controller, floppy controller, two serial ports, one parallel port, and IDE hard disk drive.

The 82374SB also contains support for SMM power management



290476-1

NOTE:
SMM Interface and INIT are on the 82347SB only.

Simplified ESC Block Diagram

82374EB/82374SB EISA SYSTEM COMPONENT (ESC)

CONTENTS

CONTENTS

1

CONTENTS

CONTENTS

CONTENTS

CONTENTS

1

CONTENTS

PAGE

1.0 ARCHITECTURAL OVERVIEW

The PCI-EISA bridge chip set provides an I/O subsystem core for the next generation of high-performance personal computers (e.g., those based on the Intel486™ or Pentium® processors). System designers can take advantage of the power of the PCI (Peripheral Component Interconnect) for the local I/O bus while maintaining access to the large base of EISA and ISA expansion cards, and corresponding software applications. Extensive buffering and buffer management within the PCI-EISA bridge ensures maximum efficiency in both bus environments.

The chip set consists of two components—the 82375EB/SB PCI-EISA Bridge (PCEB) and the 82374EB/SB EISA System Component (ESC). These components work in tandem to provide an EISA I/O subsystem interface for personal computer platforms based on the PCI standard. This section provides an overview of the PCI and EISA Bus hierarchy followed by an overview of the PCEB and ESC components.

Bus Hierarchy—Concurrent Operations:

Figure 1 shows a block diagram of a typical system using the PCI-EISA Bridge chip set. The system contains three levels of buses structured in the following hierarchy:

- Host Bus as the execution bus
- PCI Bus as a primary I/O bus
- EISA Bus as a secondary I/O bus

This bus hierarchy allows concurrency for simultaneous operations on all three bus environments. Data buffering permits concurrency for operations that cross over into another bus environment. For example, a PCI device could post data into the PCEB, permitting the PCI Local Bus transaction to complete in a minimum time and freeing up the PCI Local Bus for further transactions. The PCI device does not have to wait for the transfer to complete to its final destination. Meanwhile, any ongoing EISA Bus transactions are permitted to complete. The posted data is then transferred to its EISA Bus destination when the EISA Bus is available. The PCI-EISA Bridge chip set implements extensive buffering for PCI-to-EISA and EISA-to-PCI bus transactions. In addition to concurrency for the operations that cross bus environments, data buffering allows the fastest operations within a particular bus environment (via PCI burst transfers and EISA burst transfers).

The PCI Local Bus with 132 MByte/sec and EISA with 33 MByte/sec peak data transfer rate represent bus environments with significantly different bandwidths. Without buffering, transfers that cross the single bus environment are performed at the speed of the slower bus. Data buffers provide a mechanism for data rate adoption so that the operation of the fast bus environment (PCI), i.e. usable bandwidth, is not significantly impacted by the slower bus environment (EISA).

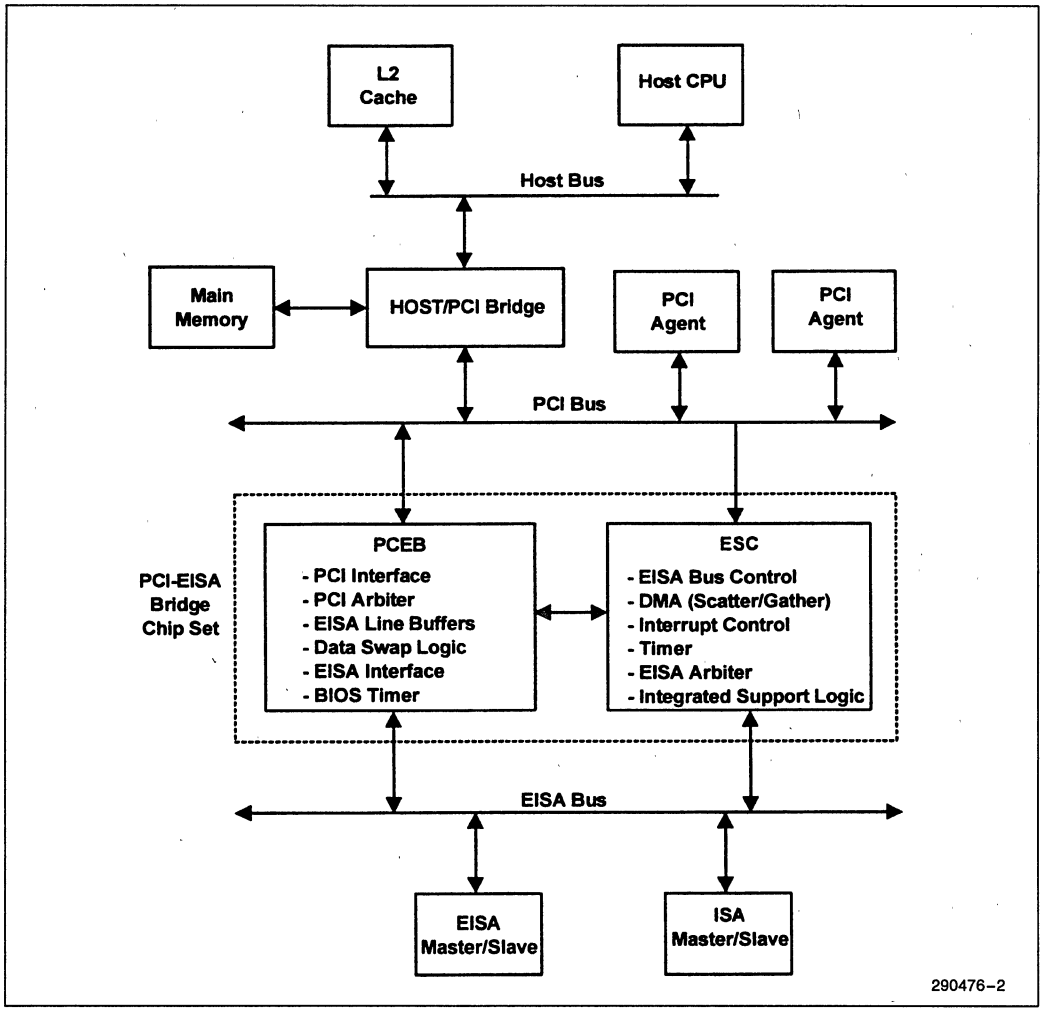


Figure 1. PCI-EISA Chip Set System Block Diagram

PCI Bus

The PCI Bus has been defined to address the growing industry needs for a standardized *local bus* that is not directly dependent on the speed and the size of the processor bus. New generations of personal computer system software such as Windows™ and Win-NT™ with sophisticated graphical interfaces, multi-tasking and multi-threading bring new requirements that traditional PC I/O architectures can not satisfy. In addition to the higher bandwidth, reliability and robustness of the I/O subsystem is becoming increasingly important. The PCI environment addresses these needs and provides an upgrade path for the future. PCI features include:

- Processor independent
- Multiplexed, burst mode operation
- Synchronous at frequencies from 20–33 MHz
- 120 MByte/sec usable throughput (132 MByte/sec peak) for 32 bit data path
- 240 MByte/sec usable throughput (264 MByte/sec peak) for 64 bit data path
- Optional 64 bit data path with operations that are transparent with the 32 bit data path
- Low latency random access (60 ns write access latency to slave registers from a master parked on the bus)
- Capable of full concurrency with processor/memory subsystem
- Full multi-master capability allowing any PCI master peer-to-peer access to any PCI slave
- Hidden (overlapped) central arbitration
- Low pin count for cost effective component packaging (address/data multiplexed)
- Address and data parity
- Three physical address spaces: memory, I/O, and configuration
- Comprehensive support for autoconfiguration through a defined set of standard configuration functions

System partitioning shown in Figure 1 illustrates how the PCI can be used as a common interface between different portions of a system platform that are typically supplied by the chip set vendor. These portions are the Host/PCI Bridge (including a main memory DRAM controller and an optional second level cache controller) and the PCI-EISA Bridge. Thus, the PCI allows a system I/O core design to be decoupled from the processor/memory treadmill, enabling the I/O core to provide maximum benefit over multiple generations of processor/memory technology. For this reason, the PCI-EISA Bridge can be used with different processors. Regardless of the new requirements imposed on the processor side of the Host/PCI Bridge (e.g. 64-bit data path, 3.3V interface, etc.) the PCI side remains unchanged which allows reusability not only of the rest of the platform chip set (i.e. PCI-EISA Bridge) but also of all other I/O functions interfaced at the PCI level. These functions typically include graphics, SCSI, and LAN.

EISA Bus

The EISA bus in the system shown in the Figure 1.0 represents a second level I/O bus. It allows personal computer platforms built around the PCI as a primary I/O bus to leverage the large EISA/ISA product base. Combinations of PCI and EISA buses, both of which can be used to provide expansion functions, will satisfy even the most demanding applications.

Along with compatibility with 16-bit and 8-bit ISA hardware and software, the EISA bus provides the following key features:

- 32-bit addressing and 32-bit data path
- 33 MByte/sec bus bandwidth
- Multiple bus master support through efficient arbitration
- Support for autoconfiguration

Integrated Bus Central Control Functions

The PCI-EISA Bridge chip set integrates central bus functions on both the PCI and EISA Buses. For the PCI Bus, the functions include PCI bus arbitration and default bus driver. For the EISA Bus, central functions include the EISA Bus controller and EISA arbiter are integrated in the ESC component and EISA Data Swap Logic is integrated in the PCEB.

Integrated System Functions

The PCI-EISA Bridge chip set integrates system functions including PCI parity and system errors reporting, buffer coherency management protocol, PCI and EISA memory and I/O address space mapping and decoding. For maximum flexibility all of these functions are programmable allowing for variety of optional features.

1.1 PCEB Overview

The PCEB provides the interface (bridge) between PCI and EISA buses by translating bus protocols in both directions. It uses extensive buffering on both the PCI and EISA interfaces to allow concurrent bus operations. The PCEB also implements the PCI central support functions (e.g., PCI arbitration, error signal support, and subtractive decoding). The major functions provided by the PCEB are described in this section.

PCI Bus Interface

The PCEB can be either a master or slave on the PCI Bus and supports bus frequencies from 25 MHz to 33 MHz. For PCI-initiated transfers, the PCEB can only be a slave. The PCEB becomes a slave when it positively decodes the cycle. The PCEB also becomes a slave for unclaimed cycles on the PCI Bus. These unclaimed cycles are either negatively or subtractively decoded by the PCEB and forwarded to the EISA Bus.

As a slave, the PCEB supports single cycle transfers for memory, I/O, and configuration operations and burst cycles for memory operations. Note that, burst transfers cannot be performed to the PCEB's internal registers. Burst memory write cycles to the EISA Bus can transfer up to four Dwords, depending on available space in the PCEB's Posted Write Buffers. When space is no longer available in the buffers, the PCEB terminates the transaction. This supports the Incremental Latency Mechanism as defined in the Peripheral Component Interconnect (PCI) Specification. Note that, if the Posted Write Buffers are disabled, PCI burst operations are not performed and all transfers are single cycle.

For EISA-initiated transfers to the PCI Bus, the PCEB is a PCI master. The PCEB permits EISA devices to access either PCI memory or I/O. While all PCI I/O transfers are single cycle, PCI memory cycles can be either single cycle or burst, depending on the status of the PCEB's Line Buffers. During EISA reads of PCI memory, The PCEB uses a burst read cycle of four Dwords to prefetch data into a Line Buffer. During EISA-to-PCI memory writes, the PCEB uses PCI burst cycles to flush the Line Buffers. The PCEB contains a programmable Master Latency Timer that provides the PCEB with a guaranteed time slice on the PCI Bus, after which it surrenders the bus.

As a master on the PCI Bus, the PCEB generates address and command signal (C/BE#) parity for read and write cycles, and data parity for write cycles. As a slave, the PCEB generates data parity for read cycles. Parity checking is not supported.

The PCEB, as a resource, can be locked by any PCI master. In the context of locked cycles, the entire PCEB subsystem (including the EISA Bus) is considered a single resource.

PCI Bus Arbitration

The PCI arbiter supports six PCI masters—The Host/PCI bridge, PCEB, and four other PCI masters. The arbiter can be programmed for twelve fixed priority schemes, a rotating scheme, or a combination of the fixed and rotating schemes. The arbiter can be programmed for bus parking that permits the Host/PCI Bridge default access to the PCI Bus when no other device is requesting service. The arbiter also contains an efficient PCI retry mechanism to minimize PCI Bus thrashing when the PCEB generates a retry. The arbiter can be disabled, if an external arbiter is used.

EISA Bus Interface

The PCEB contains a fully EISA-compatible master and slave interface. The PCEB directly drives eight EISA slots without external data or address buffering. The PCEB is only a master or slave on the EISA Bus for transfers between the EISA Bus and PCI Bus. For transfers contained to the EISA Bus, the PCEB is never a master or slave. However, the data swap logic contained in the PCEB is involved in these transfers, if data size translation is needed. The PCEB also provide support for I/O recovery.

EISA/ISA masters and DMA can access PCI memory or I/O. The PCEB only forwards EISA cycles to the PCI Bus if the address of the transfer matches one of the address ranges programmed into the PCEB for EISA-to-PCI positive decode. This includes the main memory segments used for generating MEMCS# from the EISA Bus, one of the four programmable memory regions, or one of the four programmable I/O regions. For EISA-initiated accesses to the PCI Bus, the PCEB is a slave on the EISA Bus. I/O accesses are always non-buffered and memory accesses can be either non-buffered or buffered via the Line Buffers. For buffered accesses, burst cycles are supported.

During PCI-initiated cycles to the EISA Bus, the PCEB is an EISA master. For memory write operations through the Posted Write Buffers, the PCEB uses EISA burst transfers, if supported by the slave, to flush the buffers. Otherwise, single cycle transfers are used. Single cycle transfers are used for all I/O cycles and memory reads.

PCI/EISA Address Decoding

The PCEB contains two address decoders—one to decode PCI-initiated cycles and the other to decode EISA-initiated cycles. The two decoders permit the PCI and EISA Buses to operate concurrently.

The PCEB can also be programmed to provide main memory address decoding on behalf of the Host/PCI bridge. When programmed, the PCEB monitors the PCI and EISA bus cycle addresses, and generates a memory chip select signal (MEMCS#) indicating that the current cycle is targeted to main memory residing behind the Host/PCI bridge. Programmable features include, read/write attributes for specific memory segments and the enabling/disabling of a memory hole. If MEMCS# is not used, this feature can be disabled.

In addition to the main memory address decoding, there are four programmable memory regions and four programmable I/O regions for EISA-initiated cycles. EISA/ISA master or DMA accesses to one of these regions are forwarded to the PCI Bus.

Data Buffering

To isolate the slower EISA Bus from the PCI Bus, the PCEB provides two types of data buffers. Buffer management control guarantees data coherency.

For EISA-initiated cycles to the PCI Bus, there are four 16-byte wide Line Buffers. These buffers permit prefetching of PCI memory read data and posting of PCI memory write data.

By using burst transactions to fill or flush these buffers, if appropriate, the PCEB maximizes bus efficiency. For example, an EISA device could fill a Line Buffer with byte, word, or Dword transfers and The PCEB would use a PCI burst cycle to flush the filled line to PCI memory.

BIOS Timer

The PCEB has a 16 bit BIOS Timer. The timer can be used by BIOS software to implement timing loops. The timer count rate is derived from the EISA clock (BCLK) and has an accuracy of ± 1 μs.

1.2 ESC Overview

The ESC implements system functions (e.g., timer/counter, DMA, and interrupt controller) and EISA subsystem control functions (e.g., EISA bus controller and EISA bus arbiter). The major functions provided by the ESC are described in this section.

EISA Controller

The ESC incorporates a 32-bit master and an 8-bit slave. The ESC directly drives eight EISA slots without external data or address buffering. EISA system clock (BCLK) generation is integrated by dividing the PCI clock (divide by 3 or divide by 4) and wait-state generation is provided. The AENx and MACKx signals provide a direct interface to four EISA slots and supports eight EISA slots with encoded AENx and MACKx signals.

The ESC contains an 8-bit data bus (lower 8 bits of the EISA data bus) that is used to program the ESC's internal registers. Note that for transfers between the PCI and EISA Buses, the PCEB provides the data path. Thus, the ESC does not require a full 32 bit data bus. A full 32-bit address bus is provided and is used during refresh cycles and for DMA operations.

The ESC performs cycle translation between the EISA Bus and ISA Bus. For mis-matched master/slave combinations, the ESC controls the data swap logic that is located in the PCEB. This control is provided through the PCEB/ESC interface.

DMA Controller

The ESC incorporates the functionality of two 82C37 DMA controllers with seven independently programmable channels. Each channel can be programmed for 8 or 16 bit DMA device size, and ISA-compatible, type "A", type "B", or type "C" timings. Full 32 bit addressing is provided. The DMA controller is also responsible for generating refresh cycles.

The DMA controller supports an enhanced feature called scatter/gather. This feature provides the capability of transferring multiple buffers between memory and I/O without CPU intervention. In scatter/gather mode, the DMA can read the memory address and word count from an array of buffer descriptors, located in main memory, called the scatter/gather descriptor (SGD) table. This allows the DMA controller to sustain DMA transfers until all of the buffers in the SGD table are handled.

Interrupt Controller

The ESC contains an EISA compatible interrupt controller that incorporates the functionality of two 82C59 Interrupt Controllers. The two interrupt controllers are cascaded providing 14 external and two internal interrupts.

Advanced Programmable Interrupt Controller (APIC)

In addition to the standard EISA compatible interrupt controller described above, the ESC incorporates the Advanced Programmable Interrupt Controller (APIC). While the standard interrupt controller is intended for use in a uni-processor system, APIC can be used in either a uni-processor or multi-processor system. APIC provides multi-processor interrupt management and incorporates both static and dynamic symmetric interrupt distribution across all processors. In systems with multiple I/O subsystems, each subsystem can have its own set of interrupts.

Timer/Counter

The ESC provides two 82C54 compatible timers (Timer 1 and Timer 2). The counters in Timer 1 support the system timer interrupt (IRQ0#), refresh request, and a speaker tone output (SPKR). The counters in Timer 2 support fail-safe timeout functions and the CPU speed control.

Integrated Support Logic

To minimize the chip count for board designs, the ESC incorporates a number of extended features. The ESC provides support for ALTA20 (Fast A20GATE) and ALTRST with I/O Port 92h. The ESC generates the control signals for SA address buffers and X-Bus buffer. The ESC also provides chip selects for BIOS, the keyboard controller, the floppy disk controller, and three general purpose devices. Support for generating chip selects with an external decoder is provided for IDE, a parallel port, and a serial port. The ESC provides support for a PC/AT compatible coprocessor interface and IRQ13 generation.

Power Management (82374SB)

Extensive power management capability permits a system to operate in a low power state without being powered down. Once in the low power state (called "Fast Off" state), the computer appears to be off. For example, the SMM code could turn off the CRT, line printer, hard disk drive's spindle motor, and fans. In addition, the CPU's clock can be governed. To the user, the machine appears to be in the off state. However, the system is actually in an extremely low power state that still permits the CPU to function and maintain communication connections normally associated with today's desktops (e.g., LAN, Modem, or FAX). Programmable options provide power management flexibility. For example, various system events can be programmed to place the system in the low power state or break events can be programmed to wake the system up.

2.0 SIGNAL DESCRIPTION

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

The "#" symbol at the end of a signal indicates that the active, or asserted state occurs when the signal is at a low voltage level. When "#" is not presented after the signal name, the signal is asserted when at the high voltage level.

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

The following notations are used to describe the signal type.

in Input is a standard input-only signal.

out Totem Pole Output is a standard active driver.

o/d Open Drain Input/Output.

t/s Tri-State is a bi-directional, tri-state input/output pin.

s/t/s Sustained Tri-State is an active low tri-state signal owned and driven by one and only one agent at a time. The agent that drives a s/t/s pin low must drive it high for at least one clock before letting it float. A new agent can not start driving a s/t/s signal any sooner than one clock after the previous owner tri-states it. A pull-up sustains the inactive state until another agent drives it and is provided by the central resource.

NOTE:

During a hard reset, INTR, NMI, IGNNE#, SMI# (on 82374SB), ALTRST#, STPCLK# (on 82374SB) and ALTA20 are driven low to prevent problems associated with 5V/3.3V power sequencing. Any outputs of the ESC that are directed to a 3.3V CPU must be driven through a 5V to 3.3V translator.

2.1 PCI Local Bus Interface Signals

Pin Name	Type	Description
PCICLK	in	**PCI CLOCK:** PCICLK provides timing for all transactions on the PCI bus. The ESC uses the PCI Clock (PCICLK) to generate EISA Bus Clock (BCLK). The PCICLK is divided by 3 or 4 to generate the BCLK. The EISA Bridge supports PCI Clock frequencies of 25 MHz through 33 MHz.
PERR#	in	**PARITY ERROR:** PERR# indicates a data parity error. PERR# may be pulsed active by any agent that detects an error condition. Upon sampling PERR# active, the ESC generates an NMI interrupt to the CPU.
SERR#	in	**SYSTEM ERROR:** SERR# may be pulsed active by any agent that detects an error condition. Upon sampling SERR# active, the ESC generates an NMI interrupt to the CPU.
RESET#	in	**SYSTEM RESET:** RESET# forces the entire ESC chip into a known state. All internal ESC state machines are reset and all registers are set to their default values. RESET# may be asynchronous to PCICLK when asserted or negated. Although asynchronous, negation must be a clean, bounce-free edge. The ESC uses RESET# to generate RSTDRV signal.

2.2 EISA Bus Interface Signals

Pin Name	Type	Description
BCLKOUT	out	**EISA BUS CLOCK OUTPUT**: BCLKOUT is typically buffered to create EISA Bus Clock (BCLK). The BCLK is the system clock used to synchronize events on the EISA/ISA bus. The BCLKOUT is generated by dividing the PCICLK. The ESC uses a divide by 3 or divide by 4 to generate the BCLKOUT.
BCLK	in	**EISA BUS CLOCK**: The ESC uses BCLK to synchronize events on the EISA bus. The ESC generates or samples all the EISA/ISA bus signals on either the rising or the falling edge of BCLK.

Pin Name	Type	Description
LA[31:27]#/ CPG[4:0]	t/s	**EISA ADDRESS BUS/CONFIGURATION RAM PAGE ADDRESS**: These are multiplexed signals. These signals behave as the EISA address bus under all conditions except during access cycle to the Configuration RAM. EISA Address Bus: LA[31:27]# are directly connected to the EISA address bus. The ESC uses the address bus in conjunction with the BE[3:0]# signals as inputs to decode accesses to its internal resources except in DMA and Refresh modes. During DMA and Refresh modes, these are outputs, and the ESC uses these signals in conjunction with BE[3:0]# to drive Memory address. Configuration Ram Page Address: CPG[4:0] are connected to Configuration SRAM address lines. During I/O access to 0800h-08FFh, the ESC drives these signals with the configuration page address (the value contained in register 0C00h). The Configuration RAM Page Address function can be disabled by setting Mode Select register bit 5 = 0.
LA[26:24]# and LA[23:2]	t/s	**EISA ADDRESS BUS**: These signals are directly connected to the EISA address bus. The ESC uses the address bus in conjunction with the BE[3:0]# signals as inputs to decode accesses to its internal resources except in DMA and Refresh modes. During DMA and Refresh modes, these are outputs, and the ESC uses these signals in conjunction with BE[3:0]# to drive Memory address.
BE[3:0]#	t/s	**BYTE ENABLES**: BE[3:0]# signals are directly connected to the EISA address bus. These signals indicate which byte on the 32-bit EISA data bus are involved in the current cycle. BE[3:0]# are inputs during EISA master cycles which do not require assembly/disassembly operation. For EISA master assembly/disassembly cycles, ISA master cycles, DMA, and Refresh cycles BE[3:0]# are outputs. BE0#: Corresponds to byte lane 0-SD[7:0] BE1#: Corresponds to byte lane 0-SD[15:8] BE2#: Corresponds to byte lane 0-SD[23:16] BE3#: Corresponds to byte lane 0-SD[31:24]
M/IO#	t/s	**MEMORY OR I/O CYCLE:** M/IO# signal is used to differentiate between memory cycles and I/O cycles on the EISA bus. A High value on this signal indicates a memory cycle, and a Low value indicates an I/O cycle. M/IO# is an input to the ESC during EISA master cycles, and M/IO# is an output during ISA, DMA, and ESC initiated Refresh cycles. M/IO# is floated during ISA master initiated Refresh cycles.
W/R#	t/s	**WRITE OR READ CYCLE:** W/R# signal is used to differentiate between write and read cycles on the EISA bus. A High value on this signal indicates a Write cycle , and a Low value indicates a Read cycle. W/R# is an input to the ESC during EISA master cycles, and W/R# is an output during ISA, DMA, and Refresh cycles.

Pin Name	Type	Description
EX32#	o/d	**EISA 32 BIT DEVICE DECODE**: EX32# signal is asserted by a 32-bit EISA slave device. EX32# assertion indicates that an EISA device has been selected as a slave, and the device has a 32-bit data bus size. The ESC uses this signal as an input as part of its slave decode to determine if data size translation and/or cycle translation is required. EX32# is an output of the ESC during the last portion of the mis-matched cycle. This is an indication to the backed-off EISA master that the data translation has been completed. The backed-off EISA master uses this signal to start driving the EISA bus again.
EX16#	o/d	**EISA 16-BIT DEVICE DECODE:** EX16# signal is asserted by a 16-bit EISA slave device. EX16# assertion indicates that an EISA device has been selected as a slave, and the device has a 16 bit data bus size. The ESC uses this signal as an input as part of its slave decode to determine if data size translation and/or cycle translation is required. EX16# is an output of the ESC during the last portion of the mis-matched cycle. This is an indication to the backed-off EISA master that the data translation has been completed. The backed-off EISA master uses this signal to start driving the EISA bus again.
START#	t/s	**START CYCLE:** START# signal provides timing control at the start of an EISA cycle. START# is asserted for one BCLK. START# is an input to the ESC during EISA master cycles except portions of the EISA master to mis-matched slave cycles where it becomes an output. During ISA, DMA, and Refresh cycles START# is an output.
CMD#	out	**COMMAND:** CMD# signal provides timing control within an EISA cycle. The ESC is a central resource of the CMD# signal, and the ESC generates CMD# during all EISA cycles. CMD# is asserted from the rising edge of BCLK simultaneously with the negation of START#, and remains asserted until the end of the cycle.
EXRDY	o/d	**EISA READY:** EXRDY signal is deasserted by EISA slave devices to add wait states to a cycle. EXRDY is an input to the ESC for EISA master cycles, ISA master cycles, and DMA cycles where an EISA slave has responded with EX32# or EX16# asserted. The ESC samples EXRDY on the falling edge of BCLK after CMD# is asserted (except during DMA compatible cycles). During DMA compatible cycles, EXRDY is sampled on the second falling edge of BCLK after CMD# is driven active. For all types of cycles if EXRDY is sampled inactive, the ESC keeps sampling it on every falling edge of BCLK#. EXRDY is an output for EISA master cycles decoded as accesses to the ESC internal registers. ESC forces EXRDY low for one BCLK at the start of a potential DMA burst write cycle to insure that the initial write data is held long enough to be sampled by the memory slave.
SLBURST#	in	**SLAVE BURST:** SLBURST# signal is asserted by an EISA slave to indicate that the device is capable of accepting EISA burst cycles. The ESC samples SLBURST# on the rising edge of BCLK at the end of START# for all EISA cycles. During DMA cycles, the ESC samples SLBURST# twice; once on the rising edge of BCLK at the beginning of START# and again on the rising edge of BCLK at the end of START#.

Pin Name	Type	Description
MSBURST#	t/s	**MASTER BURST:** MSBURST# signal is asserted by an EISA master to indicate EISA burst cycles. MSBURST# is asserted by an EISA master in response to an asserted SLBURST# signal. The ESC samples SLBURST# on the rising edge of BCLK that CMD# is asserted. If asserted, the ESC samples SLBURST# on all subsequent rising edges of BCLK until sampled negated. The ESC keeps CMD# asserted during Burst cycles. MSBURST# is an output during DMA burst cycles. The ESC drives MSBURST# active on the falling edge of BCLK, one half BCLK after SLBURST# is sampled active at the end of START#.
MASTER16#	in	**MASTER 16-BIT:** MASTER16# is asserted by a 16-bit EISA Bus master or an ISA Bus master device to indicate that it has control of the EISA Bus or ISA Bus. The ESC samples MASTER16# on the rising edge of BCLK that START# is asserted. If MASTER16# is sampled asserted, the ESC determines that a 16-bit EISA Bus master or an ISA Bus master owns the Bus. If MASTER16# is sampled negated at the first sampling point, the ESC will sample MASTER16# a second time on the rising edge of BCLK at the end of START#. If MASTER16# is sampled asserted here, the ESC determines that a 32-bit EISA Bus master has downshifted to a 16-bit Bus master, and thus, the ESC will disable the data size translation function.
SD[7:0]	t/s	**SYSTEM DATA:** SD[7:0] signals are directly connected to the System Data bus. The SD[7:0] pins are outputs during I/O reads when the ESC internal registers are being accessed and during interrupt acknowledge cycles. The SD[7:0] pins are input during I/O writes cycles when the ESC internal registers are being accessed.

2.3 ISA Bus Signals

Pin Name	Type	Description
BALE	out	**BUS ADDRESS LATCH ENABLE:** BALE signal is asserted by the ESC to indicate that a address (SA[19:0], LA[23:17]), AEN and SBHE# signal lines are valid. The LA[23:17] address lines are latched on the trailing edge of BALE. BALE remains active throughout DMA and ISA Master cycles and Refresh cycles.
SA[1:0]	t/s	**ISA ADDRESS BITS 0& 1:** SA[1:0] are the least significant bits of the ISA address bus. SA[1:0] are inputs to the ESC during ISA master cycles except during ISA master initiated Refresh cycles. The ESC uses the SA[1:0] in conjunction with SBHE# to generate BE[3:0]# on the EISA bus. The SA[1:0] are outputs of the ESC during EISA master cycles and DMA cycles. The ESC generates these from BE[3:0]#.
SBHE#	t/s	**ISA BYTE HIGH ENABLE:** SBHE# signal indicates that the high byte on the ISA data bus (SD[15:8]) is valid. SBHE# is an input to the ESC during ISA master cycles, except during ISA master initiated Refresh cycles. The ESC uses the SBHE# in conjunction with SA[1:0] to generate BE[3:0]# on the EISA bus. SBHE# is an output during EISA master and DMA cycles.

Pin Name	Type	Description
M16#	o/d	**MEMORY CHIP SELECT 16:** M16# is an input when the ESC component owns the ISA bus. M16# is an output when an external ISA bus Master owns the ISA bus. The ISA slave memory drives this signal Low if it is a 16-bit memory device. For ISA to EISA translation cycles, the ESC combinatorially asserts M16# if either EX32# or EX16# are asserted. This signal has an external pull-up resistor.
IO16#	o/d	**16 BIT I/O CHIP SELECT:** IO16# signal is used to indicate a 16-bit I/O bus cycle. This signal is asserted by the I/O devices to indicate that they support 16-bit I/O bus cycles. All I/O accesses to the ESC registers are run as 8-bit I/O bus cycles. This signal has an external pull-up resistor.
MRDC#	t/s	**MEMORY READ:** MRDC# signal indicates a read cycle to the ISA memory devices. MRDC# is the command to a memory slave that it may drive data onto the ISA data bus. MRDC# is an output when the ESC owns the ISA bus. MRDC# is an input when an external ISA Bus master owns the ISA Bus. This signal is driven by the ESC during refresh cycles.
MWTC#	t/s	**MEMORY WRITE:** MWTC# signal indicates a write cycle to the ISA memory devices. MWTC# is the command to a memory slave that it may latch data from the ISA data bus. MWTC# is an output when the ESC owns the ISA bus. MWTC# is an input when an ISA Bus master owns the ISA Bus.
SMRDC#	out	**SYSTEM MEMORY READ:** SMRDC# signal is asserted by the ESC to request a memory slave to drive data onto the data lines. SMRDC# indicates that the memory read cycle is for an address below the 1 MByte range on the ISA bus. This signal is also asserted during refresh cycles.
SMWTC#	out	**SYSTEM MEMORY WRITE:** SMWTC# signal is asserted by the ESC to request a memory slave to accept data from the data lines. SMWTC# indicates that the memory write cycle is for an address below the 1 MByte range.
IORC#	t/s	**I/O READ:** IORC# is the command to an ISA I/O slave device that it may drive data on to the data bus (SD[15:0]). The device must hold the data valid until after IORC# is negated. IORC# is an output when the ESC component owns the ISA bus. IORC# is an input when an ISA Bus master owns the ISA Bus.
IOWC#	t/s	**I/O WRITE:** IOWC# is the command to an ISA I/O slave device that it may latch data from the ISA data bus (SD[15:0]). IOWC# is an output when the ESC component owns the ISA Bus. IOWC# is an input when an ISA Bus master owns the ISA Bus.
CHRDY	o/d	**I/O CHANNEL READY:** CHRDY when asserted allows ISA Bus resources request additional time (wait-states) to complete the cycle. CHRDY is an input when the ESC owns the ISA Bus. CHRDY is an input to the ESC during compatible DMA cycles. CHRDY is an output during ISA Bus master cycles to PCI slave or ESC internal register. The ESC will ignores CHRDY for ISA-Bus master accessing an ISA-Bus slave.
IOCHK#	in	**I/O CHANNEL CHECK:** IOCHK# can be asserted by any resource on the ISA Bus. When asserted, it indicates that a parity or an uncorrectable error has occurred for a device or memory on the ISA Bus. A NMI will be generated to the CPU if enabled.

Pin Name	Type	Description
NOWS#	o/d	**ZERO WAIT STATES:** NOWS# indicates that an peripheral device wishes to execute a zero wait-state bus cycle (the normal default 16-bit ISA bus memory or I/O cycle is 3 BCLKS). When NOWS# is asserted, a 16-bit memory cycle will occur in two BCLKs and a 16-bit I/O cycle will occur in three BCLKs. When NOWS# is asserted by an 8-bit device the default 6 BCLKs cycle is shortened to 4 or 5 BCLKs. NOWS# is an input when the ESC performing bus translation cycles. NOWS# is an output when the ESC internal registers are accessed. If CHRDY and NOWS# are both asserted during the same clock then NOWS# will be ignored and wait-states will be added as a function of CHRDY (CHRDY has precedence over NOWS#).
OSC	in	**OSCILLATOR:** OSC is the 14.31818 MHz signal with 50% duty cycle. OSC is used by the ESC timers.
RSTDRV	out	**RESET DRIVE:** RSTDRV is asserted by the ESC. An asserted RSTDRV causes a hardware reset of the devices on the ISA Bus. RSTDRV is asserted whenever the RESET# input to the ESC is asserted.
REFRESH#	t/s	**REFRESH:** REFRESH# is used by the ESC as an output to indicate when a refresh cycle is in progress. It should be used to enable the SA[15:0] address to the row address inputs of all banks of dynamic memory on the ISA bus so that when MRDC# goes active, the entire expansion bus dynamic memory is refreshed. Memory slaves must not drive any data onto the bus during refresh and should not add wait states since this will affect the entire system throughput. As an output, this signal is driven directly onto the ISA bus. This signal is an output only when the ESC DMA Refresh is a master on the bus responding to an internally generated request for Refresh. Upon RESET this pin will tristate. Note that address lines [15:8] are driven during refresh, but the value is meaningless and is not used to refresh ISA bus memory. REFRESH# may asserted by an expansion bus adapter acting as a 16-bit ISA bus master.
AEN#	out	**ADDRESS ENABLE:** AEN# is driven high for Bus master cycles. AEN# is driven low for DMA cycles. and Refresh cycles. AEN# is used to disable I/O devices from responding to DMA and Refresh cycles. System designs which do not used the slots specific AENs (AEN[4:1]/EAEN[4:1]) provided by the ESC can use the AEN# signal to generate their own slot specific AENs.
AEN[4:1]/ EAEN[4:1]	out	**SLOT SPECIFIC ADDRESS ENABLE/ENCODED SLOT SPECIFIC ADDRESS ENABLE:** These pins have a slightly different function depending on the ESC configuration (Mode Select register bit 1 and bit 0). Slot Specific Address Enable: If the ESC is programmed to support 4 EISA slots, these signals function as Slot Specific Address Enables (AEN[4:1]). Encoded Slot Specific Address Enable: If the ESC has been programmed to support more than 4 EISA slots, then these signals behave as Encoded Address Enables (EAEN[4:1]). A discrete decoder is required to generate slot specific AENs. Refer to Section 5.8.1 AEN GENERATION for a detailed description of these signals.

2.4 DMA Signal Description

Pin Name	Type	Description
DREQ[7:5,3:0]	in	**DMA REQUEST:** DREQ signals are either used to request DMA service from the ESC or used to gain control of the ISA Bus by a ISA Bus master. The active level (high or low) is programmed in the Command registers. When the Command register bit 6 is programmed to 0, DREQ are asserted high, otherwise the DREQ are asserted low. All inactive to active edges of DREQ are assumed to be asynchronous. The request must remain asserted until the appropriate DACK is negated. At power-up and after RESET, these lines should be low (negated).
DACK# [7:5,3:0]	out	**DMA ACKNOWLEDGE:** DACK# indicate that a request for DMA service from the DMA subsystem has been recognized or that an ISA Bus master has been granted the bus. The level of the DACK lines when asserted may be programmed to be either high or low. This is accomplished by programming the DMA Command register. These lines should be used to decode the DMA slave device with the IORC# or IOWC# line to indicate selection. If used to signal acceptance of a bus master request, this signal indicates when it is legal to assert MASTER16#. If the DMA controller has been programmed for a timing mode other than compatible mode, and another device has requested the bus, and a 4 μs time has elapsed, DACK# will be negated and the transfer stopped before the transfer is complete. In this case, the transfer will be restarted at the next arbitration period in which the channel wins the bus. Upon reset these lines are negated.
EOP	t/s	**END OF PROCESS:** EOP pin acts in one of two modes, and it is directly connected to the TC line of the ISA Bus. In the first mode, EOP-In, the pin is an input and can be used by a DMA slave to stop a DMA transfer. In the second mode, TC-Out, it is used as a terminal count output by DMA slaves. An active pulse is generated when the byte counter reaches its last value. **EOP-In Mode:** During DMA, for all transfer types, the EOP pin is sampled by the ESC. If it is sampled asserted, the address bus is tristated and the transfer is terminated. **TC-Out Mode:** The EOP output will be asserted after a new address has been output if the byte count expires with that transfer. The EOP (TC) will stay asserted until AEN# is negated unless AEN is negated during an autoinitialization. EOP (TC) will be negated before AEN is negated during an autoinitialization. **Intout Mode:** In this mode the EOP signal has the same behavior as the Chaining Interrupt or the Scatter-Gather interrupt to the host processor (IRQ13). If a scatter-gather or chaining buffer is expired, EOP will go active on the falling edge of BCLK. Only the currently active channel's interrupt will be reflected on this pin. Other channel's with active interrupts pending will not affect the EOP pin. Whenever all the DMA channels are not in use, the EOP pin is kept in output mode and negated. After reset, the EOP pin is kept in output mode and negated.

2.5 EISA Arbitration Signals

Pin Name	Type	Description
MREQ[3:0]#	in	**MASTER REQUEST:** MREQ[3:0]# are slot specific signals used by EISA bus masters to request bus access. MREQ# once asserted, must remain asserted until the corresponding MACK# is asserted. The MREQ# is negated on the falling edge of BCLK slightly before the end of a master transfer. The LA[ ], BE[ ]#, M/IO#, and W/R# lines should be floated on or before the rising edge of BCLK after MREQ# is negated. The end of the last bus cycle is derived from CMD# in this case. The MREQ# signals are asserted on the falling edge of BCLK. MREQ# is always sampled on the rising edge of BCLK. MREQ# is synchronous with respect to BCLK. After asserting MREQ# , the corresponding master must not assert MREQ# until 1.5 BCLKs after CMD# is negated.
MREQ[7:4]#/ PIRQ[0:3]#	in	**MASTER REQUEST/PCI INTERRUPT REQUEST:** These pins behave in one of two modes depending on the state of the Mode Select Register bit 1 and bit 0. **Master Request:** MREQ# lines are slot specific signals used by EISA bus masters to request bus access. This signal behave in the same manner as MREQ[3:0]# signals. **PCI Interrupt Request:** PIRQ# are used to generate asynchronous interrupts to the CPU via the Programmable Interrupt Controller (82C59) integrated in the ESC. These signals are defined as level sensitive and are asserted low. The PIRQx# can be shared with PC compatible interrupts IRQ3:IRQ7, IRQ9:IRQ15. The PIRQx# Route Control Register determines which PCI interrupt is shared with which PC compatible interrupt.

Register Bit[1:0]	Pins MREQ7#/ PIRQ0#	MREQ6#/ PIRQ1#	MREQ5#/ PIRQ2#	MREQ4#/ PIRQ3#
00	PIRQ0#	PIRQ1#	PIRQ2#	PIRQ3#
01	PIRQ0#	PIRQ1#	MREQ5#	MREQ4#
10	PIRQ0#	MREQ6#	MREQ5#	MREQ4#
11	MREQ7#	MREQ6#	MREQ5#	MREQ4#

Pin Name	Type	Description
MACK[3:0]#/ EMACK[3:0]	out	**MASTER ACKNOWLEDGE:/ENCODED MASTER ACKNOWLEDGE:** These pins behave in one of two modes depending on the state of the Mode Select register bit 1 and bit 0. If the ESC is programmed to support 4 EISA slots, then these pins are used as MACK#. If the ESC is programmed to support more than 4 EISA slots, then these pins are used as EMACK# **Master Acknowledge:** The MACK[3:0]# signals are asserted from the rising edge of BCLK at which time the bus master may begin driving the LA[ ], BE[ ]#, M/IO#, and W/R# lines on the next falling edge of BCLK. MACK# will stay asserted until the rising edge of BCLK when MREQ# is sampled negated. MACK# is sampled by EISA Bus masters on the falling edge of BCLK. If another device has requested the bus, MACK# will be negated before MREQ# is negated. When MACK# is negated, the granted device has a maximum of 8 μs to negate MREQ# and begin a final bus cycle. The ESC may negate the MACK# signal a minimum of one BCLK after asserting it if another device (or refresh) is requesting the bus. Upon reset MACK# is negated. **Encoded Master Acknowledge:** EMACK# behaves like MACK#. The difference is that a discrete decoder is required to generate MACK# for the EISA Bus masters. Refer to Section 5.8.2 MACK Generation for details.

2.6 Timer Unit Signal

Pin Name	Type	Description
SPKR	out	**SPEAKER DRIVE:** SPKR is the output of Timer 1, Counter 2 and is "ANDed" with Port 061h bit 1 to provide Speaker Data Enable. This signal drives an external speaker driver device, which in turn drives the ISA system speaker. SPKR has a 24 mA drive capability. Upon reset, its output state is low.
SLOWH#	out	**SLOW DOWN CPU:** SLOWH# is the output of Timer 2, Counter 2. This counter is used to slow down the main CPU of its execution via the CPU's HOLD pin by pulse width modulation. The first read of I/O register in the 048h-04Bh range will enable SLOWH# signal to follow the output of the Timer 2, Counter 2. Upon reset, SLOWH# is negated. **Hardware Reset (Strapping Option)** During hardware reset this signal is an input and the level on the pin at the end of the reset sequence determines where BIOS resides. A high level indicates that BIOS resides on the X-Bus and a low level indicates that BIOS resides on the ISA Bus. The status is used by the ESC, to control the X-Bus transceivers during BIOS access. **NOTE:** For the 82374EB, this pin has an internal weak pull-up of approximately 8 KΩ. For proper configuration of the BIOS location during reset, a weak external pull-down resistor (approx. 500Ω) must be connected to this pin. An external pull-down resistor is not needed for the 82374SB.

2.7 Interrupt Controller Signals

Pin Name	Type	Description
IRQ[15:9], IRQ8#, IRQ[7:3,1]	in	**INTERRUPT REQUEST:** IRQ These signals provide both system board components and EISA bus I/O devices with a mechanism for asynchronously interrupting the CPU. The assertion mode of each interrupt can be programmed to be edge or level triggered. An asserted IRQ input must remain asserted until after the falling edge of INTA#. If the input is negated before this time, a DEFAULT IRQ7 will occur when the CPU acknowledges the interrupt. IRQ8# requires an external pull-up resistor (8 KΩ–10 KΩ).
INTR	out	**CPU INTERRUPT:** INTR is driven by the ESC to signal the CPU that an Interrupt request is pending and needs to be serviced. It is asynchronous with respect to BCLK or PCICLK and it is always an output. The interrupt controllers must be programmed following a reset to ensure that this pin takes on a known state. Upon reset the state of this pin is undefined.
NMI	out	**NON-MASKABLE INTERRUPT:** NMI is used to force a non-maskable interrupt to the CPU. The CPU registers an NMI when it detects a rising edge on NMI. NMI will remain active until a read from the CPU to the NMI register at port 061h is detected by the ESC. This signal is set to low upon reset.

2.8 APIC Bus Signals

Pin Name	Type	Description
APICCLK	in	**APIC BUS CLOCK:** APICCLK provides the timing reference for the APIC Bus. Changes on APICD[1:0]# are synchronous to the rising edge of APICCLK.
APICD[1:0]	od	**APIC DATA:** APICD1 and APICD0 are the APIC data bus signals. Interrupt messages are sent/received over this bus. APIC arbitration uses APICD1.

2.9 System Power Management Signals (82374SB Only)

Pin Name	Type	Description
STPCLK#	out	**STOP CLOCK:** STPCLK# is asserted by the ESC in response to one of many maskable hardware or software events. For 3.3V processors that are not 5V tolerant, STPCLK# is driven to the CPU STPCLK# pin through a 5V to 3.3V translator. When the CPU samples STPCLK# asserted it responds by stopping its internal clock. After a hard reset, this signal is negated.
SMI#	out	**SYSTEM MANAGEMENT INTERRUPT:** SMI# is asserted by the ESC in response to one of many maskable hardware or software events. For 3.3V processors that are not 5V tolerant, SMI# is driven to the CPU SMI# pin through a 5V to 3.3V translator The CPU recognizes the falling edge of SMI# as the highest priority interrupt in the system. The CPU responds by entering SMM (System Management Mode). SMI# is negated during and following reset. After a hard reset, this signal is negated.

Pin Name	Type	Description
EXTSMI#	in	**EXTERNAL SYSTEM MANAGEMENT INTERRUPT:** EXTSMI# is a falling edge triggered input to the ESC indicating that an external device is requesting the system to enter SMM mode. When enabled via the SMI Enable Register, a falling edge on EXTSMI# results in the assertion of the SMI# signal to the CPU. EXTSMI# is an asynchronous input to the ESC.
INIT/TEST	in	**INITIALIZE/TEST:** On the 82374SB, the function of this pin is selected by the value on the GPCS0# pin at reset. If GPCS0# is low, INIT is selected and if GPCS0# is high, TEST is selected. On the 82374EB, this pin only functions as the TEST pin. **INIT** INIT is connected to the INIT pin on the CPU and indicates to the ESC that a CPU soft reset is occuring. When asserted, the ESC ensures that STPCLK# is negated when the CPU comes out of the soft reset. The ESC also blocks SMI# generation when INIT is asserted. **TEST** For TEST signal description, see the TEST signal section.
STPGNT#	in	**STPCLK# GRANT:** When asserted, STPGNT# indicates to the ESC that a Stop grant PCI special cycle was recognized by the PCEB. The ESC may then negate the STPCLK# signal when the STPCLK# Timer expires.

2.10 ESC/PCEB Interface Signals

2.10.1 ARBITRATION AND INTERRUPT ACKNOWLEDGE CONTROL

Pin Name	Type	Description
EISAHOLD	out	**EISA HOLD:** EISAHOLD is used to request control of the EISA bus from its default owner, the PCEB. This signal is synchronous to PCICLK and is asserted when RESET# is asserted.
EISAHLDA	in	**EISA HOLD ACKNOWLEDGE:** EISAHLDA in used by the PCEB to inform the ESC that it has been granted ownership of EISA bus. This signal is synchronous to PCICLK.
PEREQ#/ INTA#	in	**PCI TO EISA REQUEST OR INTERRUPT ACKNOWLEDGE:** PEREQ#/INTA# is a dual function signal. The context of the signal pin is determined by the state of EISAHLDA signal. When EISAHLDA is deasserted this signal has the context of Interrupt Acknowledge i.e. if PEREQ#/INTA# is asserted it indicates to the ESC that current cycle on the EISA is an interrupt acknowledge. When EISAHLDA is asserted this signal has the context of PCI-to-EISA Request i.e. if PEREQ#/INTA# is asserted it indicates to the ESC that PCEB needs to obtain the ownership of the EISA bus on behalf of an PCI agent. This signal is synchronous to the PCICLK and it is driven inactive when RESET# is asserted.

2.10.2 PCEB BUFFER COHERENCY CONTROL

Pin Name	Type	Description
NMFLUSH#	t/s	**NEW MASTER FLUSH:** NMFLUSH# is a bi-directional signal which is used to provide handshake between PCEB and ESC to control flushing of system buffers on behalf of EISA masters. During an EISA bus ownership change, before ESC can grant the bus to the EISA master (or DMA) it must ensure that system buffers are flushed and buffers pointing (potentially) towards EISA subsystem are disabled. The ESC asserts NMFLUSH# signal for one PCI clock indicating the request for system buffer flushing. (After driving NMFLUSH# asserted for 1 PCI clock the ESC tri-states NMFLUSH# signal.) When PCEB samples NMFLUSH# asserted it starts immediately to drive NMFLUSH# asserted and initiates internal and external requests for buffer flushing. After all buffers have been flushed (indicated by the proper handshake signals), the PCEB negates NMFLUSH# for 1 PCI clock and stops driving it. When the ESC samples the signal deasserted that indicates that all system buffers are flushed, it grants EISA bus to an EISA master (or DMA). The ESC resumes responsibility of default NMFLUSH# driver and starts driving NMFLUSH# deasserted until the next time a new EISA master (or DMA) wins arbitration. This signal is synchronous with PCICLK and is negated by the ESC at reset.
AFLUSH#	t/s	**APIC FLUSH:** AFLUSH# is bi-directional signal between the PCEB and ESC that controls system buffer flushing on behalf of the APIC. After a reset the ESC negates AFLUSH# until the APIC is initialized and the first interrupt request is recognized.
SDCPYUP	out	**SYSTEM (DATA) COPY UP:** SDCPYUP is used to control the direction of the byte copy operation. A High on the signal indicates a COPY UP operation where the lower byte lower word of the SD data bus is copied on to the higher byte or higher word of the bus. A Low on the signal indicates a COPY DOWN operation where the higher byte(s) of the data bus are copied on to the lower byte(s) of the bus. The PCEB uses the signal to perform the actual data byte copy operation during mis-matched cycles.
SDOE[2:0]#	out	**SYSTEM DATA OUTPUT ENABLES:** SDOE# enable the SD data output of the PCEB Data Swap Buffers on to EISA bus. The ESC activates these signals only during mis-matched cycles. The PCEB uses these signal to enable the SD data buffers as follows: SDOE0#: Enables byte lane 0 SD[7:0] SDOE1#: Enables byte lane 1 SD[15:8] SDOE2#: Enables byte lane 2 SD[23:16] and byte lane 3 SD[31:24]
SDLE[3:0]#	out	**SYSTEM DATA LATCH ENABLES:** SDLE[3:0]# enable the latching of EISA data bus These signals are activated only during mis-matched cycles except PCEB initiated write cycle. The PCEB uses these signals to latch the SD data bus as follows: SDLE0#: Latch byte lane 0 SD[7:0] SDLE1#: Latch byte lane 0 SD[15:8] SDLE2#: Latch byte lane 0 SD[23:16] SDLE3#: Latch byte lane 0 SD[31:24]

2.11 Integrated Logic Signals

2.11.1 EISA ADDRESS BUFFER CONTROL

Pin Name	Type	Description
SALE#	out	**SA LATCH ENABLE:** SALE# is directly connected to F543s which buffer the LA addresses from the SA addresses. The rising edge of SALE# latches the LA address bit LA[19:2] to the SA address bit SA[19:2].
LASAOE#	out	**LA TO SA ADDRESS OUTPUT ENABLE:** LASAOE# is directly connected to the SA output buffer enables of the F543s. The ESC asserts LASAOE# during EISA master cycles. When LASAOE# is asserted, the LA to SA output buffers of the F543s are enabled.
SALAOE#	out	**SA TO LA ADDRESS OUTPUT ENABLE:** SALAOE# is connected to the LA output buffer enables of the F543s. This signal functionally is the exact opposite of LASAOE# signals. The ESC asserts SALAOE# during ISA master cycles. When LASAOE# is asserted, the SA to LA output buffers of the F543s are enabled.

2.11.2 COPROCESSOR INTERFACE

Pin Name	Type	Description
FERR#	in	**NUMERIC CO-PROCESSOR ERROR:** FERR# signal is tied to the Co-processor error signal of the CPU. If FERR# is asserted (Co-processor error detected by the CPU), an internal IRQ13 is be generated and the INTR from the ESC will be asserted.
IGNNE#	out	**IGNORE NUMERIC ERROR:** IGNNE# is tied to the ignore numeric error pin of the CPU. IGNNE# is asserted and internal IRQ13 is negated from the falling edge of IOWC# during an I/O write to location 00F0h. IGNNE# will remain asserted until FERR# is negated. During reset, this signal is driven low.

2.11.3 BIOS INTERFACE

Pin Name	Type	Description
LBIOSCS#	out	**LATCHED BIOS CHIP-SELECT:** LBIOSCS# indicates the that the current address is for the system BIOS. The ESC generates this signal by decoding the EISA LA addresses. The ESC uses a transparent latch to latch the decoded signal. The LBIOSCS# is latched on the falling edge of BALE and qualified with REFRESH#.

2.11.4 KEYBOARD CONTROLLER INTERFACE

Pin Name	Type	Description
KYBDCS#	out	**KEYBOARD CHIP SELECT:** KYBDCS# is connected to the chip select of the 82C42. KYBDCS# is active for I/O addresses 0060h and 0064h.
ALTRST#	out	**ALTERNATE RESET:** ALTRST# is used to reset the CPU under program control. This signal is AND'ed together externally with the reset signal (RSTAR#) from the keyboard controller to provide a software means of resetting the CPU. This provides a faster means of reset than is provided by the Keyboard controller. Writing a 1 to bit 0 in the Port 92 register will cause this signal to pulse active (low) for approximately 4 BCLK's. Before another ALTRST# pulse can be generated, bit 0 must be written back to a 0. During reset, this signal is driven low.
ALTA20	out	**ALTERNATE A20:** ALTA20 is used to force A20M# to the CPU low for support of real mode compatible software. This signal is externally OR'ed with the ALTA20 signal from the Keyboard controller and CPURST to control the A20M# input of the CPU. Writing a "0" to bit 1 of Port 92h Register will force ALTA20 inactive (low). This in turn will drive A20M# to the CPU low, if A20GATE from the keyboard controller is also low. Writing a "1" to bit 1 of the Port 92h Register will force ALTA20 active (high), which in turn will drive A20M# to the CPU high, regardless of the state of ALTA20 from the keyboard controller. Upon reset, this signal is driven low.
ABFULL	in	**AUXILIARY BUFFER FULL:** ABFULL is tied directly to the ABFULL signal on the keyboard controller on the system board. This signal indicates that the keyboard controller auxiliary buffer for the mouse interface is full. See the CLKDIV Register desciption for programming the ABFULL function. If this function is not used, ABFULL should be tied low through a 1K resistor.

2.11.5 REAL TIME CLOCK INTERFACE

Pin Name	Type	Description
RTCALE	out	**REAL TIME CLOCK ADDRESS LATCH ENABLE:** RTCALE is directly connected to the system Real Time Clock. The RTC uses this signal to latch the appropriate memory address. A write to port 070h with the appropriate Real Time Clock memory address that will be written to or read from will cause RTCALE to go active.
RTCRD#/ PIRQ3#	out	**REAL TIME CLOCK READ COMMAND/PCI INTERRUPT REQUEST 3:** This signal pin has two functions and the function is selected via the Mode Select Register. When functioning as RTCRD#, this signal is asserted for I/O reads from address 0071h. If the Power On Password protection is enabled (I/O Port 92h bit 3 = 1), then for accesses to RTC addresses 36h–3Fh (Port 70h), RTCRD# will not be asserted. For details on PIRQ3#, see the Mode Select Register description. For the PIRQ3# function, an external pull-up resistor (10-20 K) must be added to this signal.

Pin Name	Type	Description
RTCWR#/ PIRQ2#	out	**REAL TIME CLOCK WRITE COMMAND/PCI INTERRUPT REQUEST 2:** This signal pin has two functions, and the function is selected via the Mode Select Register. When functioning as RTCWR#, this signal is asserted for I/O writes to address 0071h. If the Power On Password protection is enabled (I/O Port 92h bit 3 = 1) then for accesses to RTC addresses 36h–3Fh (Port 70h) RTCWR# will not be generated. For details on PIRQ2#, see the Mode Select Register description. For the PIRQ2# function, an external pull-up resistor (10K–20K) must be added to this signal.

2.11.6 FLOPPY DISK CONTROLLER INTERFACE

Pin Name	Type	Description
FDCCS#/ PIRQ1#	out	**FLOPPY DISK CONTROLLER CHIP SELECT/PCI INTERRUPT REQUEST 1:** This signal has two functions and the function is selected via the Mode Select Register. As FDCCS# is asserted for I/O cycles to the floppy drive controller. When functioning as FDCCS#, this signal is also asserted when IDECS1# is decoded. See the Mode Select Register description for details on the PIRQ1# function of this signal. Note that for the PIRQ1# function, an external pull-up resistor (10 KΩ–20 KΩ) must be added to this signal.
DSKCHG	in	**DISK CHANGE:** DSKCHG signal is tied directly to the DSKCHG signal of the floppy controller. This signal is inverted and driven onto system data line 7 (SD7) during I/O read cycles to floppy address locations 3F7h (primary) or 377h (secondary) as indicated by the table below. Note that the primary and secondary locations are programmed in the X-Bus Address Decode Enable/Disable Register "A".

FDCCS# Decode	IDECSx# Decode	State of SD7 (output)	State of XBUSOE#
Enabled	Enabled	Tri-stated	Enabled
Enabled	Disabled	Driven via DSKCHG	Disabled
Disabled	Enabled	Tri-stated	Disabled (note)
Disabled	Disabled	Tri-stated	Disabled

NOTE:

This mode is not supported because of potential contention between the X-Bus buffer and a floppy on the ISA bus driving the system bus at the same time during shared I/O accesses.

This signal is also used to determine if the floppy controller is present on the X-Bus. It is sampled on the trailing edge of RESET, and If high, the Floppy is present. For systems that do not support a Floppy via the ESC, this pin should strapped low. If sampled low, the SD7 function, and XBUSOE# will not be enable for accesses to the floppy disk controller.

Pin Name	Type	Description
DLIGHT#/ PIRQ0#	out	**FIXED DISK ACTIVITY LIght/PCI INTERRUPT REQUEST 0:** This signal has two functions, depending on the programming of the Mode Select Register. As DLIGHT#, this signal controls the fixed disk X light. When low, the light is on. When high, the light is off. If either bit 6 or bit 7 of the Port 92 register is set to a 1 (bit 6 and 7 are internally NOR'ed together), DLIGHT# is driven active (low). Setting both bits 6 and 7 low will cause DLIGHT# to be driven high. For the PIRQ0# function, see the Mode Select Register description. Note that for the PIRQ0# function, an external pull-up resistor (10 KΩ–20 KΩ) must be added to this signal.

2.11.7 CONFIGURATION RAM INTERFACE

Pin Name	Type	Description
CRAMRD#	out	**CONFIGURATION RAM READ COMMAND:** CRAMRD# is connected directly to the system Configuration RAM. The ESC asserts CRAMRD# for I/O reads from the address range programmed into the low and high bytes of the configuration RAM command registers.
CRAMWR#	out	**CONFIGURATION RAM WRITE COMMAND:** This is an active Low output. CRAMWR# is connected directly to the system Configuration RAM. The ESC activates CRAMWR# for I/O writes to the address range programmed into the low and high bytes of the configuration RAM command registers.

2.11.8 X-BUS CONTROL AND GENERAL PURPOSE DECODE

Pin Name	Type	Description
XBUST/R#	out	**X-BUS DATA TRANSMIT/RECEIVE:** XBUST/R# is tied directly to the direction control of a 74F245 that buffers the X-Bus data, XD(7:0), from the system data bus, SD(7:0). XBUST/R# is driven high (transmit) during I/O and memory reads for EISA and ISA masters. For DMA cycles (channel 2 only), XBUST/R# is driven high for the following cases: 1. Memory read, I/O write cycles where LBIOSCS# is asserted. 2. I/O read, memory write cycles where Digital Output Register bit 3 is set to 1. XBUST/R# is driven low (receive) under all other conditions.

Pin Name	Type	Description
XBUSOE#	out	**X-BUS DATA OUTPUT ENABLE:** XBUSOE# is tied directly to the output enable of a 74F245 that buffers the X-Bus data, XD(7:0), from the system data bus, SD(7:0). For EISA and ISA master memory read or write cycles, XBUSOE# is asserted when LBIOSCS# is asserted. Otherwise, XBUSOE# is not asserted. For EISA and ISA master I/O read or write cycles, SBUSOE# is asserted if an ISC supported X-Bus device has been decoded, and the decoding for that device has been enabled via the proper configuration registers. An exception to this is during an I/O read access to floppy location 3F7h (primary) or 377h (secondary) if the IDE decode space is disabled (i.e., IDE is not present on the X-Bus). In this case, XBUSOE# is not asserted. XBUSOE# is also not asserted during an I/O access to the floppy controller if DSKCHG is sampled low at reset. XBUSOE# is not asserted during DMA cycles, except for channel 2 DMA. For channel 2 DMA, XBUSOE# is asserted.
GPCS[2:0]#/ ECS[2:0]	out	**GENERAL PURPOSE CHIP SELECT/ENCODED CHIP SELECT:** These are dual function signals. The function of these pins is selected through the Mode Select Register bit 4. **General Purpose Chip Select:** GPCS[2:0]# are chip selects for peripheral devices. The peripheral devices can be mapped in the I/O range by programming the General Purpose Chip Select Base Address registers and General Purpose Mask registers (offset 64h-6Eh). **Encoded Chip Select:** ECS[2:0] provide encoded chip select decoding for serial ports, parallel port, IDE and general purpose devices. The device chip selects for the peripheral devices are generated by using a F138 with ECS[2:0] as inputs. **Hardware Reset (Test Mode)** **82374SB:** During Reset, GPCS0/ECS0 is an input signal. The level of this signal is sampled at the end of the reset sequence to determine whether the TEST pin is used as the current TEST function (sampled "1") or as the INIT signal (sampled "0"). After reset, the existing GPCS/ECS functionality on this pin is maintained. Note that an internal pull-up of approximately 8 KΩ is included on this pin. If the INIT mode on the TEST pin is to be selected, an external pull-down of approximately 500Ω should be connected to the pin.

2.12 Test Signal

Pin Name	Type	Description
INIT/TEST	in	**TEST:** On the 82374EB, this pin only functions as a TEST pin. On the 82374SB, the function of this pin is selected by the value on the GPCS0# pin at reset. If GPCS0# is low, INIT is selected and if GPCS0# is high, TEST is selected. **INIT** For INIT signal description, see the Power Management Signal section. **TEST** TEST is used to tri-state all of the outputs. For normal operations, this signal should be tied to V_{CC}. For test mode, this pin should be tied to ground.

3.0 REGISTER DESCRIPTION

The ESC contains ESC configuration registers, DMA registers, Timer Unit registers, Interrupt Unit registers, and EISA configuration registers. All of the registers are accessible from the EISA bus. During a reset the ESC sets its internal registers to predetermined **default** states. The default values are indicated in the individual register descriptions.

The following notation is used to describe register access attributes:

RO **Read Only**. If a register is read only, writes have no effect.

WO **Write Only**. If a register is write only, reads have no effect.

R/W **Read/Write**. A register with this attribute can be read and written. Note that individual bits in some read/write registers may be read only.

3.1 Configuration Registers

The ESC's configuration registers are accessed through an indexing scheme. The index address register is located at I/O address 0022h, and the index data register is located at I/O address 0023h. The offset (data) written into the index address register selects the desired configuration register. Data for the selected configuration register can be read from or written to by performing a read or a write to the index data register. See the Address Decode section for a summary of configuration register index addresses.

Some of the ESC registers described in this section contain reserved bits. These bits are labeled "R". Software must deal correctly with fields that are reserved. On reads, software must use appropriate masks to extract the defined bits and not rely on reserved bits being any particular value. On writes, software must ensure that the values of reserved bit positions are preserved. That is, the values of reserved bit positions must first be read, merged with the new values for other bit positions and then written back.

In addition to reserved bits within a register, the ESC's configuration space contains address locations that are marked "Reserved" (See Address Decode Section). The ESC responds to accesses to these address locations by completing the Host cycle. When a reserved register location is read, 0000h is returned. Writes to reserved registers have no effect on the ESC.

3.1.1 ESCID—ESC ID REGISTER

Address Offset: 02h
Default Value: 00h
Attribute: Read/Write
Size: 8 Bits

Since the ESC configuration registers are accessed by the index addressing mechanism using I/O Ports 22h, and 23h, it is possible that another device in the system might use the same approach for configuration. In order to avoid contention with similar index register devices, the ID register must be written with 0Fh. The ESC will not respond to accesses to any other configuration register until the ID byte has been written in the ESC ID Register.

Bit	Description
7:0	**ESC ID Byte:** These bits must be written to a value of 0Fh before the ESC will respond to any other configuration register access. After a reset has occurred all of the configuration registers, except this register, are disabled.

3.1.2 RID—REVISION ID REGISTER

Address Offset: 08h
Default Value: 02h (82374EB: A-2 stepping)
03h (82374SB: B-0 stepping)
Attribute: Read only
Size: 8 Bits

This 8-bit register contains device stepping information. Writes to this register have no effect.

Bit	Description
7:0	**Revision ID Byte:** These bits contain the stepping information about the device. The register is hardwired to the default value during manufacturing. The register is read only. Writes have no effect on the register value.

3.1.3 MS—MODE SELECT REGISTER

Address Offset: 40h
Default Value: 20h
Attribute: Read/Write
Size: 8 Bits

This register selects the various functional modes of the ESC.

Bit	Description
7	**Reserved**
6	**MREQ[7:4]#/PIRQ[3:0]# Enable:** This bit enables the selected (MREQ[7:4]#/PIRQ[3:0]# functionality. 1 = Enabled; 0 = Disabled
5	**Configuration RAM Address:** This bit is used to enable or disable the configuration RAM Page Address (CPG[4:0]) generation. If this bit is set to 1, accesses to the configuration RAM space will generate the RAM page address on the LA[31:27]# pins. If this bit is set to 0, the CPG[4:0] signals will not be activated. The default for this bit is 1.

Pin Name	Description
4	**General Purpose Chip Selects:** This bit is used to select the functionality of the GPCS[2:0]#/ ECS[2:0] pins. If the bit is set to 0, the GPCS[2:0]# functionality is selected. If the bit is set to 1, the ESC[2:0] functionality is selected.
3	**System Error:** This bit is used to disable (0) or enable (1) the generation of NMI based on SERR# signal pulsing active. When this bit = 1 (and NMIs are enabled via the NMIERTC Register) and SERR# is asserted, the NMI signal is asserted. When this bit = 0, the NMI signal is negated and SERR# is disabled from generating an NMI. Note that other NMI sources are enabled/disabled via the NMISC Register.
2:0	**PIRQx Mux/Mapping Control:** These bits select muxing/mapping of PIRQ[3:0]# with MREQ[7:4] and group of X-Bus signals (DLIGHT#, RTCWR#, RTCRD#). Different bit combinations select the number of EISA slots or group of X-Bus signals which can be supported with the certain number of PIRQx# signals by determining the functionality of pins AEN[4:1]/EAEN[4:1], MACK[3:0]#/EMACK[3:0]#, MREQ[7:4]#/PIRQ[3:0]#, DLIGHT#/ PIRQ0#, FDCCS#/PIRQ1#, RTCWR#/PIRQ2#, RTCRD#/PIRQ3# as shown in Table 1.

Table 1. Mode Select Register

Bits [2:0]	Signal Function						
	AEN[4:1]/ EAEN[4:1]#	**MACK[3:0]#/ EMACK[3:0]#**	**MREQ[7:4#/ PIRQ[0:3]#**	**DLIGHT#/ PIRQ0#**	**FDDCS#/ PIRQ1#**	**RTCWR#/ PIRQ2#**	**RTCRD#/ PIRQ3#**
000	EAEN[4:1]#	EMACK[3:0]#	MREQ[7:4#	PIRQ0#	PIRQ1#	PIRQ2#	PIRQ3#
001	EAEN[4:1]#	EMACK[3:0]#	MREQ[7:4#	PIRQ0#	PIRQ1#	RTCWR#	RTCRD#
010	EAEN[4:1]#	EMACK[3:0]#	MREQ[7:4#	PIRQ0#	FDDCS#	RTCWR#	RTCRD#
011	EAEN[4:1]#	EMACK[3:0]#	MREQ[7:4#	DLIGHT#	FDDCS#	RTCWR#	RTCRD#
100	AEN[4:1]	MACK[3:0]#	PIRQ[0:3]#	DLIGHT#	FDDCS#	RTCWR#	RTCRD#
101	EAEN[4:1]#	EMACK[3:0]#	PIRQ0#, PIRQ1#, MREQ5#, MREQ4#	DLIGHT#	FDDCS#	RTCWR#	RTCRD#
110	EAEN[4:1]#	EMACK[3:0]#	PIRQ0#, MREQ6#, MREQ5#, MREQ4#	DLIGHT#	FDDCS#	RTCWR#	RTCRD#
111	EAEN[4:1]#	EMACK[3:0]#	MREQ[7:4#	DLIGHT#	FDDCS#	RTCWR#	RTCRD#

3.1.4 BIOSCSA—BIOS CHIP SELECT A REGISTER

Address Offset: 42h
Default Value: 10h
Attribute: Read/Write
Size: 8 Bits

The LBIOSCS# signal is used to decode access to the motherboard BIOS. The ESC decodes memory access to the following address ranges, and if the range has been enabled the LBIOSCS# signal is always asserted for memory reads in the enabled BIOS range. If the BIOS Write Enable bit is set in the configuration register BIOSCSB, the LBIOSCS# is also asserted for memory write cycles.

Bit	Description
7:6	**Reserved**
5	**Enlarged BIOS:** During Memory access to locations FFF80000h–FFFDFFFFh with this bit set, LBIOSCS# will be asserted for memory read cycles. If bit 3 of BIOSCSB is set, then LBIOSCS# will be asserted for write cycles as well.
4	**High BIOS:** During Memory access to locations 0F0000h–0FFFFFh, FF0000h–FFFFFFh, FFFF0000h–FFFFFFFFh with this bit set, LBIOSCS# will be asserted for memory read cycles. If bit 3 of BIOSCSB is set, then LBIOSCS# will be asserted for write cycles as well.
3	**Low BIOS 4:** During Memory access to locations 0EC000h–0EFFFFh, FFEEC000h–FFEEFFFFh, FFFEC000h–FFFEFFFFh with this bit set, LBIOSCS# will be asserted for memory read cycles. If bit 3 of BIOSCSB is set, then LBIOSCS# will be asserted for write cycles as well.
2	**Low BIOS 3:** During Memory access to locations 0E8000h–0EBFFFh, FFEE8000h–FFEEBFFFh, FFFE8000h–FFFEBFFFh with this bit set, LBIOSCS# will be asserted for memory read cycles. If bit 3 of BIOSCSB is set, then LBIOSCS# will be asserted for write cycles as well.
1	**Low BIOS 2:** During Memory access to locations 0E4000h–0E7FFFh, FFEE4000h–FFEE7FFFh, FFFE4000h–FFFE7FFFh with this bit set, LBIOSCS# will be asserted for memory read cycles. If bit 3 of BIOSCSB is set, then LBIOSCS# will be asserted for write cycles as well.
0	**Low BIOS 1:** During Memory access to locations 0E0000h–0E3FFFh, FFEE0000h–FFEE3FFFh, FFFE0000h–FFFE3FFFh with this bit set, LBIOSCS# will be asserted for memory read cycles. If bit 3 of BIOSCSB is set, then LBIOSCS# will be asserted for write cycles as well.

3.1.5 BIOSCSB—BIOS CHIP SELECT B REGISTER

Address Offset: 43h
Default Value: 00h
Attribute: Read/Write
Size: 8 Bits

The LBIOSCS# signal is used to decode access to the motherboard BIOS. The ESC decodes memory access to the following address ranges, and if the range has been enabled the LBIOSCS# signal is always asserted for memory reads in the enabled BIOS range. If the BIOS Write Enable bit is set in the configuration register BIOSCSB, the LBIOSCS# is also asserted for memory write cycles.

Bit	Description
7:4	**Reserved**
3	**BIOS Write Enable:** When enabled LBIOSCS# is asserted for memory read AND write cycles for addresses in the decoded and enabled BIOS range, otherwise LBIOSCS# is asserted for memory read cycles ONLY.
2	**16 Meg BIOS:** During Memory access to locations FF0000h–FFFFFFh with this bit set, LBIOSCS# will be asserted for memory read cycles. If bit 3 of BIOSCSB is set, then LBIOSCS# will be asserted for write cycles as well.
1	**High VGA BIOS:** During Memory access to locations 0C4000h–0C7FFFh with this bit set, LBIOSCS# will be asserted for memory read cycles. If bit 3 of BIOSCSB is set, then LBIOSCS# will be asserted for write cycles as well.
0	**Low VGA BIOS:** During Memory access to locations 0C0000h–0C3FFFh with this bit set, LBIOSCS# will be asserted for memory read cycles. If bit 3 of BIOSCSB is set, then LBIOSCS# will be asserted for write cycles as well.

3.1.6 CLKDIV—EISA CLOCK DIVISOR REGISTER

Address Offset: 4Dh
Default Value: xx001000b
Attribute: Read/Write
Size: 8 Bits

This register is used to select the integer value used to divide the PCI clock (PCICLK) to generate the EISA Bus Clock (BCLK) and enable/disable the co-processor error support. In addition, for the 82374SB, the register controls the ABFULL and KBFULL functions.

Bit	Description
7:6	**Reserved**
5	**Co-processor Error:** The state of this bit determines if the FERR# signal is connected to the ESC internal IRQ13 interrupt signal. If this bit is set to 1, the ESC will assert IRQ13 to the interrupt controller if FERR# signal is asserted. If this bit is set to 0, then the FERR# signal is ignored by the ESC (i.e. this signal is not connected to any logic in the ESC).
4	**82374EB: Reserved** **82374SB: ABFULL (With IRQ12):** When bit 4 = 0, the internal IRQ12 is directed to the interrupt controller and transitions on ABFULL have no affect on this interrupt signal. When bit 4 = 1, the assertion of ABFULL is latched and directed to the internal IRQ12 signal in the following manner: • If the interrupt controller is programmed for edge detect mode on IRQ12, a low-to-high transition is generated on the internal IRQ12 signal. Transitions on the IRQ12 input pin are not reflected on the internal IRQ12 signal. • If the interrupt controller is programmed for level-sensitive mode, a high-to-low transition is generated on the internal IRQ12 signal. Transitions on the IRQ12 input pin are also reflected on the internal IRQ12 signal. The latching of the ABFULL signal is cleared by an I/O read of address 60h (no aliasing) or by a hard reset.
3	**82374EB: Reserved** **82374SB: Keyboard Full (KBFULL):** This bit selects the edge-detect KBFULL function on the IRQ1 input signal. When bit 3 = 0, IRQ1 is directed to the interrupt controller. When bit 3 = 1 (default), IRQ1 is latched and directed to the interrupt controller. The latched IRQ1 is cleared by an I/O read of address 60h (no aliasing) or by a hard reset.
2:0	**Clock Divisor:** These bits are used to select the integer that is used to divide the PCICLK down to generate the BCLK. Upon reset, these bits are set to 000b (divisor of 4). **Bit[2:0] — Divisor — BCLK** 000 — 4 (33.33 MHz) — 8.33 MHz 001 — 3 (25 MHz) — 8.33 MHz 010 — Reserved 011 — Reserved 1xx — Reserved

3.1.7 PCSA—PERIPHERAL CHIP SELECT A REGISTER

Address Offset: 4Eh
Default Value: x0000111b
Attribute: Read/Write
Size: 8 Bits

This register is used to enable or disable accesses to the RTC, keyboard controller, Floppy Disk controller, and IDE. Disabling any of these bits will prevent the chip select and X-Bus transceiver control signal (XBUSOE#) for that device from being generated. This register is also used to select which address range (primary or secondary) will be decoded for the resident floppy controller and IDE. It also allows control of where the keyboard controller is physically located (X-Bus or elsewhere).This insures that there is no contention with the X-Bus transceiver driving the system data bus during read accesses to these devices.

Bit	Description
7	**Reserved**
6	**Keyboard Controller Mapping:** 0 = keyboard controller mapped to the X-Bus (default). 1 = keyboard controller not mapped to the X-Bus. When bit 6 = 0, the keyboard controller encoded chip select signal and the X-Bus transceiver enable (XBUSOE#) are generated for accesses to address locations 60h (82374EB/SB), 62h (82374EB only), 64h (82374EB/SB) and 66h (82374EB only). When bit 6 = 1, the keyboard controller chip select signals are generated for accesses to these address locations. However XBUSOE# is disabled. Bit 1 must be 1 for either value of this configuration bit to decode an access to locations 60h, 62h, 64h, or 66h.
5,3:2	**Floppy Disk and IDE, Floppy Disk Decodes:** Bits 2 and 3 are used to enable or disable the floppy locations as indicated. Bit 2 defaults to enabled (1) and bit 3 defaults to disabled (0) when a reset occurs. Bit 5 is used to select between the primary and secondary address range used by the Floppy Controller and the IDE. Only primary or only secondary can be programmed at any one time. This bit defaults to primary (0). The following table shows how these bits are used to select the floppy controller:

Address	Bit 2	Bit 3	Bit 5	DSKCHG	FDCCS#
X	X	X	X	0	1
3F0h,3F1h	X	1	0	1	0
3F2h–3F7h	1	X	0	1	0 (Note)
370h,371h	X	1	1	1	0
372h–37Fh	1	X	1	1	0 (Note)

NOTE:
If IDE decode is enabled, all accesses to locations 03F6h and 03F7h (primary) or 0376h and 0377h (secondary) will result in decode for IDECS1# (FDCCS# will not be generated). An external AND gate can be used to tie IDECS1# and FDCCS# together to insure that the floppy is enabled for these accesses.

Pin Name	Description
4	**IDE DECODE:** Bit 4 is used to enable or disable IDE locations 1F0h–1F7h (primary) or 170h–177h (secondary) and 3F6h,3F7h (primary) or 376h,377h (secondary). **82374EB:** When this bit is set to 0, the IDE encoded chip select signals and the X-Bus transceiver signal (XBUSOE#) are not generated for these addresses. **82374SB:** When this bit is set to 0, the IDE encoded chip select signals and the X-Bus transceiver signal (XBUSOE#) are not generated for addresses 1F0h–1F7h (primary) or 170h–177h (secondary) and 3F6h, or 376h. Note that read/write accesses to addresses 377h and 3F7h are not disabled and still generate XBUSOE#.
1	**KEYBOARD CONTROLLER DECODE:** Enables (1) or disables (0) the keyboard controller address locations 60h (82374EB/SB), 62h (82374EB only), 64h (82374EB/SB), and 66h (82374EB only). When this bit is set to 0, the keyboard controller encoded chip select signals and the X-Bus transceiver signal (XBUSOE#) are not generated for these locations. Note that the value of this bit affects control function (keyboard controlling mapping) provided by bit 6 of this register.
0	**Bit 0: REAL TIME CLOCK DECODE:** Enables (1) or disables (0) the RTC address locations 70h–77h. When this bit is set to 0, the RTC encoded chip select signals RTCALE, RTCRD, RTCWR#, and XBUSOE# signals are not generated for these addresses.

3.1.8 PCSB—PERIPHERAL CHIP SELECT B REGISTER

Address Offset: 4Fh
Default Value: CFh
Attribute: Read/Write
Size: 8 Bits

This register is used to enable or disable generation of the X-Bus transceiver signal (XBUSOE#) for accesses to the serial ports and parallel port locations. When disabled, the XBUSOE# signal for that device will not be generated.

Bit	Description
7	**CRAM Decode:** This bit is used to enable (1) or disable (0) I/O write accesses to location 0C00h and I/O read/write accesses to locations 0800h–08FFh. The configuration RAM read and write (CRAMRD#, CRAMWR#) strobes are valid for accesses to 0800h–08FFh.
6	**Port 92 Decode:** This bit is used to disable (0) access to Port 92. This bit defaults to enable (1) at PCIRST.
5:4	**Parallel Port Decode:** These bits are used to select which Parallel Port address range (LPT1, 2,or 3) is decoded. **Bits[5:4] Decode** 00 LPT1 (3BCh–3BFh) 01 LPT2 (378h–37Fh) 10 LPT3 (278h–27Fh) 11 Disabled

Pin Name	Description
3:2	**Serial Port B Address Decode:** If either COM1 or COM2 address ranges are selected, these bits default to disabled upon PCIRST. **Bits[3:2] Decode** 00 3F8h–3FFh (COM1) 01 2F8h–2FFh (COM2) 10 Reserved 11 Port A disabled
1:0	**Serial Port A Address Decode:** If either COM1 or COM2 address ranges are selected, these bits default to disabled upon PCIRST. **Bits[1:0] Decode** 00 3F8h–3FFh (COM1) 01 2F8h–2FFh (COM2) 10 Reserved 11 Port A disabled

3.1.9 EISAID[4:1]—EISA ID REGISTERS

Address Offset: 50h, 51h, 52h, 53h
Default Value: 00h, 00h, 00h, 00h
Attribute: Read/Write
Size: 8 Bits each

These 8 bit registers contain the EISA motherboard ID. The data in the register is reflected on the data bus for I/O cycles addressed to 0C80h–0C83h respectively.

Bit	Description
7:0	**EISA ID Byte:** These bits contain the EISA Motherboard ID information. On power up these bits default to 00h. These bit are written with the ID value during configuration. The value of these bits are reflected in I/O registers 0C80h–0C83h.

3.1.10 SGRBA—SCATTER/GATHER RELOCATE BASE ADDRESS REGISTER

Address Offset: 57h
Default Value: 04h
Attribute: Read/Write
Size: 8 Bits

The value programmed in this register determines the high order I/O address of the S-G registers. The default value is 04h.

Bit	Description
7:0	**S-G Relocate Byte:** These bits determine the I/O location of the Scatter Gather Registers. The Scatter-Gather register relocation range is xx10h–xx3Fh (default 0410h–043Fh). These bits determine the Byte 1 of the I/O address. Address signals LA[15:8] are compared against the contents of this register (bit[7:0]) to determine I/O accesses to the Scatter-Gather registers. The default on Power up is 04h.

3.1.11 APICBASE—APIC BASE ADDRESS RELOCATION

Address Offset: 59h
Default Value: 00h
Attribute: Read/Write
Size: 8 Bits

The APICBASE Register provides the modifier for the APIC base address.

82374EB:
APIC is mapped in the CPU memory space at the locations FEC0_x000h and FEC0_x010h (x=0-Fh). The value of "x" is defined by bits[5:2]. Thus, the relocation register provides a 4 KByte address granularity. The default value of 00h provides APIC unit mapping at the addresses FEC00000h and FEC00010h.

82374SB:
APIC is mapped in the CPU memory space at the locations FEC0_xy00h and FEC0_xy10h (x=0-Fh, y=0,4,8,Ch). The value of "y" is defined by bits[1:0] and value of "x" is defined by bits[5:2]. Thus, the relocation register provides a 1 KByte address granularity (i.e. potentially up to 64 I/O APICs can be uniformly addresses in the memory space). The default value of 00h provides APIC unit mapping at the addresses FEC00000h and FEC00010h.

Bit	Description
7:6	**Reserved**
5:2	**X-Base Address—R/W:** Bits[5:2] are compared to host address bits A[15:12], respectively.
1:0	**82374EB: Reserved** **82374SB: Y-Base Address—R/W:** Bits[1:0] are compared to host address bits A[11:10], respectively.

3.1.12 PIRQ[0:3]#—PIRQ ROUTE CONTROL REGISTERS

Address Offset: 60h, 61h, 62h, 63h
Default Value: 80h
Attribute: Read/Write
Size: 8 Bits

These registers control the routing of PCI Interrupts (PIRQ[0:3]#) to the PC compatible interrupts. Each PCI interrupt can be independently routed to 1 of 11 compatible interrupts.

Interrupt Steering Programming Considerations
When using the PCI programmable interrupt steering feature, the following programming considerations apply:

1. Any interrupt steered to by a PIRQx# must be programmed to level sensitive mode.
2. For an interrupt used as a PIRQx#, that IRQ pin is also level sensitive. It is not permissible to use an interrupt on the EISA/ISA Bus as edge triggered as well as on the PCI Bus as level sensitive.
3. Registers that must be programmed when using a PIRQx# include the Mode Select Registers, Edge Level Registers, PIRQ[3:0]# Route Control Registers, and the Interrupt Mask Registers (listed in suggested programming order)

Bit	Description
7	**Routing of Interrupts:** When enabled (0) this bit routes the PCI Interrupt signal to the PC compatible interrupt signal specified in bits[6:0]. After a reset or a power-on this bit is disabled (set to 1).
6:0	**IRQx# Routing Bits:** These bits specify which IRQ signal to generate when the PCI Interrupt for this register has been triggered. **Bits[6:0] IRQx# Bits[6:0] IRQx#** 0000000 Reserved 0001001 IRQ9 0000001 Reserved 0001010 IRQ10 0000010 Reserved 0001011 IRQ11 0000011 IRQ3 0001100 IRQ12 0000100 IRQ4 0001101 Reserved 0000101 IRQ5 0001110 IRQ14 0000110 IRQ6 0001111 IRQ15 0000111 IRQ7 0010000 to 0001000 Reserved 1111111 Reserved

3.1.13 GPCSLA[2:0]—GENERAL PURPOSE CHIP SELECT LOW ADDRESS REGISTER

Address Offset: 64h, 68h, 6Ch
Default Value: 00h
Attribute: Read/Write
Size: 8 Bits

This register contains the low byte of the General Purpose Peripheral mapping address. The contents of this register are compared with the LA[7:0] address lines. The contents of this register, the GPCSHA Register and the GPCSM Register control the generation the GPCS[2:0]# signal or the ESC[2:0] signal (101, 110 combination). If Mode Select Register (offset 40h) bit 4=1, offset register 6Ch is ignored.

Bit	Description
7:0	**GPCS Low Address Byte:** The contents of these bits are compared with the address lines LA[7:0] to generate the GPCS[2:0]# signal or the ECS[2:0] combination for this register. The mask register (GPCSM[2:0]) determines which bits to use during the comparison.

3.1.14 GPCSHA[2:0]—GENERAL PURPOSE CHIP SELECT HIGH ADDRESS REGISTER

Address Offset: 65h, 69h, 6Dh
Default Value: C0h
Attribute: Read/Write
Size: 8 Bits

This register contains the high byte of the General Purpose Peripheral mapping address. The contents of this register are compared with the LA[15:8] address lines. The contents of this register, the GPCSLA Register and the GPCSM Register control the generation the GPCS[2:0]# signal or the ESC[2:0] signal (101, 110 combination). If Mode Select Register (offset 40h) bit 4=1, offset register 6Dh is ignored.

Bit	Description
7:0	**GPCS High Address Byte:** The contents of these bits are compared with the address lines LA[15:8] to generate the GPCS[2:0]# signal or the ECS[2:0] combination for this register.

3.1.15 GPCSM[2:0]—GENERAL PURPOSE CHIP SELECT MASK REGISTER

Address Offset: 66h, 6Ah, 6Eh
Default Value: 00h
Attribute: Read/Write
Size: 8 Bits

This register contains the mask bits for determining the address range for which the GPCSx# signals are generated. If a register bit is set to a 1 then the corresponding bit in the GPCSL register is not compared with the address signal in the generation of the GPCSx# signals. If Mode Select Register (offset 40h) bit 4 = 1, offset register 6Eh is ignored.

Bit	Description
7:0	**GPCS Mask Register:** The contents of these bits are used to determine which bits to compare GPCSLA[2:0] with the address lines LA[7:0]. A 1 bit means the bit should not be compared.

3.1.16 GPXBC—GENERAL PURPOSE PERIPHERAL X-BUS CONTROL REGISTER

Address Offset: 6Fh
Default Value: xxxx x000b
Attribute: Read/Write
Size: 8 Bits

The register controls the generation of the X-BUS buffer output enable (XBUSOE#) signal for I/O accesses to the peripherals mapped in the General Purpose Chip Select address decode range. This register determines if the General Purpose Peripheral is placed on the XBUS or not. If the General Purpose Peripheral is on the X-Bus, then the corresponding bit is set to 1. Otherwise the bit is set to 0.

Bit	Description
7:3	**Reserved**
2	**XBUSOE# Generation for GPCS2#:** When this bit is enabled XBUSOE# will be generated when GPCS2# is generated; 1 = Enabled, 0 = Disabled.
1	**XBUSOE# Generation for GPCS1#:** When this bit is enabled XBUSOE# will be generated when GPCS1# is generated; 1 = Enabled, 0 = Disabled.
0	**XBUSOE# Generation for GPCS0#:** When this bit is enabled XBUSOE# will be generated when GPCS0# is generated;1 = Enabled, 0 = Disabled.

3.1.17 PAC—PCI/APIC CONTROL REGISTER

Address Offset: 70h
Default Value: 00h
Attribute: Read/Write
Size: 8 Bits

The PAC Register controls the operation of the INTR signal in APIC/PIC configuration and the routing of the System Management Interrupt (SMI).

Bit	Description
7:2	**Reserved**
1	**SMI Routing Control (SMIRC):** When SMIRC = 1, the SMI is routed via the APIC. When SMIRC = 0, the SMI is routed via the SMI# signal. Note that when SMRC = 1, INTR can not be routed through the APIC, since it is sharing the APIC interrupt input with SMI#.
0	**INTR Routing Control (INTRC):** When APIC is enabled (in mixed or pure APIC mode), this bit allows the ESC's external INTR signal to be masked (forces INTR to the inactive state but does not tri-states the signal). Thus, the CPU's INTR pin can be used (by providing a simple -gate) for the APIC Local Interrupt (LINTRx). However, INTR must not be masked via this bit when APIC is disabled and INTR is the only mechanism to signal the 8259 recognized interrupts to the CPU. When INTRC = 1, INTR is disabled (APIC must be enabled). When INTRC = 0, INTR is enabled.

3.1.18 TESTC—TEST CONTROL REGISTER

Address Offset: 88h
Default Value: 00h
Attribute: Read/Write
Size: 8 Bits

This register provides control for ESC manufacturing test modes. The functionality of this register is reserved.

3.1.19 SMICNTL—SMI CONTROL REGISTER

Address Offset: A0h
Default Value: 08h
Attribute: Read/Write
Size: 8 Bits

For the 82374SB, the SMICNTL Register provides Fast Off Timer control, STPCLK# enable/disable, and CPU clock scaling. This register also enables/disables the system management interrupt (SMI).

Bit	Description
7	**Reserved: Must be 0 when writing this register.**
6:4	**Reserved**
3	**Fast Off Timer Freeze (CTMRFRZ):** This field enables/disables the Fast Off Timer. When this bit is 1, the Fast Off timer stops counting. This prevents time-outs from occurring while executing SMM code. When this bit is 0, the Fast Off timer counts.

Pin Name	Description
2	**STPCLK# Scaling Enable (CSTPCLKSC):** This bit enables/disables control of the STPCLK# high/low times by the clock scaling timers. When bit 2 = 1, the STPCLK# signal scaling control is enabled. When enabled (and bit 1 = 1, enabling the STPCLK# signal), the high and low times for the STPCLK# signal are controlled by the Clock Scaling STPCLK# High Timer and Clock Scaling STPCLK# Low Timer Registers, respectively. When bit 2 = 0 (default), the scaling control of the STPCLK# signal is disabled.
1	**STPCLK# Signal Enable (CSTPCLKE):** This bit permits software to place the CPU into a low power state. When bit 1 = 1, the STPCLK# signal is enabled and a read from the APMC Register causes STPCLK# to be asserted. When bit 1 = 0 (default), the STPCLK# signal is disabled and is negated (high). Software can set this bit to 0 by writing a 0 to it or by any write to the APMC Register.
0	**SMI# Gate (CSMIGATE):** When bit 0 = 1, the SMI# signal is enabled and a system management interrupt condition causes the SMI# signal to be asserted. When bit 0 = 0 (default), the SMI# signal is masked and negated. This bit only affects the SMI# signal and does not effect the detection/recording of SMI events (i.e., This bit does not effect the SMI status bits in the SMIREQ Register). Thus, SMI conditions can be pending when this bit is set to 1. If an SMI is pending when this bit is set to 1, the SMI# signal is asserted.

3.1.20 SMIEN—SMI ENABLE REGISTER

Address Offset: A2-A3h
Default Value: 0000h
Attribute: Read/Write
Size: 16 Bits

For the 82374SB, his register enables the generation of SMI (asserting the SMI# signal) for the associated hardware events (bits[5:0]), and software events (bit 7). When a hardware event is enabled, the occurrence of a corresponding event results in the assertion of SMI#, if enabled via the SMICNTL Register. The SMI# is asserted independent of the current power state (Power-On or Fast Off). The default for all sources in this register is disabled.

Bit	Description
15:8	**Reserved**
7	**APMC Write SMI Enable:** This bit enables SMI for writes to the APMC Register. When bit 7 = 1, writes to the APMC Register generate an SMI. When bit 7 = 0, writes to the APMC Register do not generate an SMI.
6	**EXTSMI# SMI Enable:** When bit 6 = 1, asserting the EXTSMI# input signal generates an SMI. When bit 6 = 0, asserting EXTSMI# does not generate an SMI.
5	**Fast Off Timer SMI Enable:** This bit enables the Fast Off Timer to generate an SMI. When bit 5 = 1, the timer generates an SMI when it decrements to zero. When bit 5 = 0, the timer does not generate an SMI.
4	**IRQ12 SMI Enable (PS/2 Mouse Interrupt):** This bit enables the IRQ12 signal to generate an SMI. When bit 4 = 1, asserting the IRQ12 input signal generates an SMI. When bit 4 = 0, asserting IRQ12 does not generate an SMI.
3	**IRQ8 SMI Enable (RTC Alarm Interrupt):** This bit enables the IRQ8 signal to generate an SMI. When bit 3 = 1, asserting the IRQ8 input signal generates an SMI. When bit 3 = 0, asserting IRQ8 does not generate an SMI.

Bit	Description
2	**IRQ4 SMI Enable (COM2/COM4 Interrupt or Mouse):** Thsi bit enables the IRQ4 signal to generate an SMI. When bit 1 = 1, asserting the IRQ3 input signal generates an SMI. When bit 2 = 0, asserting IRQ4 does not generate an SMI.
1	**IRQ3 SMI Enable (COM1/COM3 Interrupt or Mouse):** This bit enables the IRQ3 signal to generate an SMI. When bit 1 = 1, asserting the IRQ3 input signal generates an SMI. When bit 1 = 0, asserting IRQ3 does not generate an SMI.
0	**IRQ1 SMI Enable (Keyboard Interrupt):** This bit enables the IRQ1 signal to generate an SMI. When bit 0 = 1, asserting the IRQ1 input signal generates an SMI. When bit 0 = 0, asserting IRQ1 does not generate an SMI.

3.1.21 SEE—SYSTEM EVENT ENABLE REGISTER

Address Offset: A4-A7h
Default Value: 00000000h
Attribute: Read/Write
Size: 32 Bits

For the 82374SB, this register enables hardware events as system events or break events for power management control. Note that all of the functional bits in the SEE Register provide system event control. In addition, all bits also provide break event control. The default for each system/break event in this register is disabled.

System events: Activity by these events can keep the system from powering down. When a system event is enabled, the corresponding hardware event activity prevents a Fast Off powerdown condition. Anytime the corresponding hardware event occurs (signal is asserted), the Fast Off Timer is re-loaded with its initial count.

Break events: These events can awaken a powered down system. When a break event is enabled, the corresponding hardware event activity powers up the system by negating STPCLK#. Note that STPCLK# is not negated until the stop grant special cycle has been generated by the CPU. Thus, from the time that STPCLK# is asserted until the stop grant cycle is returned, the occurrence of subsequent break events are latched in the ESC.

NOTE:
INIT is always enabled as a break event. However, INIT only causes a break event after a stop grant special cycle has been received. If INIT is asserted while STPCLK# is active and then negated before the stop grant cycle is received, INIT does not cause a break event.

Bit	Description
31	**Fast Off SMI Enable (FSMIEN):** When bit 31 = 1 (enabled), an SMI causes a system event that re-loads the Fast Off Timer and a break event that negates the STPCLK# signal. When bit 31 = 0 (disabled), an SMI does not re-load the Fast Off Timer or negate the STPCLK# signal.
30	**Reserved**
29	**Fast Off NMI Enable (FNMIEN):** When bit 29 = 1 (enabled), an NMI (e.g., parity error) causes a system event that re-loads the Fast Off Timer and a break event that negates the STPCLK# signal. When bit 29 = 0 (disabled), an SMI does not re-load the Fast Off Timer or negate the STPCLK# signal.

Bit	Description
28:16	**Reserved**
15:3	**Fast Off IRQ[15:3] Enable (FIRQ[15:3]EN):** These bits are used to prevent the system from entering Fast Off and break any current powerdown state when the selected hardware interrupt occurs. When a bit = 1 (enabled), the corresponding interrupt causes a system event that re-loads the Fast Off Timer and a break event that negates the STPCLK# signal. When a bit = 0 (disabled), the corresponding interrupt does not re-load the Fast Off Timer or negate the STPCLK# signal.
2	**Reserved**
1:0	**Fast Off IRQ[1:0] Enable (FIRQ[1:0]EN):** These bits are used to prevent the system from entering Fast Off and break any current powerdown state when the selected hardware interrupt occurs. When a bit = 1, the corresponding interrupt causes a system event that re-loads the Fast Off Timer and a break event that negates the STPCLK# signal. When a bit = 0 (disabled), the corresponding interrupt does not re-load the Fast Off Timer or negate the STPCLK# signal.

3.1.22 FTMR—FAST OFF TIMER REGISTER

Address Offset: A8h
Default Value: 0Fh
Attribute: Read/Write
Size: 8 Bits

For the 82374SB, the Fast Off Timer is used to indicate (through an SMI) that the system has been idle for a pre-programmed period of time. The Fast Off Timer consists of a count-down timer and the value programmed into this register is loaded into the Fast Off Timer when an enabled system event occurs. When the timer expires, an SMI special cycle is generated. When the Fast Off Timer is enabled (bit 3 = 0 in the SMICNTL Register), the timer counts down from the value loaded into this register. The count time interval is one minute. When the Fast Off Timer reaches 00h, an SMI is generated and the timer is re-load with the value programmed into this register. If an enabled system event occurs before the Fast Off Timer reaches 00h, the Fast Off Timer is re-loaded with the value in this register.

NOTE:
Before writing to the FTMR Register, the Fast Off Timer must be stopped via bit 3 of the SMICNTL Register. In addition, this register should NOT be programmed to 00h.

Bit	Description
7:0	**Fast Off Timer Value:** Bits[7:0] contain the starting count value. A read from the FTMR Register returns the value last written.

3.1.23 SMIREQ—SMI REQUEST REGISTER

Address Offset: AA-ABh
Default Value: 00h
Attribute: Read/Write
Size: 16 Bits

For the 82374SB, the SMIREQ Register contains status bits indicating the cause of an SMI. When an enabled event causes an SMI, the ESC automatically sets the corresponding event's status bit to 1. Software sets the status bits to 0 by writing a 0 to them. Only the ESC hardware can set status bits to a 1. Software

writing a 1 to any of the status bits has no effect. If software attempts to set a status bit to 0 at the same time that the ESC is setting it to 1, the bit is set to 1 (i.e., the ESC hardware dominates).

The SMI handler can query the status bits to see what caused the SMI and then branch to the appropriate routine. As the individual routines complete the handler resets the appropriate status bit by writing a 0 to the corresponding bit.

Each of the SMIREQ bits is set by the ESC in response to the activation of the corresponding SMI event. If the SMI event is still active when the corresponding SMIREQ bit is set to 0, the ESC does not set the status bit back to a 1 (i.e., there is only one status indication per active SMI event).

When an IRQx signal is asserted, the corresponding RIRQx bit is set to a 1. If the IRQx signal is still active when software sets the RIRQx bit to 0, RIRQx is not set back to a 1. The IRQx may be negated before software sets the RIRQx bit to 0. If the RIRQx bit is set to 0 at the same time a new IRQx is activated, RIRQx remains at 1. This indicates to the SMI handler that a new SMI event has been detected.

NOTE:

1. The SMIREQ bits are set, cleared, or read independently of each other and independently of the CSMIGATE bit in the SMICNTL Register.
2. If an IRQx is set in level mode and shared by two devices, the IRQ should not be enabled as an SMI# event. The ESC's SMIREQ bits are essentially set with an edge. When the second IRQ occurs on a shared IRQ, there is no second edge and the SMI# will not be generated for the second IRQ.

Bit	Description
15:8	**Reserved**
7	**APM SMI Status (RAPMC):** The ESC sets this bit to 1 to indicate that a write to the APM Control Register caused an SMI. Software sets this bit to a 0 by writing a 0 to it.
6	**EXTSMI# SMI Status (REXT):** The ESC sets this bit to 1 to indicate that EXTSMI# caused an SMI. Software sets this bit to a 0 by writing a 0 to it.
5	**Fast Off Timer Expired Status (RFOT):** The ESC sets this bit to 1 to indicate that the Fast Off Timer expired and caused an SMI. Software sets this bit to a 0 by writing a 0 to it. When the Fast Off Timer expires, the ESC sets this bit to a 1. Note that the timer re-starts counting one the next clock after it expires.
4	**IRQ12 Request SMI Status (RIRQ12):** The ESC sets this bit to 1 to indicate that IRQ12 caused an SMI. Software sets this bit to a 0 by writing a 0 to it.
3	**IRQ8# Request SMI Status:** The ESC sets this bit to 1 to indicate that IRQ8# caused an SMI. Software sets this bit to a 0 by writing a 0 to it.
2	**IRQ4 Request SMI Status:** The ESC sets this bit to 1 to indicate that IRQ4 caused an SMI. Software sets this bit to a 0 by writing a 0 to it.
1	**IRQ3 Request SMI Status:** The ESC sets this bit to 1 to indicate that IRQ3 caused an SMI. Software sets this bit to a 0 by writing a 0 to it.
0	**IRQ1 Request SMI Status:** The ESC sets this bit to 1 to indicate that IRQ1 caused an SMI. Software sets this bit to a 0 by writing a 0 to it.

3.1.24 CTLTMRCLOCK SCALE STPCLK# LOW TIMER

Address Offset: ACh
Default Value: 00h
Attribute: Read/Write
Size: 8 Bits

For the 82374SB, the value in this register defines the duration of the STPCLK# asserted period when bit 2 in the SMICNTL Register is set to 1. The value in this register is loaded into the STPCLK# Timer when STPCLK# is asserted. However, the timer does not start until the Stop Grant Bus Cycle is received. The STPCLK# timer counts using a 32 μs clock.

Bit	Description
7:0	**Clock Scaling STPCLK# Low Timer Value:** Bits[7:0] define the duration of the STPCLK# asserted period during clock throttling.

3.1.25 CTLTMRH—CLOCK SCALE STPCLK# HIGH TIMER

Address Offset: AEh
Default Value: 00h
Attribute: Read/Write
Size: 8 Bits

For the 82374SB, the value in this register defines the duration of the STPCLK# negated period when bit 2 in the SMICNTL Register is set to 1. The value in this register is loaded into the STPCLK# Timer when STPCLK# is negated. The STPCLK# timer counts using a 32 μs clock.

Bit	Description
7:0	**Clock Scaling STPCLK# High Timer Value:** Bits[7:0] define the duration of the STPCLK# negated period during clock throttling.

3.2 DMA Register Description

The ESC contains DMA circuitry that incorporates the functionality of two 82C37 DMA controllers (DMA1 and DMA2). The DMA registers control the operation of the DMA controllers and are all accessible from the EISA Bus. This section describes the DMA registers. Unless otherwise stated, a reset sets each register to its default value. The operation of the DMA is further described in Chapter 6.0, DMA Controller.

3.2.1 DCOM—COMMAND REGISTER

Register Location: 08h—Channels 0-3
0D0h—Channels 4-7
Default Value: 00h
Attribute: Write Only
Size: 8 Bits

This 8-bit register controls the configuration of the DMA. It is programmed by the microprocessor in the Program Condition and is cleared by reset or a Master Clear instruction. Note that disabling Channels 4-7 will also disable Channels 0-3, since Channels 0-3 are cascaded onto Channel 4. The DREQ and DACK# channel assertion sensitivity is assigned by channel group, not per individual Channel. For priority resolution the DMA

consists of two logical channel groups–Channels 0-3 (Controller 1-DMA1) and Channels 4-7 (Controller 2-DMA2). Both groups may be assigned fixed priority, one group can be assigned fixed priority and the second rotating priority, or both groups may be assigned rotating priority. A detailed description of the channel priority scheme is found in the DMA functional description, Section 6.5. Following a reset or DMA Master Clear, both DMA-1 and DMA-2 are enabled in fixed priority, the DREQ sense level is active high, and the DACK# assertion level is active low.

Bit	Description
7	**DACK# Assert Level:** Bit 7 controls the DMA channel request acknowledge (DACK#) assertion level. Following reset, the DACK# assertion level is active low. The low level indicates recognition and acknowledgment of the DMA request to the DMA slave requesting service. Writing a 0 to bit 7 assigns active low as the assertion level. When a 1 is written to this bit, a high level on the DACK# line indicates acknowledgment of the request for DMA service to the DMA slave.
6	**DREQ Sense Assert Level:** Bit 6 controls the DMA channel request (DREQ) assertion detect level. Following reset, the DREQ sense assert level is active high. In this condition, an active high level sampled on DREQ is decoded as an active DMA channel request. Writing a 0 to bit 6 assigns active high as the sense assert level. When a 1 is written to this bit, a low level on the DREQ line is decoded as an active DMA channel request.
5	**Reserved:** Must be 0.
4	**DMA Group Arbitration:** Each channel group is individually assigned either fixed or rotating arbitration priority. At reset, each group is initialized in fixed priority. Writing a 0 to bit 4 assigns fixed priority to the channel group, while writing a 1 assigns rotating priority to the group.
3	**Reserved:** Must be 0.
2	**DMA Group Enable:** Writing a 1 to this bit disables the DMA channel group, while writing a 0 to this bit enables the DMA channel group. Both channel groups are enabled following reset. Disabling Channel group 4-7 also disables Channel group 0-3, which is cascaded through Channel 4.
1:0	**Reserved:** Must be 0.

3.2.2 DCM—DMA CHANNEL MODE REGISTER

Register Location: 0Bh—Channels 0-3
0D6h—Channels 4-7
Default Value: 000000xxb
Attribute: Write Only
Size: 8 Bits

Each channel has a Mode Register associated with it. The Mode registers provide control over DMA Transfer type, transfer mode, address increment/decrement, and autoinitialization. When writing to the register, bits[1:0] determine which channel's Mode Register will be written and are not stored. Only bits[7:2] are stored in the mode register. This register is set to the default value upon reset and Master Clear. Its default value is Verify transfer, autoinitialize disable, Address increment, and Demand mode. Channel 4 defaults to cascade mode and cannot be programmed for any mode other than cascade mode.

Bit	Description
7:6	**DMA Transfer Mode:** Each DMA channel can be programmed in one of four different modes: single transfer, block transfer, demand transfer and cascade. **Bits[7:6] Transfer Mode** 00 Demand mode 01 Single mode 10 Block mode 11 Cascade mode
5	**Address Increment/Decrement Select:** Bit 5 controls address increment/decrement during multi-byte DMA transfers. When bit 5 = 0, address increment is selected. When bit 5 = 1, address decrement is selected. Address increment is the default after a PCIRST# cycle or Master Clear command.
4	**Autoinitialize Enable:** When bit 4 = 1, the DMA restores the Base Page, Address, and Word count information to their respective current registers following a terminal count (TC). When bit 4 = 0, the autoinitialize feature is disabled and the DMA does not restore the above mentioned registers. A PCIRST# or Master Clear disables autoinitialization (sets bit 4 to 0).
3:2	**DMA Transfer Type:** Verify, write and read transfer types are available. Verify transfer is the default transfer type upon PCIRST# or Master Clear. Write transfers move data from an I/O device to memory. Read transfers move data from memory to an I/O device. Verify transfers are pseudo transfers; addresses are generated as in a normal read or write transfer and the device responds to EOP etc. However, with Verify transfers, the ISA memory and I/O cycle lines are not driven. Bit combination 11 is illegal. When the channel is programmed for cascade ([7:6] = 11) the transfer type bits are irrelevant. **Bits[3:2] Transfer Type** 00 Verify transfer 01 Write transfer 10 Read Transfer 11 Illegal
1:0	**DMA Channel Select:** Bits[1:0] select the DMA Channel Mode Register that will be written by bits[7:2]. **Bits[1:0] Channel** 00 Channel 0 (4) 01 Channel 1 (5) 10 Channel 2 (6) 11 Channel 3 (7)

3.2.3 DCEM—DMA CHANNEL EXTENDED MODE REGISTER

Register Location: 040Bh—Channels 0-3
04D6h—Channels 4-7
Default Value: 000000xxb
Attribute: Write Only
Size: 8 Bits

Each channel has an Extended Mode Register. The register is used to program the DMA device data size, timing mode, EOP input/output selection, and Stop register selection. When writing to the register, bits[1:0] determine which channel's Extended Mode Register will be written and are not stored. Only bits[7:2] are stored in the Extended Mode Register. Four timing modes are available: ISA-compatible, A, B, and Burst.

The default bit values for each DMA group are selected upon reset. A Master Clear or any other programming sequence will not set the default register settings. The default programmed values for DMA1 Channels 0-3 are 8-bit I/O Count by Bytes, Compatible timing, and EOP output. The default values for DMA2 Channels 4-7 are 16-bit I/O Count by Words with shifted address, Compatible timing, and EOP output. These default settings provide a rigorous ISA-compatible DMA implementation.

NOTE:
DMA1/DMA2 refer to the original PC-AT implementation which used two discrete 8237 DMA controllers. In this context, DMA1 refers to DMA Channels 0-3 and DMA2 refers to DMA Channels 4-7. The PC-AT used Channel 4 (Channel 0 of DMA2) as a cascade channel for DMA1. Consequently, Channel 4 is not used in compatible DMA controllers although the compatible DMA registers are kept to maintain compatibility with the original PC-AT. Because Channel 4 is not used, the DMA controller does not support extended registers for Channel 4.

Bit	Description
7	**Stop Register:** Bit 7 of this register selects whether or not the Stop registers associated with this channel are to be used. Normally the Stop Registers will not be used. This function was added to help support data communication or other devices that work from a ring buffer in memory. Upon reset, the bit 7 is set to 0-Stop register disabled. The detailed Stop register functional description discusses the use of the Stop registers.
6	**EOP Input/Output:** Bit 6 of the Extended Mode register selects whether the EOP signal is to be used as an output during DMA on this channel or an input. EOP will generally be used as an output, as was available on the PCAT. The input function was added to support Data Communication and other devices that would like to trigger an autoinitialize when a collision or some other event occurs. The direction of EOP is switched when DACK is changed (when a different channel wins the arbitration and is granted the bus). There may be some overlap of the ESC driving the EOP signal along with the DMA slave. However, during this overlap both devices will be driving the signal to a low level (negated). For example, assume Channel 2 is about to go inactive (DACK negated) and channel 1 is about to go active. If Channel 2 is programmed for "EOP OUT" and Channel 1 is programmed for "EOP IN", when Channel 2's DACK is negated and Channel 1's DACK is asserted, the ESC may be driving EOP to a low value on behalf of Channel 2 at the same time the device connected to Channel 1 is driving EOP in to the ESC, also at an inactive level. This overlap will only last until the ESC EOP output buffer is tristated, and will not effect the DMA operation. Upon reset, the value of bit 6 is 0 (EOP output selected).

Bit	Description
5:4	**DMA Cycle Timing Mode:** The ESC supports four DMA transfer timings: ISA-compatible, Type A, Type B, and Burst. Each timing and its corresponding code are described below. Upon reset, compatible timing is selected and the value of these bits is "00". The cycle timings noted below are for a BCLK (8.33 MHz maximum BCLK frequency). DMA cycles to ISA expansion bus memory will default to compatible timing if the channel is programmed in one of the performance timing modes (Type A, B, or Burst). 00 Compatible Timing DMA slaves on the ISA bus may run compatible DMA cycles. Bits[5:4] must be programmed to 00. Compatible timing is provided for DMA slave devices, which, due to some design limitation, cannot support one of the faster timings. Compatible timing runs at 9 BCLKs (1080 ns/single cycle) and 8 BCLKs (960 ns/cycle) during the repeated portion of a BLOCK or DEMAND mode transfers. 01 Type "A" Timing Type "A" timing is provided to allow shorter cycles to EISA memory. If ISA memory is decoded, the system automatically reverts to ISA DMA type compatible timing on a cycle-by-cycle basis. Type "A" timing runs at 7 BCLKs (840 ns/single cycle) and 6 BCLKs (720 ns/cycle) during the repeated portion of a BLOCK or DEMAND mode transfer. Type "A" timing varies from compatible timing primarily in shortening the memory operation to the minimum allowed by system memory. The I/O portion of the cycle (data setup on write, I/O read access time) is the same as with compatible cycles. The actual active command time is shorter, but it is expected that the DMA devices which provide the data access time or write data setup time should not require excess IOR# or IOW# command active time. Because of this, most ISA DMA devices should be able to use type "A" timing. 10 Type "B" Timing Type "B" timing is provided for 8-/16- bit ISA or EISA DMA devices which can accept faster I/O timing. Type "B" only works with EISA memory. Type "B" timing runs at 6 BCLKs (720 ns/single cycle) and 4 BCLKs (480 ns/cycle) during the repeated portion of a BLOCK or DEMAND mode transfer. Type "B" timing requires faster DMA slave devices than compatible timing in that the cycles are shortened so that the data setup time on I/O write cycles is shortened and the I/O read access time is required to be faster. Some of the current ISA devices should be able to support type "B" timing, but these will probably be more recent designs using relatively fast technology. 11 Type "C" Timing (Burst) Burst timing is provided for high performance EISA DMA devices. The DMA slave device needs to monitor the EXRDY and IORC# or IOWC# signals to determine when to change the data (on writes) or sample the data (on reads). This timing will allow up to 33 MBytes per second transfer rate with a 32-bit DMA device and 32-bit memory. Note that 8- or 16-bit DMA devices are supported (through the programmable Address size) and that they use the "byte lanes" natural to their size for the data transfer. As with all bursts, the system will revert to two BCLK cycles if the memory does not support burst. When a DMA burst cycle accesses non-burst memory and the DMA cycle crosses a page boundary into burstable memory, the ESC will continue performing non-burst cycles. This will not cause a problem since the data is still transferred correctly.

Bit	Description
3:2	**Addressing Mode:** The ESC supports 8-, 16-, and 32-bit DMA device data sizes. The four data size options are programmable with bits[3:2]. Both the 8 bit I/O, "Count By Bytes" Mode and the 16-bit I/O, "Count By Words" (Address Shifted) Mode are ISA compatible. The 16-bit and 32-bit I/O, "Count By Bytes" Modes are EISA extensions. Byte assembly/disassembly is performed by the EISA Bus Controller. Each of the data transfer size modes is discussed below. 00 8-Bit I/O, "Count By Bytes" Mode In 8 bit I/O, "count by bytes" mode, the address counter can be programmed to any address. The count register is programmed with the "number of bytes minus 1" to transfer. 01 16-Bit I/O, "Count By Words" (Address Shifted) Mode In "count by words" mode (address shifted), the address counter can be programmed to any even address, but must be programmed with the address value shifted right by one bit. The Page registers are not shifted during DMA transfers. Thus, the least significant bit of the Low Page register is ignored when the address is driven out onto the bus. The Word Count register is programmed with the number of words minus 1 to be transferred. 10 32-Bit I/O, "Count By Bytes" Mode In 32-bit "count by bytes" mode, the address counter can be programmed to any byte address. For most DMA devices, however, it should only be programmed to a Dword aligned address. If the starting address is not Dword aligned then the DMA controller will do a partial Dword transfer during the first and last during the first and last transfers if necessary. The bus controller logic will do the byte/word assembly necessary to read or write any size memory device and both the DMA and bus controllers support burst for this mode. In this mode, the Address register is usually incremented or decremented by four and the byte count is usually decremented by four. The Count register should be programmed with the number of bytes to be transferred minus 1. 11 16-Bit I/O, "Count By Bytes" Mode In 16-bit "count by bytes" mode, the address counter can be programmed to any byte address. For most DMA devices, however, it should be programmed only to even addresses. If the address is programmed to an odd address, then the DMA controller will do a partial word transfer during the first and last transfer if necessary. The bus controller will do the byte/word assembly necessary to write any size memory device. In this mode, the Address register is incremented or decremented by two and the byte count is decremented by the number of bytes transferred during each bus cycle. The Word Count register is programmed with the "number of bytes minus 1" to be transferred. This mode is offered as an extension of the two ISA compatible modes discussed above. This mode should only be programmed for 16 bit ISA DMA slaves.
1:0	**DMA Channel Select:** Bits[1:0] select the particular channel that will have its DMA Channel Extend Mode Register programmed with bits[7:2]. **Bits[1:0]** — **Channel** 00 — Channel 0 (4) 01 — Channel 1 (5) 10 — Channel 2 (6) 11 — Channel 3 (7)

3.2.4 DR—DMA REQUEST REGISTER

Register Location: 09h—Channels 0-3
0D2h—Channels 4-7
Default Value: 000000xxb
Attribute: Write Only
Size: 8 Bits

Each channel has a Request bit associated with it in one of the two Request Registers. The Request register is used by software to initiate a DMA request. The DMA responds to the software request as though DREQ[x] is asserted. These requests are non-maskable and subject to prioritization by the Priority Encoder network (refer to the Channel Priority Functional Description). Each register bit is set or reset separately under software control or is cleared upon generation of a TC. The entire register is cleared upon reset or a Master Clear. It is not cleared upon a RSTDRV output. To set or reset a bit, the software loads the proper form of the data word. Bits[1:0] determine which channel Request register will be written. In order to make a software request, the channel must be in Block Mode. The Request register status for DMA1 and DMA2 is output on bits[7:4] of a Status register read to the appropriate port.

Bit	Description
7:3	**Reserved:** Must be 0.
2	**DMA Channel Service Request:** Writing a 0 to bit 2 resets the individual software DMA channel request bit. Writing a 1 to bit 2 will set the request bit. The request bit for each DMA channel is reset to 0 upon a reset or a Master Clear.
1:0	**DMA Channel Select:** Bits[1:0] select the DMA channel mode register to program with bit 2. **Bits[1:0] Channel** 00 Channel 0 01 Channel 1 (5) 10 Channel 2 (6) 11 Channel 3 (7)

3.2.5 MASK REGISTER—WRITE SINGLE MASK BIT

Register Location: 0Ah—Channels 0-3
0D4h—Channels 4-7
Default Value: 000001xxb
Attribute: Write Only
Size: 1 Bit/Channel

Each DMA channel has a mask bit that can disable an incoming DMA channel service request DREQ[x] assertion. Two registers store the current mask status for DMA1 and DMA2. Setting the mask bit disables the incoming DREQ[x] for that channel. Clearing the mask bit enables the incoming DREQ[x]. A channel's mask bit is automatically set when the Current Word Count register reaches terminal count (unless the channel is programmed for autoinitialization). Each mask bit may also be set or cleared under software control. The entire register is also set by a reset or a Master Clear. Setting the entire register disables all DMA requests until a clear Mask register instruction allows them to occur. This instruction format is similar to the format used with the Request register.

Individually masking DMA Channel 4 (DMA controller 2, Channel 0) will automatically mask DMA Channels [3:0], as this Channel group is logically cascaded onto Channel 4. Setting this mask bit disables the incoming DREQ's for Channels [3:0].

Bit	Description
7:3	**Reserved:** Must be 0.
2	**DMA Channel Mask Set/Clear:** Writing a 1 to bit 2 sets the mask bit and disables the incoming DREQ for the selected channel. Writing a 0 to bit 2 clears the mask bit and enables the incoming DREQ for the elected channel.
1:0	**DMA Channel Select:** Bits[1:0] select the DMA Channel Mode Register to program with bit 2. **Bits[1:0] Channel** 00 Channel 0 (4) 01 Channel 1 (5) 10 Channel 2 (6) 11 Channel 3 (7)

3.2.6 WAMB—WRITE ALL MASK BITS REGISTER

Register Location: 0Fh—Channels 0-3
0DEh—Channels 4-7
Default Value: 0Fh
Attribute: Read/Write
Size: 8 Bits

This command allows enabling and disabling of incoming DREQ assertions by writing the mask bits for each controller, DMA1 or DMA2, simultaneously rather than by individual channel as is done with the "Write Single Mask Bit" command. Two registers store the current mask status for DMA1 and DMA2. Setting the mask bit disables the incoming DREQ[x] for that channel. Clearing the mask bit enables the incoming DREQ[x]. Unlike the "Write Single Mask Bit" command, this command includes a status read to check the current mask status of the selected DMA channel group. When read, the mask register current status appears on bits[3:0]. A channel's mask bit is automatically set when the Current Word Count register reaches terminal count (unless the channel is programmed for autoinitialization). The entire register is also set by a reset or a Master Clear. Setting the entire register disables all DMA requests until a clear Mask register instruction allows them to occur.

Two important points should be taken into consideration when programming the mask registers. First, individually masking DMA Channel 4 (DMA controller 2, Channel 0) will automatically mask DMA Channels [3:0], as this channel group is logically cascaded onto Channel 4. Second, masking off DMA controller 2 with a write to port 0DEh will also mask off DREQ assertions from DMA controller 1 for the same reason: when DMA Channel 4 is masked, so are DMA Channels 0-3.

Bit	Description
7:4	**Reserved:** Must be 0.
3:0	**Channel Mask Bits:** Setting the bit(s) to a 1 disables the corresponding DREQ(s). Setting the bit(s) to a 0 enables the corresponding DREQ(s). Bits[3:0] are set to 1 upon PCIRST# or Master Clear. When read, bits[3:0] indicate the DMA channel [3:0] ([7:4]) mask status. **Bit Channel** 0 0(4) 1 1(5) 2 2(6) 3 3(7) **NOTE:** Disabling channel 4 also disables channels 0-3 due to the cascade of DMA1 through channel 4 of DMA2.

3.2.7 DS—DMA STATUS REGISTER

Register Location: 08h—Channels 0-3
0D0h—Channels 4-7
Default Value: 00h
Attribute: Read Only
Size: 8 Bits

Each DMA controller has a read-only Status register. A Status register read is used when determining which channels have reached terminal count and which channels have a pending DMA request. Bits[3:0] are set every time a TC is reached by that channel. These bits are cleared upon reset and on each Status Read. Bits[7:4] are set whenever their corresponding channel is requesting service.

Bit	Description
7:4	**Request Status:** When a valid DMA request is pending for a channel (on its DREQ signal line), the corresponding bit is set to 1. When a DMA request is not pending for a particular channel, the corresponding bit is set to 0. The source of the DREQ may be hardware, a timed-out block transfer, or a software request. Note that channel 4 does not have DREQ or DACK lines, so the response for a read of DMA2 status for channel 4 is irrelevant. **Bit Channel** 4 0 5 1(5) 6 2(6) 7 3(7)
3:0	**Terminal Count Status:** When a channel reaches terminal count (TC), its status bit is set to 1. If TC has not been reached, the status bit is set to 0. Note that channel 4 is programmed for cascade, and is not used for a DMA transfer. Therefore, the TC bit response for a status read on DMA2 for channel 4 is irrelevant. **Bit Channel** 0 0 1 1(5) 2 2(6) 3 3(7)

3.2.8 DB&CA—DMA BASE AND CURRENT ADDRESS REGISTER (8237 COMPATIBLE SEGMENT)

Register Location: 000h—DMA Channel 0
002h—DMA Channel 1
004h—DMA Channel 2
006h—DMA Channel 3
0C0h—DMA Channel 4
0C4h—DMA Channel 5
0C8h—DMA Channel 6
0CCh—DMA Channel 7
Default Value: 0000h
Attribute: Read/Write
Size: 16 Bits per channel

Each channel has a 16-bit Current Address register. This register holds the value of the 16 least significant bits of the full 32-bit address used during DMA transfers. The address is automatically incremented or decremented after each transfer and the intermediate values of the address are stored in the Current Address register during the transfer. This register is written to or read from by the microprocessor or bus master in successive 8-bit bytes. The programmer must issue the "Clear Byte Pointer Flip-Flop" command to reset the internal byte pointer and correctly align the write prior to programming the Current address register. After clearing the Byte Pointer Flip-flop, the first write to the Current Address port programs the low byte, bits[7:0], and the second write programs the high byte, bits[15:8]. This procedure applies for read cycles also. It may also be re-initialized by an autoinitialize back to its original value. autoinitialize takes place only after a TC or EOP.

Each channel has a Base Address register located at the same port address as the corresponding Current Address register. These registers store the original value of their associated Current registers. During autoinitialize these values are used to restore the Current registers to their original values. The Base registers are written simultaneously with their corresponding Current register in successive 8-bit bytes by the microprocessor. The Base registers cannot be read by any external agents.

In Scatter-Gather Mode these registers store the lowest 16-bits of the current memory address. During a Scatter-Gather transfer the DMA will load a reserve buffer into the base memory address register.

In Chaining Mode these register store the lowest 16-bits of the current memory address. The CPU will program the base register set with a reserve buffer.

Bit	Description
15:0	**Base and Current Address:** These bits represent the 16 least significant address bits used during DMA transfers. Together with the DMA Low Page register, they help form the ISA-compatible 24-bit DMA address. As an extension of the ISA compatible functionality, the DMA High Page register completes the 32-bit address needed when implementing ESC extensions such as DMA to the PCI bus slaves that can take advantage of full 32-bit addressability. Upon reset or Master Clear, the value of these bits is 0000h.

3.2.9 DB&CBW—DMA BASE AND CURRENT BYTE/WORD COUNT REGISTER (8237 COMPATIBLE SEGMENT)

Register Location: 001h—DMA Channel 0
003h—DMA Channel 1
005h—DMA Channel 2
007h—DMA Channel 3
0C2h—DMA Channel 4
0C6h—DMA Channel 5
0CAh—DMA Channel 6
0CEh—DMA Channel 7
Default Value: 0000h
Attribute: Read/Write
Size: 16 Bits per channel

Each channel has a 16-bit Current Byte/Word Count register. This register determines the lower 16 bits for the number of transfers to be performed. There is a total of 24 bits in the Byte/Word Count registers. The uppermost 8 bits are in the High Byte/Word Count register. The actual number of transfers will be one more than the number programmed in the Current Byte/Word Count register (i.e., programming a count of 100 will result in 101 transfers). The byte/word count is decremented after each transfer. The intermediate value of the byte/word count is stored in the register during the transfer. When the value in the register goes from zero to 0FFFFFFh, a TC will be generated.

Following the end of a DMA service it may also be re-initialized by an autoinitialization back to its original value. autoinitialize can occur only when a TC occurs. If it is not autoinitialized, this register will have a count of FFFFh after TC.

When the Extended Mode register is programmed for "count by word" transfers to/from a 16-bit I/O, with shifted address, the Byte/Word count will indicate the number of 16-bit words to be transferred.

When the Extended Mode register is programmed for "count by byte" transfers, the Byte/Word Count will indicate the number of bytes to be transferred. The number of bytes does not need to be a multiple of the transfer size in this case.

Each channel has a Base Byte/Word Count register located at the same port address as the corresponding Current Byte/Word Count register. These registers store the original value of their associated Current registers. During autoinitialize these values are used to restore the Current registers to their original values. The Base registers cannot be read by any external agents.

In Scatter-Gather mode these registers store the lowest 16-bits of the current Byte/Word Count. During a Scatter-Gather transfer the DMA will load a reserve buffer into the base Byte/Word Count register.

In Chaining Mode these register store the lowest 16-bits of the current Byte/Word Count. The CPU will then program the base register set with a reserve buffer.

Bit	Description
15:0	**Base and Current Byte/Word Count:** These bits represent the lower 16 byte/word count bits used when counting down a DMA transfer. Upon reset or Master Clear, the value of these bits is 0000h.

3.2.10 DMA BASE AND CURRENT HIGH BYTE/WORD COUNT REGISTER; DMA BASE HIGH BYTE/ WORD COUNT REGISTER

Register Location: 401h—DMA Channel 0
403h—DMA Channel 1
405h—DMA Channel 2
407h—DMA Channel 3
4C6h—DMA Channel 5
4CAh—DMA Channel 6
4CEh—DMA Channel 7
Default Value: 00h
Attribute: Read/Write
Size: 8 Bits per channel

Each channel has a 8-bit Current High Byte/Word Count register. This register provides the uppermost 8 bits for the number of transfers to be performed. The byte/word count is decremented after each transfer. The intermediate value of the byte/word count is stored in the register during the transfer. When the value in the register goes from zero to FFFFh, a TC may be generated.

Following the end of a DMA service it may also be re-initialized by an autoinitialization back to its original value. autoinitialize can occur only when a TC occurs. If it is not autoinitialized, this register will have a count of FFFFh after TC.

The High Byte/Word Count register must be the last Byte/Word Count register programmed. Writing to the 8237 Compatible Byte/Word Count registers will clear the High Byte/Word Count register to 00h.

When the Extended Mode register is programmed for "count by word" transfers to/from a 16-bit I/O, with shifted address, the Byte/Word count will indicate the number of 16-bit words to be transferred.

When the Extended Mode register is programmed for "count by byte" transfers, the Byte/Word Count will indicate the number of bytes to be transferred. The number of bytes does not need to be a multiple of the transfer size in this case.

Each channel has a Base High Byte/Word Count register located at the same port address as the corresponding Current High Byte/Word Count register. These registers store the original value of their associated Current registers. During autoinitialize these values are used to restore the Current registers to their original values. Normally, the Base registers are written simultaneously with their corresponding Current register in successive 8 bit bytes by the microprocessor. However, in Chaining Mode only the Base register set is programmed and the Current register is not effected. The Base registers cannot be read by any external agents.

In Scatter-Gather mode these registers store the lowest 8 bits of the current High Byte/Word Count. During a Scatter-Gather transfer the DMA will load a reserve buffer into the base High Byte/Word Count register.

In Chaining Mode these register store the lowest 8 bits of the current High Byte/Word Count. The CPU will then program the base register set with a reserve buffer.

Bit	Description
7:0	**Base and Current High Byte/Word Count:** These bits represent the 8 high order byte/word count bits used when counting down a DMA transfer. Upon reset or Master Clear, the value of these bits is 00h.

3.2.11 DMA MEMORY LOW PAGE REGISTER; DMA MEMORY BASE LOW PAGE REGISTER

Register Location: 087h—DMA Channel 0
083h—DMA Channel 1
081h—DMA Channel 2
082h—DMA Channel 3
08Bh—DMA Channel 5
089h—DMA Channel 6
08Ah—DMA Channel 7
Default Value: 00h
Attribute: Read/Write
Size: 8 Bits per channel

Each channel has an 8-bit Low Page register associated with it. The DMA memory Low Page register contains the eight second most-significant bits of the 32-bit address. It works in conjunction with the DMA controller's High Page register and Current Address register to define the complete (32-bit) address for the DMA channel. This 8-bit register is read or written directly by the processor or bus master. It may also be re-initialized by an autoinitialize back to its original value. autoinitialize takes place only after a TC or EOP.

Each channel has a Base Low Page Address register located at the same port address as the corresponding Current Low Page register. These registers store the original value of their associated Current Low Page registers. During autoinitialize these values are used to restore the Current Low Page registers to their original values. The 8-bit Base Low Page registers are written simultaneously with their corresponding Current Low Page register by the microprocessor. The Base Low Page registers cannot be read by any external agents.

During Scatter-Gather these registers store the 8 bits from the third byte of the current memory address. During a Scatter-Gather transfer the DMA will load a reserve buffer into the base memory address register.

In Chaining Mode these register store the 8 bits from the third byte of the current memory address. The CPU will program the base register set with a reserve buffer.

Bit	Description
7:0	**DMA LOW PAGE AND BASE LOW PAGE:** These bits represent the eight second most-significant address bits when forming the full 32-bit address for a DMA transfer. Upon reset or Master Clear, the value of these bits is 00h.

3.2.12 DMAP—DMA PAGE REGISTER

Register Location: 080h, 84h, 85h, 86h, 88h, 8Ch, 8Dh, 8Eh
Default Value: xxh
Attribute: Read/Write
Size: 8 Bits

These registers have no effect on the DMA operation. These registers provide extra storage space in the I/O space for DMA routines.

Bit	Description
7:0	**DMA PAGE:** These bit have no effect on the DMA operation. These bits only provide storage space in the I/O map.

3.2.13 DMALPR—DMA LOW PAGE REFRESH REGISTER

Register Location: 08Fh
Default Value: xxh
Attribute: Read/Write
Size: 8 Bits

The contents of this register are driven on the address byte 2 (LA[23:16]#) during Refresh cycles.

Bit	Description
7:0	**DMA LOW PAGE REFRESH:** The contents of the bits are driven on to the address bus(LA[23:16]) during refresh.

3.2.14 DMAMHPG—DMA MEMORY HIGH PAGE REGISTER; DMA MEMORY BASE HIGH PAGE REGISTER

Register Location: 0487h—DMA Channel 0
0483h—DMA Channel 1
0481h—DMA Channel 2
0482h—DMA Channel 3
048Bh—DMA Channel 5
0489h—DMA Channel 6
048Ah—DMA Channel 7
Default Value: 00h
Attribute: Read/Write
Size: 8 Bits per channel

Each channel has an 8-bit High Page register. The DMA memory High Page register contains the eight most-significant bits of the 32-bit address. It works in conjunction with the DMA controller's Low Page register and Current Address register to define the complete (32-bit) address for the DMA channels and corresponds to the "Current Address" register for each channel. This 8-bit register is read or written directly by the processor or bus master. It may also be re-initialized by an autoinitialize back to its original value. Autoinitialize takes place only after a TC or EOP.

This register is reset to 00h during the programming of both the low page register and the Current Address register. Thus, if this register is not programmed after the other address and Low Page registers are programmed, then its value will be zero. In this case, the DMA channel will operate the same as an 82C37 (from an addressing standpoint). This is the address compatibility mode.

If the high 8 bits of the address are programmed after the other addresses, then the channel will modify its operation to increment (or decrement) the entire 32-bit address. This is unlike the 82C37 "Page" register in the original PCs which could only increment to a 64K boundary (for 8-bit channels) or 128K (for 16-bit channels). This is extended address mode. In this mode, the ISA bus controller will generate the signals MEMR# and MEMW# only for addresses below 16 MBytes.

Each channel has a Base High Page Address register located at the same port address as the corresponding Current High Page Address register. These registers store the original value of their associated Current registers. During autoinitialize these values are used to restore the Current registers to their original values. The 8 bit Base High Page registers are written simultaneously with their corresponding Current register by the microprocessor. The Base registers cannot be read by any external agents.

During Scatter-Gather these registers store the 8 bits from the highest byte of the current memory address. During a Scatter-Gather transfer the DMA will load a reserve buffer into the base memory address register.

In Chaining Mode these register store the 8 bits from the highest byte of the current memory address. The CPU will program the base register set with a reserve buffer.

Bit	Description
7:0	**DMA High Page and Base High Page:** These bits represent the eight most-significant address bits when forming the full 32-bit address for a DMA transfer. Upon reset or Master Clear, the value of these bits is 00h.

3.2.15 DMAHPGR—DMA HIGH PAGE REGISTER REFRESH

Register Location: 048Fh
Default Value: xxh
Attribute: Read/Write
Size: 8 Bits per channel

The contents of this register are driven on the address byte 3 (LA[31:24]#) during Refresh cycles.

Bit	Description
7:0	**DMA High Page Refresh:** The contents of the bits are driven on to the address bus (LA[31:24]) during refresh.

3.2.16 STOP REGISTERS

Register Location: 04E0h—CH0 Stop Reg Bits[7:2]
04E1h—CH0 Stop Reg Bits[15:8]
04E2h—CH0 Stop Reg Bits[23:16]
04E4h—CH1 Stop Reg Bits[7:2]
04E5h—CH1 Stop Reg Bits[15:8]
04E6h—CH1 Stop Reg Bits[23:16]
04E8h—CH2 Stop Reg Bits[7:2]
04E9h—CH2 Stop Reg Bits[15:8]
04EAh—CH2 Stop Reg Bits[23:16]
04ECh—CH3 Stop Reg Bits[7:2]
04EDh—CH3 Stop Reg Bits[15:8]
04EEh—CH3 Stop Reg Bits[23:16]
04F4h—CH5 Stop Reg Bits[7:2]
04F5h—CH5 Stop Reg Bits[15:8]
04F6h—CH5 Stop Reg Bits[23:16]
04F8h—CH6 Stop Reg Bits[7:2]
04F9h—CH6 Stop Reg Bits[15:8]
04FAh—CH6 Stop Reg Bits[23:16]
04FCh—CH7 Stop Reg Bits[7:2]
04FDh—CH7 Stop Reg Bits[15:8]
04FEh—CH7 Stop Reg Bits[23:16]
Default Value: See Below
Attribute: Read/Write
Size: See Below

The Stop registers are used to support a common data communication structure, the ring buffer. The ring buffer data structure and Stop Register operation are described in Section 6.7.4. The Stop registers, in conjunction with a channel's Base and Current address and byte count registers, are used to define a fixed portion of memory for use by the ring buffer data structure. Following a reset, these registers are not reset to 0.

Bit	Description
23:2	**Upper, Mid, Lower Stop Bits:** These 22 bits provide the Stop Address. If the Stop function is enabled then the channel will Stop whenever its Memory Address matches the Stop Address. Bits[23:16] are the upper stop bits. Bits[15:8] are the mid stop bits and bits[7:2] are the lower stop bits. Bits[1:0] are not used and are don't cares.

3.2.17 CHAIN—CHAINING MODE REGISTER

Register Location: 040Ah—Channels 0-3
04D4h—Channels 4-7
Default Value: 000000xxb
Attribute: Write Only
Size: 8 Bits

Each channel has a Chaining Mode register. The Chaining Mode register enables or disables DMA buffer chaining and indicates when the DMA Base registers are being programmed. When writing to the register, bits[1:0] determine which channel's Chaining Mode register to program. The chaining status and interrupt status for all channels can be determined by reading the Chaining Mode Status, Channel Interrupt Status, and Chain Buffer Expiration Control registers. The Chaining Mode register is reset to zero upon reset, access (read or write) of a channel's Mode register or Extended Mode register, or a Master Clear. The values upon reset are, disable chaining mode and generate IRQ13.

Bit	Description
7:5	**Reserved:** Must be 0.
4	**Buffer Expired Signal:** After one of the two buffers in the DMA expires then the DMA will inform the CPU that the next buffer should be loaded into the base register set. This bit determines whether IRQ13 or EOP should be used to inform the CPU that the buffer is complete; 1 = generate TC, 0 = Generate IRQ13; 1 = Programming complete, 0 = Don't start chaining.
3	**Base Register Programming:** After the reserve buffer's address and word count are written to the base register set, this bit should be set to 1 to inform the DMA that the second buffer is ready for transfer.
2	**Buffer Chaining Mode:** Bit 2 enables the chaining mode logic. If the bit is set to 1 after the initial DMA address and word count are programmed, then the Base address and word count are available for programming the next buffer in the chain. 1 = Enable chaining, 0 = Disable chaining.
1:0	**DMA Channel Select:** Bits[1:0] select the DMA channel mode register to program with bits[4:2]. **Bits[1:0] Channel** 00 0 or 4 01 1 or 5 10 2 or 6 11 3 or 7

3.2.18 CHAINSTA—CHAINING MODE STATUS REGISTER

Register Location: 04D4h
Default Value: 00h
Attribute: Read Only
Size: 8 Bits

This register is read only and is used to determine if chaining mode for a particular channel is enabled or disabled. A 1 read in this register indicates that the channel's chaining mode is enabled. A 0 indicates that the chaining mode is disabled. All Chaining mode bits are disabled after a reset with reset. After the DMA is used in Chaining mode the CPU will need to clear the Chaining mode enable bit if non-Chaining mode is desired.

Bit	Description
7:5, 3:0	**Chaining Mode Status:** If this bit is set to 1 then this channel has chaining enabled by writing 1 to bit 2 of the Chaining Mode Register. This bit can be reset to 0 by either writing a 0 to bit 2 of the Chaining Mode Register or reset being asserted or by a Master Clear Command.
4	**Reserved**

3.2.19 CHINTST—CHANNEL INTERRUPT STATUS REGISTER

Register Location: 040Ah
Default Value: 00h
Attribute: Read Only
Size: 8 Bits

Channel Interrupt Status is a read only register and is used to indicate the source (channel) of a DMA chaining interrupt on IRQ13. The DMA controller asserts IRQ13 after reaching terminal count, with chaining mode enabled. It does not assert IRQ13 during the initial programming sequence that loads the Base registers. After a reset, a read of this register will produce 00h.

Bit	Description
7:5, 3:0	**Chaining Interrupt Status:** When a channel interrupt status read returns a 0, bits[7:5,3:0] indicates that channel did not assert IRQ13. When a channel interrupt status read returns a 1, then that channel asserted IRQ13 after reaching a Terminal Count.
4	**Reserved**

3.2.20 CHAINBEC—CHAIN BUFFER EXPIRATION CONTROL REGISTER

Register Location: 040Ch
Default Value: 00h
Attribute: Read Only
Size: 8 Bits

This register is read only and reflects the outcome of the expiration of a chain buffer. A Chain Buffer Expiration Control register bit with 0 indicates the DMA controller asserts IRQ13 when the DMA controller reaches terminal count. A 1 indicates the DMA controller asserts TC when the DMA controller reaches terminal count. This bit is programmed in bit 4 of the Chaining Mode register.

Bit	Description
7:5, 3:0	**Chaining Buffer Expired:** When a chain buffer expiration control read returns 0, bit[7:5,3:0] indicates that Channel [7:5,3:0] will assert IRQ13 when the DMA channel reaches terminal count. When a chain buffer expiration control read returns 1, bit[7:5,3:0] indicates that Channel [7:5,3:0] will assert TC when the DMA controller reaches terminal count. This bit will reset to 0 following a reset.
4	**Reserved**

3.2.21 SCATGA—SCATTER-GATHER COMMAND REGISTER

Register Location: 0410h—Channels 0
0411h—Channels 1
0412h—Channels 2
0413h—Channels 3
0415h—Channels 5
0416h—Channels 6
0417h—Channels 7
Default Value: 00xxxx00b
Attribute: Write Only, Relocateable
Size: 8 Bits

The Scatter-Gather command register controls operation of the Descriptor Table aspect of S-G transfers. The S-G command register is write only. The current S-G transfer status can be read in the S-G channel's corresponding S-G Status register. The S-G command register can initiate a S-G transfer, and stop a transfer.

Scatter-Gather commands are issued with command codes. Bits[1:0] are used to implement the code mechanism. The S-G codes are described in the table below. Bit 7 is used to control the IRQ13/EOP assertion that follows a terminal count. Bit 6 controls the effect of bit 7. Common Scatter-Gather command writes are listed in Table 2.

Table 2. Scatter Gather Command Bits

Command	Bits	
	7654	3210
No S-G operation (S-G NOOP)	0000	0000b
Start S-G	xx00	0001b
Stop S-G	xx00	0010b
Issue IRQ13 on Terminal Count	0100	00xxb
Issue EOP on Terminal Count	1100	00xxb

Note that the "x" don't care states in Table 2 do not preclude programming those bits during the command write. For instance, for any S-G command code on bits[1:0], an optional selection of IRQ13 or EOP can take place if bit 7 is set to 1 and the appropriate choice is made for bit 6. All 0's in the command byte indicate an S-G NOOP: no S-G command is issued, and EOP/IRQ13 modification is disabled. Note that an EOP/IRQ13 modification can be made while disabling the S-G command bits (bits[1:0] = 00b); conversely, an S-G command may be issued while EOP/IRQ13 modification is disabled (bit 6 = 0b). After a reset, or Master Clear, IRQ13 is disabled and EOP is enabled.

The Start command assumes the Base and Current registers are both empty and will request a prefetch automatically. It also sets the status register to S-G Active, Base Empty, Current Empty, not Terminated, and Next Null Indicator to 0. The EOP/IRQ13 bit will still reflect the last value programmed.

Bit	Description
7	**EOP/IRQ13 Selection:** Bit 7 is used to select whether EOP or IRQ will be asserted at termination caused by the last buffer expiring. The last buffer can be either the last buffer in the list or the last buffer loaded in the DMA while it is suspended. If this bit is set to 1 then EOP will be asserted whenever the last buffer is completed. If this bit is set to 0 then IRQ13 will be asserted whenever the last buffer is completed. EOP can be used to alert an expansion bus I/O device that a scatter-gather termination condition was reached; the I/O device in turn can assert its own interrupt request line, and invoke a dedicated interrupt handling routine. IRQ13 should be used whenever the CPU needs to be notified directly. Following reset, or Master Clear, the value stored for this bit is 0, and IRQ13 is selected. Bit 6 must be set to a 1 to enable this bit during an S-G Command register write. When bit 6 is a 0 during the write, bit 7 will not have any effect on the current EOP/IRQ13 selection.
6	**Enable IRQ13/EOP Programming:** Enabling IRQ13/EOP programming allows initialization or modification of the S-G termination handling bits. If bit 5 is reset to 0, bit 7 will not have any effect on the state of IRQ13 or EOP assertion. When bit 5 is set to a 1, bit 7 determines the termination handling following a terminal count.
5:2	**Reserved**
1:0	**S-G Command Code** **Bits[1:0] Function** 00 No S-G command operation is performed. Bits[7:5] may still be used to program EOP/IRQ13 selection. 01 The Start command initiates the scatter-gather process. Immediately after the Start command is issued a request is issued to fetch the initial buffer to fill the Base Register set in preparation for performing a transfer. The Buffer Prefetch request has the same priority with respect to other channels as the DREQ it is associated with. Within the channel, DREQ is higher in priority than a prefetch request. 10 The Stop command halts a Scatter-Gather transfer immediately. When a Stop command is given, the Terminate bit in the S-G Status register and the DMA channel mask bit are both set. 11 Reserved The S-G Status register contains information on the S-G transfer status. This register maintains dynamic status information on S-G Transfer Activity, the Current and Base Buffer state, S-G Transfer Termination, and the End of the List indicator.

3.2.22 SCAGAST—SCATTER-GATHER STATUS REGISTER

Register Location: Channels 0
0419h—Channels 1
041Ah—Channels 2
041Bh—Channels 3
041Dh—Channels 5
041Eh—Channels 6
041Fh—Channels 7
Default Value: 08h
Attribute: Read Only, Relocatable
Size: 8 Bits

The Scatter-Gather Status Register provides Scatter-Gather process status information to the CPU or Master. An active bit is set to 1 after the S-G Start command is issued. The active bit will be 0 before the initial start command, following a terminal count, and after an S-G Stop command is issued. The Current Buffer and Base Buffer State Bits indicate whether the corresponding register has a buffer loaded. It is possible for the Base Buffer State to be set while the Current Buffer State is cleared. When the Current Buffer transfer is complete, the Base Buffer will not be moved into the Current Buffer until the start of the next data transfer. Thus, the Current Buffer State is empty (cleared), while the Base Buffer State is full (set). The Terminate bit is set active after a Stop command, after TC for the last buffer in the list and both Base and Current buffers have expired. The EOP and IRQ13 Bits indicate which end of process indicator will be used to alert the system of an S-G process termination. The EOL status bit is set if DMA controller has loaded the last buffer of the Link List.

Bit	Description
7	**Next Link Null Indicator:** If the Next SGD fetched from memory during a fetch operation has the EOL value (1), the current value of the Next Link register is not overwritten. Instead, bit 7 of the channel's S-G Status register, the Next Link Null indicator, is set to a 1. If the fetch returns a EOL value not equal to (1), this bit is reset to 0. This status bit is written after every fetch operation. Following reset, or Master Clear, this bit is reset to 0. This bit is also cleared by an S-G Start Command Write.
6	**Reserved**
5	**IRQ13 or EOP on Last Buffer:** When the IRQ13/EOP status bit is 1, EOP was either defaulted to at reset or selected through the S-G Command register as the S-G process termination indicator. EOP will be issued to alert the system when a terminal count occurs or following the Stop Command. When this bit is returned as a 0, an IRQ13 will be issued to alert the CPU of this same status.
4	**Reserved**
3	**S-G Base Buffer State:** When the Base Buffer status bit contains a 0, the Base Buffer is empty. When the Base Buffer Status bit is set to 1, the Base buffer has a buffer link loaded. Note that the Base Buffer State may be set while the Current buffer state is cleared. This condition occurs when the Current Buffer expires following a transfer; the Base Buffer will not be moved into the Current Register until the start of the next DMA transfer.
2	**S-G Current Buffer State:** When the Current Buffer status bit contains a 0, the Current Buffer is empty. When the Current Buffer status bit is set to 1, the Current Buffer has a buffer link loaded and is considered full. Following reset, bit 2 is reset to 0.
1	**Reserved**

Bit	Description
0	**S-G Active:** The Scatter-Gather Active bit indicates the current S-G transfer status. Bit 0 will be a 1 after a S-G Start Command is issued. Bit 0 will be a 0 before the Start command is issued. Bit 0 will be a 0 after terminal count on the last buffer on the channel is reached. Bit 0 will also be a 0 after a S-G Stop command has been issued. Following reset, or Master Clear, this bit is reset to 0.

3.2.23 SCAGAD—SCATTER-GATHER DESCRIPTOR TABLE POINTER REGISTER

Register Location: 0420h–0423h—Channels 0
0424h–0424h—Channels 1
0428h–042Bh—Channels 2
042Ch–042Ch—Channels 3
0434h–0437h—Channels 5
0438h–043Bh—Channels 6
043Ch–043Fh—Channels 7
Default Value: See below
Attribute: Read/Write, Relocateable
Size: 32 Bits

The SGD Table Pointer register contains the 32 bit pointer to the first SGD entry in the SGD table in memory. Before the start of a S-G transfer, this register should have been programmed to point to the first SGD in the SGD table. Following a "Start" command, it initiates reading the first SGD entry by pointing to the first SGD entry to be fetched from the memory. Subsequently, at the end of the each buffer block transfer, the contents of the SGD table pointer registers are incremented by 8 until the end of the SGD table is reached.

When programmed by the CPU, the SGD Table Pointer Registers can be programmed with a single 32-bit PCI write. Note that the PCEB and EISA Bus Controller will split the 32-bit write into four 8-bit writes.

Following a prefetch to the address pointed to by the channel's SGD table pointer register, the new Memory Address is loaded into the Base Address register, the new Byte Count is loaded into the Base Byte Count register, and the newly fetched Next SGD replaces the current Next SGD value.

The end of the SGD table is indicated by a End of Table field having a MSB equal to 1. When this value is read during a SGD fetch, the current SGD value is not replaced. Instead, bit 7 of the channel's status register is set to a 1 when the EOL is read from memory.

Bit	Description
31:0	**SGD Table Pointer:** The SGD table pointer register contains a 32-bit pointer to the main memory location where the software maintains the Scatter Gather Descriptors for the linked-list buffers. These bits are translated into A[31:0] signals for accessing memory on the PCI.

3.2.24 CBPFF—CLEAR BYTE POINTER FLIP FLOP REGISTER

Register Location: 00Ch—Channels 0-3
0D8h—Channels 4-7
Default Value: xxh
Attribute: Write Only
Size: 8 Bits

This command is executed prior to writing or reading new address or word count information to the DMA. This initializes the flip-flop to a known state so that subsequent accesses to register contents by the microprocessor will address upper and lower bytes in the correct sequence.

The Clear Byte Pointer command clears the internal latch used to address the upper or lower byte of the 16-bit address and Word Count registers. The latch is also cleared at power on by reset and by the Master Clear command. The Host CPU may read or write a 16-bit DMA controller register by performing two consecutive accesses to the I/O port. The Clear Byte Pointer command precedes the first access. The first I/O write to a register port loads the least significant byte, and the second access automatically accesses the most significant byte.

When the Host CPU is reading or writing DMA registers, two Byte Pointer Flip-Flops are used; one for Channels 0-3 and one for Channels 4-7. Both of these act independently. There are separate software commands for clearing each of them (0Ch for Channels 0-3, 0D8h for Channels 4-7).

Bit	Description
7:0	**Clear Byte Pointer FF:** No specific pattern. Command enabled with a write to the I/O port address.

3.2.25 DMC—DMA MASTER CLEAR REGISTER

Register Location: 00Dh—Channels 0-3
0DAh—Channels 4-7
Default Value: xxh
Attribute: Write Only
Size: 8 Bits

This software instruction has the same effect as the hardware Reset. The Command, Status, Request, and Internal First/Last Flip-Flop registers are cleared and the Mask register is set. The DMA controller will enter the idle cycle.

There are two independent Master Clear Commands, 0Dh which acts on Channels 0-3, and 0DAh which acts on Channels 4-7.

Bit	Description
7:0	**Master Clear:** No specific pattern. Command enabled with a write to the I/O port address.

3.2.26 DCM—DMA CLEAR MASK REGISTER

Register Location: 00Eh—Channels 0-3
0DCh—Channels 4-7
Default Value: xxh
Attribute: Write Only
Size: n/a

Software Command This command clears the mask bits of all four channels, enabling them to accept DMA requests. I/O port 0Eh is used for Channels 0-3 and I/O port 0DCh is used for Channels 4-7.

Bit	Description
7:0	**Clear Mask:** No specific pattern. Command enabled with a write to the I/O port address.

3.3 Timer Unit Registers

The ESC contains five counters that are equivalent to those found in the 82C54 Programmable Interval Timer. The Timer registers control these counters and can be accessed from the EISA Bus via I/O space. This section describes the counter/timer registers on the ESC. The counter/timer operations are further described in Section 8.0, Interval Timer

3.3.1 TCW—TIMER CONTROL WORD REGISTER

Register Location: 043h—Timer 1
04Bh—Timer 2
Default Value: xxh
Attribute: Write Only
Size: 8 Bits

The Timer Control Word specifies the counter selection, the operating mode, the counter byte programming order and size of the COUNT value, and whether it counts down in a 16-bit or binary-coded decimal (BCD) format. After writing the control word, a new count may be written at any time. The new value will take effect according to the programmed mode.

There are six programmable counting modes. Typically, the ESC Timer Unit Counters 0 and 2 are programmed for Mode 3, the Square Wave Mode, while Counter 1 is programmed in Mode 2, the Rate Generator Mode.

Two special commands are selected through the Control Word Register. The Counter Latch Command is selected when bits[5:4] are both 0. The Read-Back Command is selected when bits[7:6] are both 1. When either of these two commands are selected with the Control Word Register, the meaning of the other bits in the register changes. Both of these special commands, and the respective changes they make to the bit definitions in this register, are covered in detail under separate register descriptions later in this section.

Bits 4 and 5 are also used to select the count register programming mode. The programming process is simple:

1. Write a control word.
2. Write an initial count for each counter.
3. Load the LSB, MSB, or LSB then MSB.

The read/write selection chosen with the control word dictates the programming sequence that must follow when initializing the specified counter.

If a counter is programmed to read/write two-byte counts, the following precaution applies: A program must not transfer control between writing the first and second byte to another routine which also writes into that same counter. Otherwise, the counter will be loaded with an incorrect count. The count must always be completely loaded with both bytes.

Bits 6 and 7 are also used to select the counter for the control word you are writing.

Following reset, the control words for each register are undefined. You must program each timer to bring it into a known state. However, each counter OUT signal is reset to 0 following reset. The SPKR output, interrupt controller input IRQ0 (internal), bit 5 of port 061h, and the internally generated Refresh request are each reset to 0 following reset.

Bit	Description
7:6	**Counter Select:** The Counter Selection bits select the counter the control word acts upon as shown below. The Read Back Command is selected when bits[7:6] are both 1. **Bit[7:6] Function** 00 Counter 0 select 01 Counter 1 select 10 Counter 2 select 11 Read Back Command (see Section 3.3.2)
5:4	**Read/Write Select:** Bits[5:4] are the read/write control bits. The Counter Latch Command is selected when bits[5:4] are both 0. The read/write options include read/write least significant byte, read/write most significant byte, or read/write the LSB and then the MSB. The actual counter programming is done through the counter I/O port (040h, 041h, and 042h for counters 0, 1, and 2, respectively). **Bit[5:4] Function** 00 Counter Latch Command (see Section 3.3.3) 01 R/W Least Significant Byte (LSB) 10 R/W Most Significant Byte (MSB) 11 R/W LSB then MSB
3:1	**Counter Mode Selection:** Bits[3:1] select one of six possible modes of operation for the counter as shown below. Note that for the fail safe timer (timer 2, counter 0), modes 1, 2, 3, 4, and 5 are reserved. **Bit[3:1] Mode Function** 000 0 Out signal on end of count (=0) 001 1 Hardware retriggerable one-shot (Reserved for timer 2, counter 0.) X10 2 Rate generator (divide by n counter) (Reserved for timer 2, counter 0.) X11 3 Square wave output (Reserved for timer 2, counter 0.) 100 4 Software triggered strobe (Reserved for timer 2, counter 0.) 101 5 Hardware triggered strobe (Reserved for timer 2, counter 0.)
0	**Binary/BCD Countdown Select:** When bit 0=0, a binary countdown is used. The largest possible binary count is 2^{16}. When bit 0=1, a binary coded decimal (BCD) count is used. The largest BCD count allowed is 104.

3.3.2 TIMER READ BACK COMMAND REGISTER

Register Location: 043h—Timer 1
04Bh—Timer 2
Default Value: xxh
Attribute: Write Only
Size: 8 Bits

The Read-Back command is used to determine the count value, programmed mode, and current states of the OUT pin and Null Count flag of the selected counter or counters. The Read-Back command is written to the Control Word register, which latches the current states of the above mentioned variables. The value of the counter and its status may then be read by I/O access to the counter address.

Status and/or count may be latched on one, two, or all three of the counters by selecting the counter during the write. The Count latched will stay latched until read, regardless of further latch commands. The count must be read before newer latch commands latch a new count. The Status latched by the read-back command will also remain latched until after a read to the counter's I/O port. To reiterate, the Status and Count are unlatched only after a counter read of the Status register, the Count register, or the Status and Count register in succession.

Both count and status of the selected counter(s) may be latched simultaneously by setting both the COUNT# and STATUS# bits [5:4] = 00b. This is functionally the same as issuing two consecutive, separate read-back commands. As stated above, if multiple count and/or status read-back commands are issued to the same counter(s) without any intervening reads, all but the first are ignored.

If both count and status of a counter are latched, the first read operation from that counter will return the latched status, regardless of which was latched first. The next one or two reads (depending on whether the counter is programmed for one or two byte counts) return the latched count. Subsequent reads return an unlatched count.

A register description of the Status Byte read follows later in this section. Note that bit definitions for a write to this port changed when the read-back command was selected, when compared to a normal control word write to this same port.

Bit	Description
7:6	**Read Back Command:** When bits[7:6] are both 1, the read-back command is selected during a write to the control word. The normal meanings (mode, countdown, r/w select) of the bits in the control register at I/O address 043h change when the read-back command is selected. Following the read-back command, I/O reads from the selected counter's I/O addresses produce the current latch status, the current latched count, or both if bits 4 and 5 are both 0.
5	**Latch Status of Selected Counters:** When bit 5 is a 1, the Current Count value of the selected counters will be latched. When bit 4 is a 0, the Status will not be latched.
4	**Latch Count of Selected Counters:** When bit 4 is a 1, the Status of the selected counters will be latched. When bit 4 is a 0, the Status will not be latched. The Status byte format is described in the next register description.
3	**Counter 2:** Counter 2 is selected for the latch command selected with bits 4 and 5 if bit 3 is a 1. If bit 3 is a 0, Status and/or Count will not be latched.
2	**Counter 1:** Counter 1 is selected for the latch command selected with bits 4 and 5 if bit 2 is a 1. If bit 2 is a 0, Status and/or Count will not be latched.
1	**Counter 0:** Counter 0 is selected for the latch command selected with bits 4 and 5 if bit 1 is a 1. If bit 1 is a 0, Status and/or Count will not be latched.
0	**Reserved:** Must be 0.

3.3.3 COUNTER LATCH COMMAND REGISTER

Register Location: 043h—Timer 1
04Bh—Timer 2
Default Value: xxh
Attribute: Write Only
Size: 8 Bits

The Counter Latch command latches the current count value at the time the command is received. This command is used to insure that the count read from the counter is accurate (particularly when reading a two-byte count). The count value is then read from each counter's Count register. One, two or all three counters may be latched with one counter latch command.

If a Counter is latched once and then, some time later, latched again before the count is read, the second Counter Latch Command is ignored. The count read will be the count at the time the first Counter Latch Command was issued.

The count must be read according to the programmed format. Specifically, if the Counter is programmed for two byte counts, two bytes must be read. The two bytes do not have to be read one right after the other; read, write, or programming operations for other Counters may be inserted between them.

One precaution is worth noting. If a Counter is programmed to read/write two-byte counts, a program must not transfer control between reading the first and second byte to another routine which also reads from that same Counter. Otherwise, an incorrect count will be read. Finish reading the latched two-byte count before transferring control to another routine.

Note that bit definitions for a write to this port have changed when the read-back command was selected, when compared to a normal control word write to this same port.

Bit	Description
7:6	**Counter Selection:** Bits 6 and 7 are used to select the counter for latching. **Bit[7:6] Function** 00 Latch counter 0 select 01 Latch counter 1 select 10 Latch counter 2 select 11 Read Back Command select
5:4	**Specifies Counter Latch Command:** When bits[5:4] are both 0, the Counter Latch command is selected during a write to the control word. The normal meanings (mode, countdown, r/w select) of the bits in the control register at I/O address 043h change when the Counter Latch command is selected. Following the Counter Latch command, I/O reads from the selected counter's I/O addresses produce the current latched count.
3:0	**Reserved:** Must be 0.

3.3.4 TMSTAT—TIMER STATUS BYTE FORMAT REGISTER

Register Location: 040h—Timer 1, Counter 0
041h—Timer 1, Counter 1
042h—Timer 1, Counter 2
048h—Timer 2, Counter 0
04Ah—Timer 2, Counter 2
Default Value: 0xxxxxxxb
Attribute: Read Only
Size: 8 Bits per counter

Each Counter's Status Byte may be read following an Timer Read-Back Command. The Read-Back command is programmed through the counter control register. If "Latch Status" is chosen as a Read-Back option for a given counter, the next read from the counter's I/O port address returns the Status byte.

The Status byte returns the countdown type, either BCD or binary; the Counter Operational Mode; the Read/Write Selection status; the Null count, also referred to as the Count Register Status; and the current State of the counter OUT pin.

Bit	Description
7	**Counter OUT Pin State:** When this bit is a 1, the OUT pin of the counter is also a 1. When this bit is a 0, the OUT pin of the counter is also a 0.
6	**Count Register Status:** Also referred to as Null Count, indicates when the last count written to the Count Register (CR) has been loaded into the counting element (CE). The exact time this happens depends on the counter Mode and is described in the Mode definitions, but until the count is loaded into the counting element (CE), it can't be read from the counter. If the count is latched or read before the load time, the count value returned will not reflect the new count written to the register. When bit 6 is a 0, the count has been transferred from CR to CE and is available for reading. When bit 6 is a 1, the Null count condition exists. The count has not been transferred from CR to CE and is not yet available for reading.
5:4	**Read/Write Status:** Bits[5:4] reflect the read/write selection made through bits[5:4] of the control register. The binary codes returned during the status read match the codes used to program the counter read/write selection.
3:1	**Mode Selection Status:** Bits[3:1] return the counter mode programming. The binary code returned matches the code used to program the counter mode, as listed under the bit function above.
0	**Countdown Type Status:** Bit 0 reflects the current countdown type, either 0 for binary countdown or a 1 for binary coded decimal (BCD) countdown.

3.3.5 CAPS—COUNTER ACCESS PORTS

Register Location: 040h—Timer 1, Counter 0
041h—Timer 1, Counter 1
042h—Timer 1, Counter 2
048h—Timer 2, Counter 0
04Ah—Timer 2, Counter 2
Default Value: xxh
Attribute: Read/Write
Size: 8 Bits per counter

Each of these I/O ports is used for writing count values to the count registers; reading the current count value from the counter by either an I/O read, after a counter-latch command, or after a read-back command; and reading the Status byte following a read-back command.

Bit	Description
7:0	**Counter Access:** Each counter I/O port address is used to program the 16 bit count register. The order of programming, either LSB only, MSB only, or LSB then MSB, is defined with the Counter Control register at I/O port address 043h. The counter I/O port is also used to read the current count from the count register, and return the status of the counter programming following a read-back command.

3.4 Interrupt Controller Registers

The ESC contains an EISA compatible interrupt controller that incorporates the functionality of two 82C59 interrupt controllers. The interrupt registers control the operation of the interrupt controller and can be accessed from the EISA Bus via I/O space. This section describes the Interrupt registers. The operation of the Interrupt Controller is described in Chapter 9.0.

3.4.1 ICW1—INITIALIZATION COMMAND WORD 1

Register Location: 020h—INT CNTRL-1
0A0h—INT CNTRL-2
Default Value: xxh
Attribute: Write Only
Size: 8 Bits per controller

A write to Initialization Command Word One starts the interrupt controller initialization sequence. Addresses 020h and 0A0h are referred to as the base addresses of CNTRL-1 and CNTRL-2 respectively.

An I/O write to the CNTRL-1 or CNTRL-2 base address with bit 4 equal to 1 is interpreted as ICW1. For ESC-based EISA systems, three I/O writes to "base address + 1" must follow the ICW1. The first write to "base address + 1" performs ICW2, the second write performs ICW3, and the third write performs ICW4.

ICW1 starts the initialization sequence during which the following automatically occur:

a. The edge sense circuit is reset, which means that following initialization, an interrupt request (IRQ) input must make a low-to-high transition to generate an interrupt.

b. The Interrupt Mask register is cleared.

c. IRQ7 input is assigned priority 7.

d. The slave mode address is set to 7.

e. Special Mask Mode is cleared and Status Read is set to IRR.

f. If IC4 was set to 0, then all functions selected by ICW4 are set to zero. However, ICW4 must be programmed in the ESC implementation of this interrupt controller, and IC4 must be set to a 1.

ICW1 has three significant functions within the ESC interrupt controller configuration. ICW4 is needed, so bit 0 must be programmed to a 1. There are two interrupt controllers in the system, so bit 1, SNGL, must be programmed to a 0 on both CNTRL-1 and CNTRL-2, to indicate a cascade configuration. Bit 4 must be a 1 when programming ICW1. OCW2 and OCW3 are also addressed at the same port as ICW1. This bit indicates that ICW1, and not OCW2 or OCW3, will be programmed during the write to this port.

Bit 2, ADI, and bits[7:5], A7-A5, are specific to an MSC-85 implementation. These bits are not used by the ESC interrupt controllers. Bits[7:5,2] should each be initialized to 0.

Bit	Description
7:5	**Reserved:** A7-A5 are MCS-85 implementation specific bits. They are not needed by the ESC. These bits should be 000b when programming the ESC.
4	**ICW/OCW Select:** Bit 4 must be a 1 to select ICW1. After the fixed initialization sequence to ICW1, ICW2, ICW3, and ICW4, the controller base address is used to write to OCW2 and OCW3. Bit 4 is a 0 on writes to these registers. A 1 on this bit at any time will force the interrupt controller to interpret the write as an ICW1. The controller will then expect to see ICW2, ICW3, and ICW4.
3	**Reserved:** This bit is not used in the ESC.
2	**Reserved:** ADI ignored for the ESC.
1	**SNGL:** This bit must be programmed to a 0 to indicate that two interrupt controllers are operating in cascade mode on the ESC.
0	**IC4:** This bit must be set to a 1. IC4 indicates that ICW4 needs to be programmed. The ESC requires that ICW4 be programmed to indicate that the controllers are operating in an 80x86 type system.

3.4.2 ICW2—INITIALIZATION COMMAND WORD 2

Register Location: 021h—INT CNTRL-1
0A1h—INT CNTRL-2I
Default Value: xxh
Attribute: Write Only
Size: 8 Bits per controller

ICW2 is used to initialize the interrupt controller with the five most significant bits of the interrupt vector address. The value programmed for bits[7:3] is used by the CPU to define the base address in the interrupt vector table for the interrupt routines associated with each IRQ on the controller. Typical ISA ICW2 values are 04h for CNTRL-1 and 70h for CNTRL-2. Section 9.8.1 of the Interrupt Unit Functional Description contains a table detailing the interrupt vectors for each interrupt request level, as they would appear when the vector is driven onto the data bus.

Bit	Description
7:3	**Interrupt Vector Base Address:** Bits[7:3] define the base address in the interrupt vector table for the interrupt routines associated with each interrupt request level input. For CNTRL-1, a typical value is 00001b, and for CNTRL-2, 10000b. The interrupt controller combines a binary code representing the interrupt level to receive service with this base address to form the interrupt vector that is driven out onto the bus. For example, the complete interrupt vector for IRQ[0] (CNTRL-1), would be 0000 1000b (CNTRL-1 [7:3] = 00001b and 000b representing IRQ[0]). This vector is used by the CPU to point to the address information that defines the start of the interrupt routine.
2:0	**Interrupt Request Level:** When writing ICW2, these bits should all be 0. During an interrupt acknowledge cycle, these bits will be programmed by the interrupt controller with the interrupt code representing the interrupt level to be serviced. This interrupt code is combined with bits[7:3] to form the complete interrupt vector driven onto the data bus during the second INTA# cycle. The table in Section 9.8.1 outlines each of these codes. The code is a simple three bit binary code: 000b represents IRQ0 (IRQ8), 001b IRQ1 (IRQ9), 010b IRQ2 (IRQ10), and so on until 111b IRQ7 (IRQ15).

3.4.3 ICW3—INITIALIZATION COMMAND WORD 3 (MASTER)

Register Location: 021h—INT CNTRL-1
Default Value: xxh
Attribute: Write Only
Size: 8 Bits

The meaning of ICW3 differs between CNTRL-1 and CNTRL-2. On CNTRL-1, the master controller, ICW3 indicates which CNTRL-1 IRQ line physically connects the INT output of CNTRL-2 to CNTRL-1. ICW3 must be programmed to 04h, indicating the cascade of the CNTRL-2 INT output to the IRQ[2] input of CNTRL-1.

An interrupt request on IRQ2 causes CNTRL-1 to enable CNTRL-2 to present the interrupt vector address during the second interrupt acknowledge cycle.

Bit	Description
7:3, 1:0	**Cascade Interrupt Controller IRQs:** Bits[7:3] and bits[1:0] must be programmed to 0.
2	**Cascade Interrupt Controller IRQs:** Bit 2 must always be programmed to a 1. This bit indicates that CNTRL-2, the slave controller, is cascaded on interrupt request line two (IRQ[2]). When an interrupt request is asserted to CNTRL-2, the IRQ goes through the priority resolver. After the slave controller priority resolution is finished, the INT output of CNTRL-2 is asserted. However, this INT assertion does not go directly to the CPU. Instead, the INT assertion cascades into IRQ[2] on CNTRL-1. IRQ[2] must go through the priority resolution process on CNTRL-1. If it wins the priority resolution on CNTRL-1 and the CNTRL-1 INT signal is asserted to the CPU, the returning interrupt acknowledge cycle is really destined for CNTRL-2. The interrupt was originally requested at CNTRL-2, so the interrupt acknowledge is destined for CNTRL-2, and not a response for IRQ[2] on CNTRL-1. When an interrupt request from IRQ[2] wins the priority arbitration, in reality an interrupt from CNTRL-2 has won the arbitration. Because bit 2 of ICW3 on the master is set to 1, the master knows which identification code to broadcast on the internal cascade lines, alerting the slave controller that it is responsible for driving the interrupt vector during the second INTA# pulse.

3.4.4 ICW3—INITIALIZATION COMMAND WORD 3 (SLAVE)

Register Location: INT CNTRL-2 port address-0A1h
Default Value: xxh
Attribute: Write Only
Size: 8 Bits

On CNTRL-2 (the slave controller), ICW3 is the slave identification code broadcast by CNTRL-1 from the trailing edge of the first INTA# pulse to the trailing edge of the second INTA# pulse. CNTRL-2 compares the value programmed in ICW3 with the incoming identification code. The code is broadcast over three ESC internal cascade lines. ICW3 must be programmed to 02h for CNTRL-2. When 010b is broadcast by CNTRL-1 during the INTA# sequence, CNTRL-2 assumes responsibility for broadcasting the interrupt vector during the second interrupt acknowledge cycle.

As an illustration, consider an interrupt request on IRQ[2] of CNTRL-1. By definition, a request on IRQ[2] must have been asserted by CNTRL-2. If IRQ[2] wins the priority resolution on CNTRL-1, the interrupt acknowledge cycle returned by the CPU following the interrupt is destined for CNTRL-2, not CNTRL-1. CNTRL-1 will see the INTA# signal, and knowing that the actual destination is CNTRL-2, will broadcast a slave identification code across the internal cascade lines. CNTRL-2 will compare this incoming value with the 010b stored in ICW3. Following a positive decode of the incoming message from CNTRL-1, CNTRL-2 will drive the appropriate interrupt vector onto the data bus during the second interrupt acknowledge cycle.

Bit	Description
7:3	**Reserved:** Must be 0.
2:0	**Slave Identification Code:** The Slave Identification code must be programmed to 010b during the initialization sequence. The code stored in ICW3 is compared to the incoming slave identification code broadcast by the master controller during interrupt acknowledge cycles.

3.4.5 ICW4—INITIALIZATION COMMAND WORD 4

Register Location: 021h—INT CNTRL-1
0A1h—INT CNTRL-2
Default Value: xxh
Attribute: Write Only
Size: 8 Bits

Both ESC interrupt controllers must have ICW4 programmed as part of their initialization sequence. Minimally, the microprocessor mode bit, bit 0, must be set to a 1 to indicate to the controller that it is operating in an 80x86 based system. Failure to program this bit will result in improper controller operation during interrupt acknowledge cycles. Additionally, the Automatic End of Interrupt (AEOI) may be selected, as well as the Special Fully Nested Mode (SFNM) of operation.

The default programming for ICW4 is 01h, which selects 80x86 mode, normal EOI, buffered mode, and special fully nested mode disabled.

Bits 2 and 3 must be programmed to 0 for the ESC interrupt unit to function correctly.

Both bit 1, AEOI, and bit 4, SFNM, can be programmed if the system developer chooses to invoke either mode.

Bit	Description
7:5	**Reserved:** Must be 0.
4	**SFNM:** Bit 4, SFNM, should normally be disabled by writing a 0 to this bit. If SFNM = 1, the special fully nested mode is programmed.
3:2	**Master/Slave Buffer Mode (BUF):** Bit 3, BUF, must be programmed to 0 for the ESC. This is non-buffered mode. While different programming options are sometimes offered for bits 2 and 3, within the ESC interrupt unit, bits 2 and 3 must always be programmed to 00b.
1	**AEOI:** Bit 1, AEOI, should normally be programmed to 0. This is the normal end of interrupt. If AEOI = 1, the automatic end of interrupt mode is programmed.
0	**Microprocessor Mode:** The Microprocessor Mode bit must be programmed to 1 to indicate that the interrupt controller is operating in an 80x86 based system. Never program this bit to 0.

3.4.6 OCW1—OPERATION CONTROL WORD 1

Register Location: 021h—INT CNTRL-1
0A1h—INT CNTRL-2
Default Value: xxh
Attribute: Read/Write
Size: 8 Bits

OCW1 sets and clears the mask bits in the interrupt Mask register (IMR). Each interrupt request line may be selectively masked or unmasked any time after initialization. A single byte is written to this register. Each bit position in the byte represents the same-numbered channel: Bit 0 = IRQ[0], bit 1 = IRQ[1] and so on. Setting the bit to a 1 sets the mask, and clearing the bit to a 0 clears the mask. Note that masking IRQ[2] on CNTRL-1 will also mask all of controller 2's interrupt requests (IRQ8-IRQ15). Reading OCW1 returns the controller's mask register status.

The IMR stores the bits which mask the interrupt lines to be masked. The IMR operates on the IRR. Masking of a higher priority input will not effect the interrupt request lines of lower priority.

Unlike status reads of the ISR and IRR, for reading the IMR, no OCW3 is needed. The output data bus will contain the IMR whenever I/O read is active and the I/O port address is 021h or 0A1h (OCW1).

All writes to OCW1 must occur following the ICW1-ICW4 initialization sequence, since the same I/O ports are used for OCW1, ICW2, ICW3 and ICW4.

Bit	Description
7:0	**Interrupt Request Mask:** When a 1 is written to any bit in this register, the corresponding IRQ[x] line is masked. For example, if bit 4 is set to a 1, then IRQ[4] will be masked. Interrupt requests on IRQ[4] will not set Channel 4's interrupt request register (IRR) bit as long as the channel is masked. When a 0 is written to any bit in this register, the corresponding IRQ[x] mask bit is cleared, and interrupt requests will again be accepted by the controller. Note that masking IRQ[2] on CNTRL-1 will also mask the interrupt requests from CNTRL-2, which is physically cascaded to IRQ[2].

3.4.7 OCW2—OPERATION CONTROL WORD 2

Register Location: 020h—INT CNTRL-1
0A0h—INT CNTRL-2
Default Value: xxh
Attribute: Write Only
Size: 8 Bits

OCW2 controls both the Rotate Mode and the End of Interrupt Mode, and combinations of the two. The three high order bits in an OCW2 write represent the encoded command. The three low order bits are used to select individual interrupt channels during three of the seven commands. The three low order bits (labeled L2, L1 and L0) are used when bit 6, the SL bit, is set to a 1 during the command.

Following a reset and ICW initialization, the controller enters the fully nested mode of operation. Non-specific EOI without rotation is the default. Both rotation mode and specific EOI mode are disabled following initialization.

Bit	Description
7:5	**Rotate and EOI Codes (R, SL, EOI):** These three bits control the Rotate and End of Interrupt modes and combinations of the two. A chart of these combinations is listed above under the bit definition. **Bits[7:5] Function** 001 Non-specific EOI command 011 Specific EOI Command 101 Rotate on Non-Specific EOI Command 100 Rotate in Auto EOI Mode (Set) 000 Rotate in Auto EOI Mode (Clear) 111 *Rotate on Specific EOI Command 110 *Set Priority Command 010 No Operation * L0 - L2 Are Used
4:3	**OCW2 Select:** When selecting OCW2, bits 3 and 4 must both be 0. If bit 4 is a 1, the interrupt controller interprets the write to this port as an ICW1. Therefore, always ensure that these bits are both 0 when writing an OCW2.
2:0	**Interrupt Level Select (L2, L1, L0):** L2, L1, and L0 determine the interrupt level acted upon when the SL bit is active. A simple binary code, outlined above, selects the channel for the command to act upon. When the SL bit is inactive, these bits do not have a defined function; programming L2, L1 and L0 to 0 is sufficient in this case. **Bit[2:0] Interrupt Level** 000 IRQ 0(8) 001 IRQ 1(9) 010 IRQ 2(10) 011 IRQ 3(11) 100 IRQ 4(12) 101 IRQ 5(13) 110 IRQ 6(14) 111 IRQ 7(15)

3.4.8 OCW3—OPERATION CONTROL WORD 3

Register Location: 020h—INT CNTRL-1
0A0h—INT CNTRL-2
Default Value: x01xxx10b
Attribute: Read/Write
Size: 8 Bits

OCW3 serves three important functions; Enable Special Mask Mode, Poll Mode control, and IRR/ISR register read control.

First, OCW3 is used to set or reset the Special Mask Mode (SMM). The Special Mask Mode can be used by an interrupt service routine to dynamically alter the system priority structure while the routine is executing, through selective enabling/disabling of the other channel's mask bits.

Second, the Poll Mode is enabled when a write to OCW3 is issued with Bit 2 equal to 1. The next I/O read to the interrupt controller is treated like an interrupt acknowledge; a binary code representing the highest priority level interrupt request is released onto the bus.

Third, OCW3 provides control for reading the In-Service Register (ISR) and the Interrupt Request Register (IRR). Either the ISR or IRR is selected for reading with a write to OCW3. Bits 0 and 1 carry the encoded command to select either register. The next I/O read to the OCW3 port address will return the register status specified during the previous write. The register specified for a status read is retained by the interrupt controller. Therefore, a write to OCW3 prior to every status read command is unnecessary, provided the status read desired is from the register selected with the last OCW3 write.

Bit	Description
7	**Reserved:** Must be 0.
6	**SMM:** If ESMM = 1 and SMM = 1 the Interrupt Controller will enter Special Mask Mode. If ESMM = 1 and SMM = 0 the Interrupt Controller will revert to normal mask mode. When ESMM = 0, SMM has no effect.
5	**Enable Special Mask Mode:** When this bit is set to 1 it enables the SMM bit to set or reset the Special Mask Mode. When ESMM = 0 the SMM bit becomes a "don't care".
4:3	**OCW3 Select:** When selecting OCW3, bit 3 must be a 1 and bit 4 must be 0. If bit 4 is a 1, the Interrupt Controller interprets the write to this port as an ICW1. Therefore, always ensure that bits[4:3] are "01b" when writing an OCW3.
2	**Poll Mode Command:** When bit 2 is a 0, the Poll command is not issued. When bit 2 is a 1, the next I/O read to the Interrupt Controller is treated as an Interrupt Acknowledge cycle. An encoded byte is driven onto the data bus, representing the highest priority level requesting service.
1:0	**Register Read Command:** Bits[1:0] provide control for reading the In-Service Register (ISR) and the Interrupt Request Register (IRR). When bit 1 = 0, bit 0 will not effect the register read selection. When bit 1 = 1, bit 0 selects the register status returned following an OCW3 read. If bit 0 = 0, the IRR will be read. If bit 0 = 1, the ISR will be read. Following ICW initialization, the default OCW3 port address read will be "read IRR". To retain the current selection (read ISR or read IRR), always write a 0 to bit 1 when programming this register. The selected register can be read repeatedly without reprogramming OCW3. To select a new status register, OCW3 must be reprogrammed prior to attempting the read. **Bits[1:0] Function** 00 No Action 01 No Action 10 Read IRQ Register 11 Read IS Register

3.4.9 ELCR—EDGE/LEVEL CONTROL REGISTER

Register Location: 04D0h—INT CNTRL-1
04D1h—INT CNTRL-1
Default Value: 00h
Attribute: Read/Write
Size: 8 Bits

The Edge/Level Control Register is used to set the interrupts to be triggered by either the signal edge or the logic level. INT0, INT1, INT2, INT8, INT13 must be set to edge sensitive. After a reset all the INT signals are set to edge sensitive.

Programming Considerations:
If an interrupt is switched from level to edge sensitive, a false interrupt is generated on that interrupt line. If the IRQx line is high, then switching the level/edge bet from a 1 to a 0 causes the interrupt controller to detect an interrupt. Also note that even if this interrupt is masked when programming this register, the interrupt controller still latches the false interrupt. As soon as this interrupt is unmasked, the false interrupt is processed.

Thus, before switching the edge/level function, disable interrupts to the processor (either mask interrupts or CLI instruction). Then program the ELCR Register. Finally, re-initialize the interrupt controller to clear the false interrupt.

Bit	Description
7:0	**Edge/Level Select:** The bits select if the interrupts are triggered by either the signal edge or the logic level. A 0 bit represents an edge sensitive interrupt, and a 1 is for level sensitive. Bit[2:0] and bit 13 must be set to 0. After A reset or power-on these registers are set to 00h.

Bit	Port 04D0h	Port 04D1h
0	INT0	INT8
1	INT1	INT9
2	INT2	INT10
3	INT3	INT11
4	INT4	INT12
5	INT5	INT13
6	INT6	INT14
7	INT7	INT15

3.4.10 NMISC—NMI STATUS AND CONTROL REGISTER

Register Location: 061h
Default Value: X0X0 0000
Attribute: Read/Write, Read Only
Size: 8 Bits

This register is used to check the status of different system components, control the output of the Speaker Counter (Timer 1, Counter 2), and gate the counter output that drives the SPKR signal. This register also controls NMI generation and reports NMI source status. Note that NMI generation is globally enabled/disabled via the NMIERTC Register and NMI generation for SERR# is controlled via the MS Register. Bits[7:4] of this register are read-only and must be written as 0s when writing to this register. Bits[3:0] are read/write. Following reset, bit 7 returns the PCI System Board Parity Error status (PERR#) and bit 5 is undetermined until Counter 2 is properly programmed.

Bit	Description
7	**System Board Error—RO:** Bit 7 is set if the PERR# line is pulsed. This interrupt is enabled by setting bit 2 to 0. To reset the interrupt, set bit 2 to 0 and then set it to 1. Note that this bit does not reflect status of an NMI caused by SERR#, which is enabled and disabled/cleared via the MS Register.
6	**IOCHK# NMI Source—RO:** Bit 6 is set if an expansion board asserts IOCHK# on the ISA/EISA Bus. This interrupt is enabled by setting bit 3 to 0. To reset the interrupt, set bit 3 to 0 and then set it to 1.
5	**Timer 1, Counter 2—RO:** The Timer 1, Counter 2 OUT signal state is reflected in bit 5. The value on this bit following a read is the current state of the Counter 2 OUT signal. Counter 2 must be programmed following a reset for this bit to have a determinate value.
4	**Refresh Cycle Toggle—RO:** The Refresh Cycle Toggle signal toggles from either 0 to 1 or 1 to 0 following every refresh cycle.
3	**IOCHK# NMI Enable—R/W:** When bit 3 is a 1, IOCHK# NMI's are disabled and cleared, and when bit 3 is a 0 (default), IOCHK# NMI's are enabled.
2	**PCI System Board Error—R/W:** When bit 2 is a 1, the system board error is disabled and cleared. When bit 2 is a 0 (default), the system board parity error is enabled. Note that NMI generation for system board errors is enabled/disabled via bit 3 (System Error) of the Mode Select Register. Following reset, bit 2 is a 0, and system board errors are enabled.
1	**Speaker Data Enable—R/W:** Speaker Data Enable is ANDed with the Timer 1, Counter 2 OUT signal to drive the SPKR output signal. When bit 1 is a 0 (default), the result of the AND is always 0 and the SPKR output is always 0. When bit 1 is a 1, the SPKR output is equivalent to the Counter 2 OUT signal value.
0	**Timer 1, Counter 2 Gate Enable—R/W:** When bit 0 is a 0, Timer 1, Counter 2 counting is disabled. Counting is enabled when bit 0 is a 1. This bit controls the GATE input to Counter 2.

3.4.11 NMIERTC—NMI CONTROL AND REAL-TIME CLOCK ADDRESS

Register Location: 070h
Default Value: See below
Attribute: Write Only
Size: 8 Bits

The most-significant bit enables or disables all NMI sources including PERR#, SERR#, IOCHK#, Fail-Safe Timer, Bus Timeout, and the NMI Port. Write an 80h to The NMIERTC Register to mask the NMI signal. This register is shared with the real-time clock. The real-time-clock uses the lower six bits of this port to address memory locations. Writing to port 70h sets both the enable/disable bit and the memory address pointer. Do not modify the contents of this register without considering the effects on the state of the other bits.

Bit	Description
7	**NMI Enable:** Setting bit 7 to a 1 will disable all NMI sources. Setting the bit to a 0 enables the NMI interrupt.
6:0	**Real-Time Clock Address:** Used by the Real-Time Clock on the Base I/O component to address memory locations. Not used for NMI enabling/disabling.

3.4.12 NMIESC—NMI EXTENDED STATUS AND CONTROL REGISTER

Register Location: 0461h
Default Value: See below
Attribute: Read/Write, Read Only
Size: 8 Bits

This register is used to check the status of different system components, control the output of the Speaker Counter (Timer 1, Counter 2), and gate the counter output that drives the SPKR signal.

Bits 4, 5, 6, and 7 are read-only. Bits 0-3 are both read and write. When writing to this port, these bits must be written as 0's. Bit 7 returns the Fail-Safe Timer Status. This input comes from Timer 2, Counter 0. The current status of bit 2 enables or disables this Fail-Safe Timer NMI source. Bit 6 returns the Bus Timeout Status. Bit 6 is set if either a 64 BCLK or a 256 BCLK occurs. The current status of bit 3 enables or disables this Fail-Safe Timer NMI source. If NMI is caused by a Bus Timeout, bit 4 distinguished between the 8 μs (64 BCLK) and

32 μs (256 BCLK) timeout. Bit 5 is the current state of an I/O write to port 0462h. The current status of bit 1 enables or disables Software generated NMI. Bit 0 controls the state of the RSTDRV output signal. If bit 0 is set to 1, the RSTDRV signal is asserted and a system bus reset is performed. Bit 0 should be set long enough (> 8 BCLKs) for the system bus devices to be properly reset.

Bit	Description
7	**Fail-Safe Timer Status—RO:** This bit indicates the status of the Fail-safe Timer. When Timer 2, Counter 0 count expires, this bit is set to a 1 if bit 2 has previously been set to 1. A value of 0 indicates that the current NMI was not caused by the Fail-Safe Timer. A value of 1 indicates that the Fail-Safe timer has timed out.
6	**Bus Timeout Status—RO:** This bit indicates the status of Bus master timeout logic. If this bit is 0, the Bus Master timeout logic has not detected a bus timeout. If this bit is 1, the bus master timeout logic has detected a bus timeout.
5	**Software NMI Status—RO:** This bit indicates the status of the Software NMI port writes. A write to I/O port 0462 of any value will set this bit to 1 if bit 1 is set to 1. If this bit is 0, the current NMI was not caused by a write to the NMI Port. If this bit is 1, the current NMI was caused by a write to the NMI Port.
4	**Bus Timeout Status—RO:** This bit indicates the status of the 8 μs EISA Bus master timeout event. If the bit is 0, the current NMI was not caused by the 8 μs EISA bus master timeout. If this bit is 1, the current NMI was caused by this bus timeout.
3	**Bus Timeout Enable—R/W:** This bit enables/disables NMI EISA bus timeout. If this bit is 0, an NMI will not be generated for bus timeout. Also the NMI condition caused by the Bus timeout will be cleared. If this bit is 1 an NMI will be generated when Timer 2 Counter 0 count expires.
2	**Fail-Safe NMI Enable—R/W:** This bit enables/disables NMI when the Fail-Safe Timer timesout. If this bit is 0, an NMI will not be generated when the Timer 2 Counter 0 count expires. Also the NMI condition caused by the Fail-Safe Timer will be cleared. If this bit is 1 an NMI will be generated when Timer 2 Counter 0 count expires.
1	**Software NMI Enable—R/W:** This bit enables/disables software generated NMI. If this bit is 0, a write to I/O port 0462h will not generate an NMI. If this bit is 1 NMI will be generated for a write to I/O port 0462h.
0	**Bus Reset—R/W:** When bit 0 is a 0, RSTDRV signal function as a normal reset drive signal. When bit 0 is 1, the RSTDRV signal is asserted. Following reset, bit 0 is a 0 and the RSTDRV output is low.

3.4.13 SOFTNMI—SOFTWARE NMI GENERATION REGISTER

Register Location: 0462h
Default Value: xxh
Attribute: Write Only
Size: 8 Bits

A write to this port with any data will cause an NMI. This port provides a software mechanism to cause an NMI if interrupts are enabled.

Bit	Description
7:0	**Software NMI Port:** The bit pattern is not specific. A write to this port will generate a Software NMI if enabled.

3.5 EISA Configuration, Floppy Support, and Port 92h

3.5.1 CONFRAMP—CONFIGURATION RAM PAGE REGISTER

Register Location: 0C00h
Default Value: xxx00000b
Attribute: Read/Write
Size: 8 Bits

This register contains the Configuration RAM Page address. During accesses to the Configuration RAM (0800h–08FFh), the ESC drivers the CPG[4:0] signals with the value of bits[4:0] of this register. The CPG[4:0] signals are connected to address pins ADDR[12:8] of the Configuration RAM.

Bit	Description
7:5	**Reserved**
4:0	**CRAM Page Address:** The value of these bit selects a specific page from the Configuration RAM space. The SA[7:0] addresses select the location within this page during I/O accesses to the Configuration RAM. The value is driven onto CPG[4:0] during accesses to Configuration RAM.

3.5.2 DIGOUT—DIGITAL OUTPUT REGISTER

Register Location: 03F2h (Primary), 0372h (Secondary)
Default Value: xxxx0xxxb
Attribute: Write only
Size: 8 Bits

This register is used to prevent XBUSOE# from responding to DACK2# during a DMA read access to a floppy controller on the ISA bus. If a second floppy (residing on the ISA bus) is using DACK2# in conjunction with a floppy on the X-bus, this prevents the floppy on the X-Bus and the X-bus transceiver from responding to an access targeted for the floppy on the ISA bus. This register is also located in the floppy controller device.

Bit	Description
7:4	**Not Used:** These bits exist in the 82077 FDC. Refer to the 82077 data sheet for further details.
3	**DMA Enable:** When this bit is a 1, the assertion of DACK# will result in XBUSOE# being asserted. If this bit is 0, DACK2# has no effect on XBUSOE#. This port bit also exists on the 82077 FDC. This bit defaults to disable (0).
2:0	**Not Used:** These bits exist in the 82077 FDC. Refer to the 82077 data sheet for further details.

3.5.3 PORT 92 REGISTER

Register Location: 92h
Default Value: 00100100b
Attribute: Read/Write
Size: 8 Bits

This register is used to support the alternate reset (ALTRST#), alternate A20 (ALTA20), power-on password protection, and fixed disk light function (DLIGHT#). This register is only accessible if bit 6 in the Peripheral Chip Select Enable B Register is set to 1.

Bit	Description
7:6	**Fixed Disk Activity Light:** These bits are used to turn the Fixed Disk Activity Light on and off. When either of these bits are set to a 1, the light is turned on (DLIGHT# driven active). To turn the light off, both of these bits must be 0.
5	**Reserved:** This bit is reserved and will always return a 1 when read.
4	**Not Used:** This bit is not used and will always return a 0 when read.
3	**Power on Password Protection:** A 1 on this bit enables power-on password protection by inhibiting accesses to the RTC memory for RTC addresses (port 70h) from 36h to 3Fh. This is accomplished by not generating RTCRD# and RTCWR# signals for theses accesses.
2	**Reserved:** This bit is reserved and will always return a 1 when read.
1	**ALTA20 Signal:** Writing a 0 to this bit causes the ALTA20 signal to be driven low. Writing a 1 to this bit causes the ALTA20 signal to be driven high.
0	**ALTRST# Signal:** This read/write bit provides an alternate system reset function. This function provides an alternate means to reset the system CPU to effect a mode switch from Protected Virtual Address Mode to the Real Address Mode. This provides a faster means of reset than is provided by the Keyboard controller. This bit is set to a 0 by a system reset. Writing a 1 to this bit will cause the ALTRST# signal to pulse active (low) for approximately 4 SYSCLK's. Before another ALTRST# pulse can be generated, this bit must be written back to a 0.

3.5.4 LEISAMG—LAST EISA BUS MASTER GRANTED REGISTER

Register Location: 0464h
Default Value: xxh
Attribute: Read Only
Size: 8 Bits

This register contains information about which EISA bus master most recently had control of the EISA bus. A bit read of 0 indicates that the corresponding slot most recently was granted the bus.

Bit	Description
7:0	**Last EISA Bus Master:** A value of 1 is placed in the bit position of the most recently granted EISA Bus Master.

3.6 Power Management Registers

This section describes two power management registers that are in the 82374SBAPMS and APMC Registers. These registers are located in normal I/O space and must be accessed (via the CPU or PCI Bus) with 8-bit accesses. Note that the rest of the power management registers are part of the ESC configuration registers.

3.6.1 APMC—ADVANCED POWER MANAGEMENT CONTROL PORT

I/O Address: 0B2h
Default Value: 00h
Attribute: Read/Write
Size: 8 Bits

This register passes data (APM Commands) between the OS and the SMI handler. In addition, writes can generate an SMI and reads can cause STPCLK# to be asserted. The ESC operation is not effected by the data in this register.

Bit	Description
7:0	**APM Control Port (APMC):** Writes to this register store data in the APMC Register and reads return the last data written. In addition, writes generate an SMI, if bit 7 of the SMIEN Register and bit 0 of the SMICNTL Register are both is set to 1. Reads cause the STPCLK# signal to be asserted, if bit 1 of the SMICNTL Register is set to 1. Reads do not generate an SMI.

3.6.2 APMS—ADVANCED POWER MANAGEMENT STATUS PORT

I/O Address: 0B3h
Default Value: 00h
Attribute: Read/Write
Size: 8 Bits

This register passes status information between the OS and the SMI handler. The ESC operation is not effected by the data in this register.

Bit	Description
7:0	**APM Status Port (APMS):** Writes store data in this register and reads return the last data written.

3.7 APIC Registers

This section describes the registers used to program the Advanced Programmable Interrupt Controller. The I/O APIC registers are accessed by an indirect addressing scheme using two registers (IOREGSEL and IOWIN) that are located in the CPU's memory space (memory address specified by the APICBASE Register). To reference an I/O APIC register, a Dword memory write loads the IOREGSEL Register with a 32-bit value that specifies the APIC register. The IOWIN Register then becomes a four byte window pointing to the APIC register specified by bits[7:0] of the IOREGSEL Register. The register address table is shown in the Address Decode section.

All APIC registers are accessed using 32-bit loads and stores. This implies that to modify a field (e.g., bit, byte) in any register, the whole 32-bit register must be read, the field modified, and the 32-bits written back. In addition, registers that are described as 64-bits wide are accessed as multiple independent 32-bit registers.

3.7.1 IOREGSEL—I/O REGISTER SELECT REGISTER

Memory Address: FEC0 x000h (82374EB) (x=See APICBASE Register)

FEC0 xy00h (82374SB) (xy=See APICBASE Register)

Default Value: 00h
Attribute: Read/Write
Size: 32 Bits

This register selects an I/O APIC Unit register. The contents of the selected 32-bit register can be manipulated via the I/O Window Register.

Bit	Description
31:8	**Reserved**
7:0	**APIC Register Address:** Bits[7:0] specify the APIC register to be read/written via the IOWIN Register.

3.7.2 IOWIN—I/O WINDOW REGISTER

Memory Address: FEC0 x010h (82374EB) (x=See APICBASE Register)

FEC0 xy10h (82374SB) (xy=See APICBASE Register)

Default Value: 00h
Attribute: Read/Write
Size: 32 Bits

This register is mapped onto the I/O Unit's register selected by the IOREGSEL Register. Readability/writability by software is determined by the I/O APIC register that is currently selected.

Bit	Description
31:0	**APIC Register Data:** Memory references to this register are mapped to the APIC register specified by the contents of the IOREGSEL Register.

3.7.3 APICID—I/O APIC IDENTIFICATION REGISTER

Address Offset: 00h
Default Value: 00h
Attribute: Read/Write
Size: 32 Bits

This register contains the unit's 4-bit APIC ID. The ID serves as a physical name of the I/O APIC Unit. All APIC units using the APIC bus should have a unique APIC ID. The APIC bus arbitration ID for the I/O unit is also writtten during a write to the APICID Register (same data is loaded into both). This register must be programmed with the correct ID value before using the I/O APIC unit for message transmission.

Bit	Description
31:28	**Reserved**
27:24	**I/O APIC Identification:** This 4-bit field contains the I/O APIC identification.
23:0	**Reserved**

3.7.4 APICID—I/O APIC IDENTIFICATION REGISTER

Address Offset: 01h
Default Value: 00h
Attribute: Read Only
Size: 32 Bits

The I/O APIC Version Register identifies the APIC hardware version. Software can use this to provide compatibility between different APIC implementations and their versions. In addition, this register provides the maximum number of entries in the I/O Redirection Table.

Bit	Description
31:24	**Reserved**
23:16	**Maximum Redirection Entry:** This field contains the entry number (0 being the lowest entry) of the highest entry in the I/O Redirection Table. The value is equal to the number of interrupt input pins minus one of this I/O APIC. The range of values is 0 through 239. For the ESC, this value is 0Fh.
15:8	**Reserved**
7:0	**APIC VERSION:** This 8 bit field identifies the implementation version. The version numbers are assigned for 82489DX and APIC as follows: 0Xh = 82489DX 1Xh = APIC 20–FFh = Reserved For the ESC, the current value is 11h.

3.7.5 APICARB—I/O APIC ARBITRATION REGISTER

Address Offset: 02h
Default Value: 000F0011h
Attribute: Read Only
Size: 32 Bits

The APICARB Register contains the bus arbitration priority for the I/O APIC. This register is loaded whenever the I/O APIC ID Register is written.

Bit	Description
31:28	**Reserved**
27:24	**I/O APIC Identification:** This 4 bit field contains the I/O APIC identification.
23:0	**Reserved**

3.7.6 IOREDTBL[15:0]—I/O REDIRECTION TABLE REGISTERS

Address Offset: 10-11h (IOREDTBL0)
12-13h (IOREDTBL1)
14-15h (IOREDTBL2)
16-17h (IOREDTBL3)
18-19h (IOREDTBL4)
1A-1Bh (IOREDTBL5)
1C-1Dh (IOREDTBL6)
1E-1Fh (IOREDTBL7)
20-21h (IOREDTBL8)
22-23h (IOREDTBL9)
24-25h (IOREDTBL10)
26-27h (IOREDTBL11)
28-29h (IOREDTBL12)
2A-2Bh (IOREDTBL13)
2C-2Dh (IOREDTBL14)
2E-2Fh (IOREDTBL15)
Default Value: xx000000 00010xxxh
Attribute: Read/Write, Read Only
Size: 64 Bits each

There are 16 I/O Redirection Table entry registers. Each register is a dedicated entry for each interrupt input pin. Unlike IRQ pins of the 8259A, the notion of interrupt priority is completely unrelated to the position of the physical interrupt input pin on the APIC. Instead, software determines the vector (and therefore the priority) for each corresponding interrupt input pin. For each interrupt pin, the operating system can also specify the signal polarity (low active or high active), whether the interrupt is signaled as edges or levels, as well as the destination and delivery mode of the interrupt. The information in the redirection table is used to translate the corresponding interrupt pin information into an inter-APIC message.

For a signal on an edge-sensitive interrupt input pin to be recognized as a valid edge (and not a glitch), the input level on the pin must remain asserted until the I/O APIC Unit broadcasts the corresponding message over the APIC bus and the message has been accepted by the destination(s) specified in the destination field. Only then will the source APIC be able to recognize a new edge on that Interrupt Input pin. That new edge only results in a new invocation of the handler if its acceptance by the destination APIC causes the Interrupt Request Register bit to go from 0 to 1. (In other words, if the interrupt wasn't already pending at the destination.)

Bit	Description
63:56	**Destination Field:** If the Destination Mode of this entry is Physical Mode (bit 11 = 0), bits[59:56] contain an APIC ID. If Logical Mode is selected (bit 11 = 1), the Destination Field potentially defines a set of processors. Bits[63:56] of the Destination Field specify the logical destination address.
55:17	**Reserved**
16	**Interrupt Mask:** When this bit is 1, the interrupt signal is masked. Edge-sensitive interrupts signaled on a masked interrupt pin are ignored (i.e., not delivered or held pending). Level-asserts or negates occurring on a masked level-sensitive pin are also ignored and have no side effects. Changing the mask bit from unmasked to masked after the interrupt is accepted by a local APIC has no effect on that interrupt. This behavior is identical to the case where the device withdraws the interrupt before that interrupt is posted to the processor. It is software's responsibility to handle the case where the mask bit is set after the interrupt message has been accepted by a local APIC unit but before the interrupt is dispensed to the processor. When this bit is 0, the interrupt is not masked. An edge or level on an interrupt pin that is not masked results in the delivery of the interrupt to the destination.
15	**Trigger Mode—R/W:** The trigger mode field indicates the type of signal on the interrupt pin that triggers an interrupt. This bit is set to 1 for level sensitive and 0 for edge sensitive.
14	**Remote IRR—RO:** This bit is used for level triggered interrupts. Its meaning is undefined for edge triggered interrupts. For level triggered interrupts, this bit is set to 1 when local APIC(s) accept the level interrupt sent by the I/O APIC. The Remote IRR bit is set to 0 when an EOI message with a matching interrupt vector is received from a local APIC.
13	**Interrupt Input Pin Polarity (INTPOL)—R/W:** This bit specifies the polarity of the interrupt signal. A 0 selects high active and a 1 selects low active.
12	**Delivery Status (DELIVS)—RO:** The Delivery Status bit contains the current status of the delivery of this interrupt. Delivery Status is read-only and writes to this bit (as part of a 32 bit word) do not effect this bit. When bit 12 = 0 (IDLE), there is currently no activity for this interrupt. When bit 12 = 1 (Send Pending), the interrupt has been injected. However, its delivery is temporarily held up due to the APIC bus being busy or the inability of the receiving APIC unit to accept that interrupt at that time.
11	**Destination Mode (DESTMOD)—R/W:** This field determines the interpretation of the Destination field. When DESTMOD = 0 (physical mode), a destination APIC is identified by its ID. Bits 56 through 59 of the Destination field specify the 4 bit APIC ID. When DESTMOD = 1 (logical mode), destinations are identified by matching on the logical destination under the control of the Destination Format Register and Logical Destination Register in each Local APIC. Bits 56 through 63 (8 MSB) of the Destination field specify the 8 bit APIC ID.

Bit	Description
10:8	**Delivery Mode (DELMOD)—R/W:** The Delivery Mode is a 3 bit field that specifies how the APICs listed in the destination field should act upon reception of this signal. Note that certain Delivery Modes only operate as intended when used in conjunction with a specific trigger Mode. These restrictions are indicated in the following table for each Delivery Mode.

Bits [10:8]	Mode	Description
000	Fixed	Deliver the signal on the INTR signal of all processor cores listed in the destination. Trigger Mode for "fixed" Delivery Mode can be edge or level.
001	Lowest Priority	Deliver the signal on the INTR signal of the processor core that is executing at the lowest priority among all the processors listed in the specified destination. Trigger Mode for "lowest priority". Delivery Mode can be edge or level.
010	SMI	System Management Interrupt. A delivery mode equal to SMI requires an edge trigger mode. The vector information is ignored but must be programmed to all zeroes for future compatibility.
011	Reserved	
100	NMI	Deliver the signal on the NMI signal of all processor cores listed in the destination. Vector information is ignored. NMI must be programmed as edge-triggered for proper operation.
101	INIT	Deliver the signal to all processor cores listed in the destination by asserting the INIT signal. All addressed local APICs will assume their INIT state. INIT must be programmed as edge-triggered for proper operation
110	Reserved	
111	ExtINT	Deliver the signal to the INTR signal of all processor cores listed in the destination as an interrupt that originated in an externally connected (8259A-compatible) interrupt controller. The INTA cycle that corresponds to this ExtINT delivery is routed to the external controller that is expected to supply the vector. A Delivery Mode of "ExtINT" requires an edge trigger mode.

7:0	**Interrupt Vector (INTVEC)—R/W:** The vector field is an 8 bit field containing the interrupt vector for this interrupt. Vector values range from 10 to FEh.

4.0 ADDRESS DECODING

The ESC contains an address decoder to decode EISA/ISA master cycles. The ESC address decoder uses the address line LA[31:2], and byte enable BE[3:0]# to decode EISA master cycles. For ISA master cycles, the ESC uses address line LA[31:2], SA[1:0], and high byte enable SHBE# for address decode.

The ESC decodes the following set of addresses.

1. BIOS memory space.
2. I/O addresses contained within the ESC.
3. Configuration registers.
4. X-Bus Peripherals.
5. Memory addresses for accessing APIC.

4.1 BIOS Memory Space

The ESC supports a total of 512 KBytes of BIOS. The ESC will assert the LBIOSCS# signal for memory cycles decoded to be in the BIOS space. The 512 KBytes of BIOS includes the conventional 128 KBytes of BIOS and 384 KBytes of enlarged BIOS.

The 128 KBytes conventional BIOS memory space is mapped at 1 MByte boundary between memory address 000E0000h–000FFFFFh. The 128 KByte conventional BIOS memory space is split into one 64 KByte region, and four 16 KByte regions. These regions are Low BIOS region 1 (000E0000h–000E3FFFh), Low BIOS region 2 (000E4000h–000E7FFFh), Low BIOS region 3 (000E8000h–000EBFFFh), and Low BIOS region 4 (000EC000h–000EFFFFh) and High BIOS region (000F0000h–000FFFFFh). The ESC will assert the LBIOSCS# signal for memory cycles to these regions if the corresponding configuration bits in the BIOS Chip Select A register are set to enable (see Table 3).

The conventional BIOS is aliased at multiple memory regions. The aliased memory regions are at 16 MByte boundary (High BIOS only), 4 GByte minus 1 MByte boundary, and 4 GByte boundary. The ESC will assert LBIOSCS# for memory cycles to these aliased regions if the corresponding configuration bits in the BIOS Chip Select B register are also set to enable (see Table 3).

The ESC supported VGA BIOS on the motherboard by aliasing the VGA BIOS region to the conventional BIOS region. The VGA BIOS is accessed at memory region 0000C0000h–0000C7FFF. The VGA BIOS region is divided into a Low VGA region (000C0000h–000C3FFFh) and a High VGA region (000C4000h–000C7FFFh). If the BIOS Chip Select B register bit 0 (Low VGA BIOS Enable) and bit 1 (High VGA BIOS Enable) are set to enable, memory accesses to Low VGA BIOS region and High VGA BIOS region will be aliased to conventional Low BIOS region 1 and Low BIOS region 2 respectively and the ESC will assert LBIOSCS#

The ESC supports the 384 KBytes of enlarged BIOS as specified by the PCI specification. This 384 KByte region is mapped in memory space below the 4 GByte aliased conventional BIOS. The enlarged BIOS is accessed between FFF80000h–FFFDFFFFh memory space. If the enlarged BIOS is enabled in the BIOS Enable Chip Select 1 register bit 5 (Enlarged BIOS Enable), the ESC will assert LBIOSCS# signal for accesses to this region.

BIOS Location Auto-Detection

Some applications require that Flash-EPROM based BIOS be updated in the system from the data coming from the floppy disk. To support this, the X-bus signals must be properly controlled (i.e. the ESC's X-Bus control logic must be aware of physical BIOS location—X-bus or ISA Bus). This is supported transparently to the software by configuring ESC's X-Bus logic during RESET using the SLOWH# pin.

Logic level on SLOWH# pin is sampled at the end of reset sequence to determine whether BIOS resides on the X-Bus (1) or on the ISA bus (0). This information is used by the ESC to control the X-Bus transceivers during BIOS access.

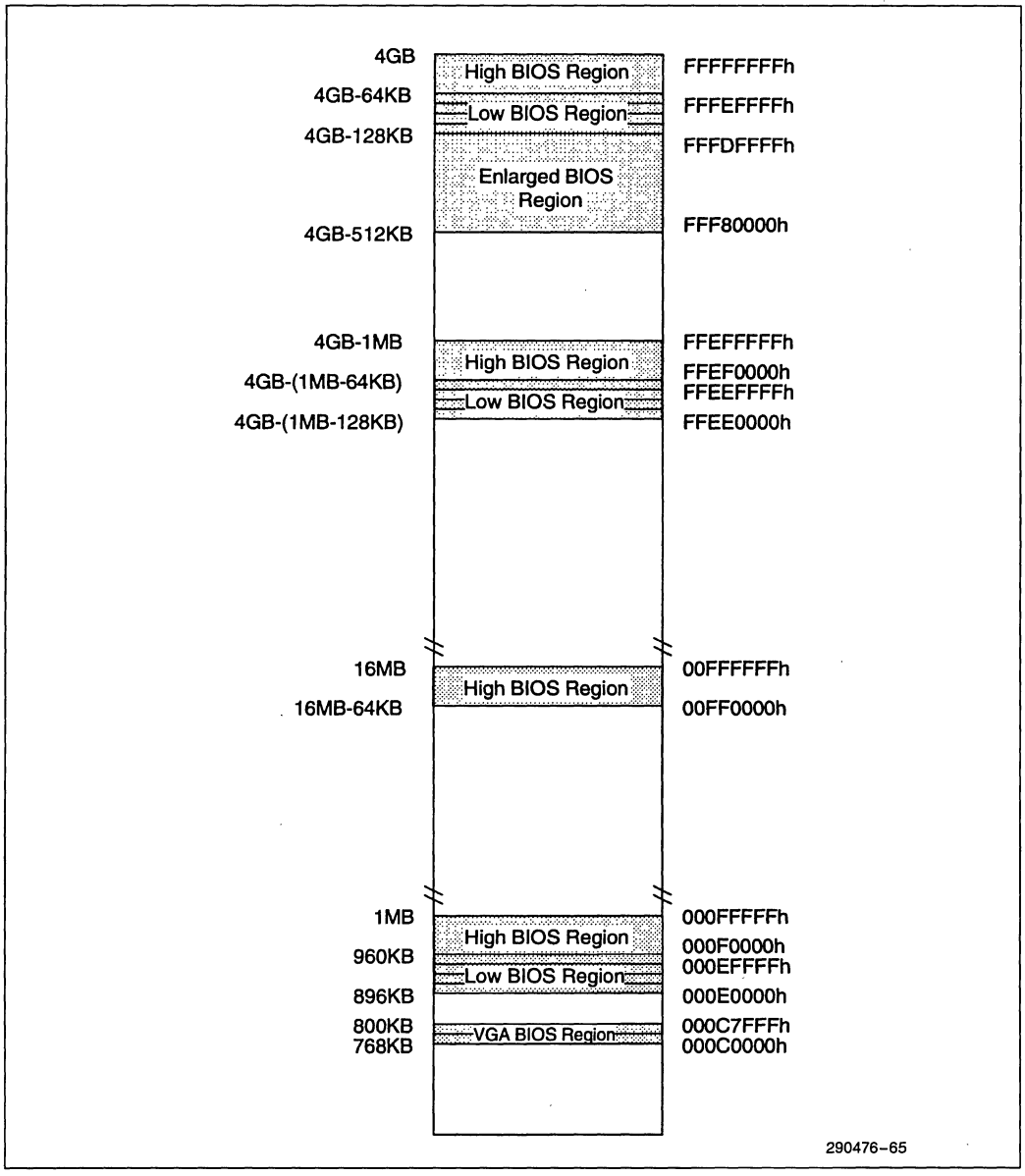


Figure 2. BIOS Memory Map

Table 3. BIOS Chip Select Enable Table

Memory Address Range	Low BIOS En				High BIOS En	ENL BIOS En	Low VGA BIOS En	High VGA BIOS En	16M BIOS En	LBIOSCS# Asserted
	1	2	3	4						
000C0000h to	x	x	x	x	x	x	**0**	x	x	No
000C3FFFh	x	x	x	x	x	x	**1**	x	x	Yes
000C4000h to	x	x	x	x	x	x	x	**0**	x	No
000C7FFFh	x	x	x	x	x	x	x	**1**	x	Yes
000E0000h to	**0**	x	x	x	x	x	x	x	x	No
000E3FFFh	**1**	x	x	x	x	x	x	x	x	Yes
000E4000h to	x	**0**	x	x	x	x	x	x	x	No
000E7FFFh	x	**1**	x	x	x	x	x	x	x	Yes
000E8000h to	x	x	**0**	x	x	x	x	x	x	No
000EBFFFh	x	x	**1**	x	x	x	x	x	x	Yes
000EC000h to	x	x	x	**0**	x	x	x	x	x	No
000EFFFFh	x	x	x	**1**	x	x	x	x	x	Yes
000F0000h to	x	x	x	x	**0**	x	x	x	x	No
000FFFFFh	x	x	x	x	**1**	x	x	x	x	Yes
(960KB to 1MB)										
00FF0000h to	x	x	x	x	**x**	x	x	x	**0**	No
00FFFFFFh	x	x	x	x	**0**	x	x	x	**0**	Yes
(16MB-64KB to	x	x	x	x	**1**	x	x		**1**	
16MB)										
FFEE0000h to	**0**	x	x	x	x	x	x	x	x	No
FFEE3FFFh	**1**	x	x	x	x	x	x	x	x	Yes
FFEE4000h to	x	**0**	x	x	x	x	x	x	x	No
FFEE7FFFh	x	**1**	x	x	x	x	x	x	x	Yes
FFEE8000h to	x	x	**0**	x	x	x	x	x	x	No
FFEEBFFFh	x	x	**1**	x	x	x	x	x	x	Yes
FFEEC000h to	x	x	x	**0**	x	x	x	x	x	No
FFEEFFFFh	x	x	x	**1**	x	x	x	x	x	Yes
FFEF0000h to	x	x	x	x	**0**	x	x	x	x	No
FFEFFFFFh	x	x	x	x	**1**	x	x	x	x	Yes
FFF80000h to	x	x	x	x	x	**0**	x	x	x	No
FFFDFFFFh	x	x	x	x	x	**1**	x	x	x	Yes
(4GB–512KB to										
4G–128KB										
FFFE0000h to	**0**	x	x	x	x	x	x	x	x	No
FFFE3FFFh	**1**	x	x	x	x	x	x	x	x	Yes
FFFE4000h to	x	**0**	x	x	x	x	x	x	x	No
FFFE7FFFh	x	**1**	x	x	x	x	x	x	x	Yes

Table 3. BIOS Chip Select Enable Table (Continued)

Memory Address Range	Low BIOS En				High BIOS En	ENL BIOS En	Low VGA BIOS En	High VGA BIOS En	16M BIOS En	LBIOSCS# Asserted
	1	2	3	4						
FFFE8000h to	x	x	**0**	x	x	x	x	x	x	No
FFFEBFFFh	x	x	**1**	x	x	x	x	x	x	Yes
FFFEC000h to	x	x	x	**0**	x	x	x	x	x	No
FFFEFFFFh	x	x	x	**1**	x	x	x	x	x	Yes
FFFF0000h to	x	x	x	x	**0**	x	x	x	x	No
FFFFFFFFh	x	x	x	x	**1**	x	x	x	x	Yes

NOTES:
1. "x" in the above table represents a don't care condition.
2. All the region control bits for the BIOS space are in the BIOS Chip Select A register and BIOS Chip Select 2 register at configuration offsets 42h and 43h respectively.

4.2 I/O Addresses Contained Within The ESC

The ESC integrates functions like DMA, Programmable Interrupt Controller, and Timers. All the compatibility registers associated with these functions are also integrated into the ESC. The ESC also integrates some additional registers like EISA System ID register in order to reduce the overall chip count in the system.

All the registers integrated in the ESC are located in the I/O range. These are 8-bit registers and are accessed through the ESC EISA interface. The ESC internal registers are at fixed I/O locations with the exception of DMA Scatter-Gather registers. The DMA Scatter-Gather registers default to the I/O addresses 0410h to 043Fh upon reset. These registers can be relocated by programming the Scatter-Gather Relocate Base Address register. The DMA Scatter-Gather registers can be relocated to I/O addresses range xx10h-xx3Fh.

Registers at I/O addresses 70h, 372h, and 3F2h are shared registers between ESC and external logic. Port 70h is duplicated in the Real Time Clock logic. Bit 3 of ports 372h and 3F2h reside in the ESC and the other bits reside in the Floppy Disk Controller. Table 4 documents the I/O address to the ESC internal registers.

Table 4. ESC I/O Register Address Map

Address (Hex)	Address (Bit)				Type	Name	Block
	FEDC	BA98	7654	3210			
0000H	0000	0000	000x	0000	R/W	DMA1 CH0 Base and Current Address	DMA
0001h	0000	0000	000x	0001	R/W	DMA1 CH0 Base and Current Count	DMA
0002h	0000	0000	000x	0010	R/W	DMA1 CH1 Base and Current Address	DMA
0003h	0000	0000	000x	0011	R/W	DMA1 CH1 Base and Current Count	DMA
0004h	0000	0000	000x	0100	R/W	DMA1 CH2 Base and Current Address	DMA

Table 4. ESC I/O Register Address Map (Continued)

Address (Hex)	Address (Bit)				Type	Name	Block
	FEDC	BA98	7654	3210			
0005H	0000	0000	000x	0101	R/W	DMA1 CH2 Base and Current Count	DMA
0006h	0000	0000	000x	0110	R/W	DMA1 CH3 Base and Current Address	DMA
0007h	0000	0000	000x	0111	R/W	DMA1 CH3 Base and Current Count	DMA
0008h	0000	0000	000x	1000	R/W	DMA1 Status(r) Command(w) Register	DMA
0009h	0000	0000	000x	1001	WO	DMA1 Write Request	DMA
000Ah	0000	0000	000x	1010	WO	DMA1 Write Single Mask Bit	DMA
000Bh	0000	0000	000x	1011	WO	DMA1 Write Mode	DMA
000Ch	0000	0000	000x	1100	WO	DMA1 Clear Byte Pointer	DMA
000Dh	0000	0000	000x	1101	WO	DMA1 Master Clear	DMA
000Eh	0000	0000	000x	1110	WO	DMA1 Clear Mask	DMA
000Fh	0000	0000	000x	1111	R/W	DMA1 Read/Write All Mask Register Bits	DMA
0020h	0000	0000	001x	xx00	R/W	INT 1 Control	PIC
0021h	0000	0000	001x	xx01	R/W	INT 1 Mask	PIC
0022h	0000	0000	0010	0010	R/W	Configuration Address Index	CONF
0023h	0000	0000	0010	0011	R/W	Configuration Data Index	CONF
0040h	0000	0000	010x	0000	R/W	Timer 1 Counter 0—System Clock	TC
0041h	0000	0000	010x	0001	R/W	Timer1 Counter 1—Refresh Request	TC
0042h	0000	0000	010x	0010	R/W	Timer 1 Counter 2—Speaker Tone	TC
0043h	0000	0000	010x	0011	WO	Timer 1 Command Mode	TC
0048h	0000	0000	010x	1000	R/W	Timer 2 Counter 0—Fail-Safe Timer	TC
0049h	0000	0000	010x	1001	R/W	Timer 2 Counter 1—Reserved	TC
004Ah	0000	0000	010x	1010	R/W	Timer 2 Counter 2—CPU Speed Control	TC
004Bh	0000	0000	010x	1011	WO	Timer 2 Command Mode Register	TC
0061h	0000	0000	0110	00x1	R/W	NMI Status and Control	Control

Table 4. ESC I/O Register Address Map (Continued)

Address (Hex)	Address (Bit)				Type	Name	Block
	FEDC	BA98	7654	3210			
0070H[1]	0000	0000	0111	0xx0	WO	NMI Mask	Control
0080h	0000	0000	100x	0000	R/W	DMA Page Register—Reserved	DMA
0081h	0000	0000	100x	0001	R/W	DMA Channel 2 Page	DMA
0082h	0000	0000	1000	0010	R/W	DMA Channel 3 Page	DMA
0083h	0000	0000	100x	0011	R/W	DMA Channel 1 Page	DMA
0084h	0000	0000	100x	0100	R/W	DMA Page Register—Reserved	DMA
0085h	0000	0000	100x	0101	R/W	DMA Page Register—Reserved	DMA
0086h	0000	0000	100x	0110	R/W	DMA Page Register—Reserved	DMA
0087h	0000	0000	100x	0111	R/W	DMA Channel 0 Page	DMA
0088h	0000	0000	100x	1000	R/W	DMA Page Register—Reserved	DMA
0089h	0000	0000	100x	1001	R/W	DMA Channel 6 Page	DMA
008Ah	0000	0000	100x	1010	R/W	DMA Channel 7 Page	DMA
008Bh	0000	0000	100x	1011	R/W	DMA Channel 5 Page	DMA
008Ch	0000	0000	100x	1100	R/W	DMA Page Register—Reserved	DMA
008Dh	0000	0000	100x	1101	R/W	DMA Page Register—Reserved	DMA
008Eh	0000	0000	100x	1110	R/W	DMA Page Register—Reserved	DMA
008Fh	0000	0000	100x	1111	R/W	DMA Refresh Page	DMA
0092h	0000	0000	1001	0010	R/W	System Control Port	Control
00A0h	0000	0000	101x	xx00	R/W	INT 2 Control	PIC
00A1h	0000	0000	101x	xx01	R/W	INT 2 Mask	PIC
00B2h	0000	0000	1011	0010	R/W	Advanced Power Management Control	APM
00B3h	0000	0000	1011	0011	R/W	Advanced Power Management Status	APM
00C0h	0000	0000	1100	000x	R/W	DMA2 CH0 Base and Current Address	DMA
00C2h	0000	0000	1100	001x	R/W	DMA2 CH0 Base and Current Count	DMA
00C4h	0000	0000	1100	010x	R/W	DMA2 CH1 Base and Current Address	DMA
00C6h	0000	0000	1100	011x	R/W	DMA2 CH1 Base and Current Count	DMA

Table 4. ESC I/O Register Address Map (Continued)

Address (Hex)	Address (Bit)				Type	Name	Block
	FEDC	BA98	7654	3210			
00C8H	0000	0000	1100	100x	R/W	DMA2 CH2 Base and Current Address	DMA
00CAh	0000	0000	1100	101x	R/W	DMA2 CH2 Base and Current Count	DMA
00CCh	0000	0000	1100	110x	R/W	DMA2 CH3 Base and Current Address	DMA
00CEh	0000	0000	1100	111x	R/W	DMA2 CH3 Base and Current Count	DMA
00D0h	0000	0000	1101	000x	R/W	DMA2 Status(r) Command(w) Register	DMA
00D2h	0000	0000	1101	001x	WO	DMA2 Write Request	DMA
00D4h	0000	0000	1101	010x	WO	DMA2 Write Single Mask Bit	DMA
00D6h	0000	0000	1101	011x	WO	DMA2 Write Mode	DMA
00D8h	0000	0000	1101	100x	WO	DMA2 Clear Byte Pointer	DMA
00DAh	0000	0000	1101	101x	WO	DMA2 Master Clear	DMA
00DCh	0000	0000	1101	110x	WO	DMA2 Clear Mask	DMA
00DEh	0000	0000	1101	111x	R/W	DMA2 Read/Write All Mask Register Bits	DMA
00F0h	0000	0000	1111	0000	WO	Reset IRQ13	IRQ13
0372h[2]	0000	0011	0111	0010	WO	Secondary Floppy Disk Digital Output	FDCCS#
03F2h[2]	0000	0011	1111	0001	WO	Primary Floppy Disk Digital Output	FDCCS#
0400h	0000	0100	0000	0000	R/W	Reserved	DMA
0401h	0000	0100	0000	0001	R/W	DMA1 CH0 Base/Current Count	DMA
0402h	0000	0100	0000	0010	R/W	Reserved	DMA
0403h	0000	0100	0000	0011	R/W	DMA1 CH1 Base/Current Count	DMA
0404h	0000	0100	0000	0100	R/W	Reserved	DMA
0405h	0000	0100	0000	0101	R/W	DMA1 CH2 Base/Current Count	DMA
0406h	0000	0100	0000	0110	R/W	Reserved	DMA
0407h	0000	0100	0000	0111	R/W	DMA1 CH3 Base/Current Count	DMA
0408h	0000	0100	0000	1000	R/W	Reserved	DMA

Table 4. ESC I/O Register Address Map (Continued)

Address (Hex)	Address (Bit)				Type	Name	Block
	FEDC	BA98	7654	3210			
0409H	0000	0100	0000	1001	R/W	Reserved	DMA
040Ah	0000	0100	0000	1010	R/W	DMA Chaining Mode Status/Interrupt Pending	DMA
040Bh	0000	0100	0000	1011	WO	DMA1 Extended Mode	DMA
040Ch	0000	0100	0000	1100	WO	Chaining Buffer Control	DMA
040Dh	0000	0100	0000	1101	R/W	Reserved	DMA
040Eh	0000	0100	0000	1110	R/W	Reserved	DMA
040Fh	0000	0100	0000	1111	R/W	Reserved	DMA
0410h	0000	0100	0010	0000	WO	DMA CH0 S-G Command	DMA
0411h	0000	0100	0010	0001	WO	DMA CH1 S-G Command	DMA
0412h	0000	0100	0010	0010	WO	DMA CH2 S-G Command	DMA
0413h	0000	0100	0010	0011	WO	DMA CH3 S-G Command	DMA
0415h	0000	0100	0010	0101	WO	DMA CH5 S-G Command	DMA
0416h	0000	0100	0010	0110	WO	DMA CH6 S-G Command	DMA
0417h	0000	0100	0010	0111	WO	DMA CH7 S-G Command	DMA
0418h	0000	0100	0010	1000	WO	DMA CH0 S-G Status	DMA
0419h	0000	0100	0010	1001	WO	DMA CH1 S-G Status	DMA
041Ah	0000	0100	0010	1010	WO	DMA CH2 S-G Status	DMA
041Bh	0000	0100	0010	1011	WO	DMA CH3 S-G Status	DMA
041Dh	0000	0100	0010	1101	WO	DMA CH5 S-G Status	DMA
041Eh	0000	0100	0010	1110	WO	DMA CH6 S-G Status	DMA
041Fh	0000	0100	0010	1111	WO	DMA CH7 S-G Status	DMA
0420h	0000	0100	0010	0000	RO	DMA CH0 S-G Descriptor Pointer	DMA
0421h	0000	0100	0010	0001	RO	DMA CH0 S-G Descriptor Pointer	DMA
0422h	0000	0100	0010	0010	RO	DMA CH0 S-G Descriptor Pointer	DMA
0423h	0000	0100	0010	0011	RO	DMA CH0 S-G Descriptor Pointer	DMA
0424h	0000	0100	0010	0100	RO	DMA CH1 S-G Descriptor Pointer	DMA

Table 4. ESC I/O Register Address Map (Continued)

Address (Hex)	Address (Bit)				Type	Name	Block
	FEDC	BA98	7654	3210			
0425H	0000	0100	0010	0101	RO	DMA CH1 S-G Descriptor Pointer	DMA
0426h	0000	0100	0010	0110	RO	DMA CH1 S-G Descriptor Pointer	DMA
0427h	0000	0100	0010	0111	RO	DMA CH1 S-G Descriptor Pointer	DMA
0428h	0000	0100	0010	1000	RO	DMA CH2 S-G Descriptor Pointer	DMA
0429h	0000	0100	0010	1001	RO	DMA CH2 S-G Descriptor Pointer	DMA
042Ah	0000	0100	0010	1010	RO	DMA CH2 S-G Descriptor Pointer	DMA
042Bh	0000	0100	0010	1011	RO	DMA CH2 S-G Descriptor Pointer	DMA
042Ch	0000	0100	0010	1100	RO	DMA CH3 S-G Descriptor Pointer	DMA
042Dh	0000	0100	0010	1101	RO	DMA CH3 S-G Descriptor Pointer	DMA
042Eh	0000	0100	0010	1110	RO	DMA CH3 S-G Descriptor Pointer	DMA
042Fh	0000	0100	0010	1111	RO	DMA CH3 S-G Descriptor Pointer	DMA
0434h	0000	0100	0011	0100	RO	DMA CH5 S-G Descriptor Pointer	DMA
0435h	0000	0100	0011	0101	RO	DMA CH5 S-G Descriptor Pointer	DMA
0436h	0000	0100	0011	0110	RO	DMA CH5 S-G Descriptor Pointer	DMA
0437h	0000	0100	0011	0111	RO	DMA CH5 S-G Descriptor Pointer	DMA
0438h	0000	0100	0011	1000	RO	DMA CH6 S-G Descriptor Pointer	DMA
0439h	0000	0100	0011	1001	RO	DMA CH6 S-G Descriptor Pointer	DMA
043Ah	0000	0100	0011	1010	RO	DMA CH6 S-G Descriptor Pointer	DMA
043Bh	0000	0100	0011	1011	RO	DMA CH6 S-G Descriptor Pointer	DMA

Table 4. ESC I/O Register Address Map (Continued)

Address (Hex)	Address (Bit)				Type	Name	Block
	FEDC	BA98	7654	3210			
043CH	0000	0100	0011	1100	RO	DMA CH7 S-G Descriptor Pointer	DMA
043Dh	0000	0100	0011	1101	RO	DMA CH7 S-G Descriptor Pointer	DMA
043Eh	0000	0100	0011	1110	RO	DMA CH7 S-G Descriptor Pointer	DMA
043Fh	0000	0100	0011	1111	RO	DMA CH7 S-G Descriptor Pointer	DMA
0461h	0000	0100	0110	0001	R/W	Extended NMI and Reset Control	Control
0462h	0000	0100	0110	0010	R/W	NMI I/O Interrupt Port	Control
0464h	0000	0100	0110	0100	RO	Last EISA Bus Master Granted (L)	Control
0480h	0000	0100	1000	0000	R/W	Reserved	DMA
0481h	0000	0100	1000	0001	R/W	DMA CH2 High Page	DMA
0482h	0000	0100	1000	0010	R/W	DMA CH3 High Page	DMA
0483h	0000	0100	1000	0011	R/W	DMA CH1 High Page	DMA
0484h	0000	0100	1000	0100	R/W	Reserved	DMA
0485h	0000	0100	1000	0101	R/W	Reserved	DMA
0486h	0000	0100	1000	0110	R/W	Reserved	DMA
0487h	0000	0100	1000	0111	R/W	DMA CH0 High Page	DMA
0488h	0000	0100	1000	1000	R/W	Reserved	DMA
0489h	0000	0100	1000	1001	R/W	DMA CH6 High Page	DMA
048Ah	0000	0100	1000	1010	R/W	DMA CH7 High Page	DMA
048Bh	0000	0100	1000	1011	R/W	DMA CH5 High Page	DMA
048Ch	0000	0100	1000	1110	R/W	Reserved	DMA
048Dh	0000	0100	1000	1101	R/W	Reserved	DMA
048Eh	0000	0100	1000	1110	R/W	Reserved	DMA
048Fh	0000	0100	100x	1111	R/W	DMA Refresh High Page	DMA
04C2h	0000	0100	1100	0010	R/W	Reserved	DMA

Table 4. ESC I/O Register Address Map (Continued)

Address (Hex)	Address (Bit)				Type	Name	Block
	FEDC	BA98	7654	3210			
04C6H	0000	0100	1100	0110	R/W	DMA CH5 High Base & Current Count	DMA
04CAh	0000	0100	1100	1010	R/W	DMA CH6 High Base & Current Count	DMA
04CEh	0000	0100	1100	1110	R/W	DMA CH7 High Base & Current Count	DMA
04D0h	0000	0100	1101	0000	R/W	INT-1 Edge/Level Control	PIC
04D1h	0000	0100	1101	0001	R/W	INT-2 Edge/Level Control	PIC
04D2h	0000	0100	1101	0010	R/W	Reserved	DMA
04D3h	0000	0100	1101	0011	R/W	Reserved	DMA
04D4h	0000	0100	1101	0100	R/W	DMA2 Chaining Mode	DMA
04D5h	0000	0100	1101	1001	R/W	Reserved	DMA
04D6h	0000	0100	1101	0010	WO	DMA2 Extended Mode	DMA
04D7h	0000	0100	1101	0111	R/W	Reserved	DMA
04D8h	0000	0100	1101	1000	R/W	Reserved	DMA
04D9h	0000	0100	1101	1001	R/W	Reserved	DMA
04DAh	0000	0100	1101	1010	R/W	Reserved	DMA
04DBh	0000	0100	1101	1011	R/W	Reserved	DMA
04DCh	0000	0100	1101	1100	R/W	Reserved	DMA
04DDh	0000	0100	1101	1101	R/W	Reserved	DMA
04DEh	0000	0100	1101	1110	R/W	Reserved	DMA
04DFh	0000	0100	1101	1111	R/W	Reserved	DMA
04E0h	0000	0100	1110	0000	R/W	DMA CH0 Stop Register Bits[7:2]	DMA
04E1h	0000	0100	1110	0001	R/W	DMA CH0 Stop Register Bits[15:8]	DMA
04E2h	0000	0100	1110	0010	R/W	DMA CH0 Stop Register Bits[23:16]	DMA
04E3h	0000	0100	1110	0011	R/W	Reserved	DNA
04E4h	0000	0100	1110	0100	R/W	DMA CH1 Stop Register Bits[7:2]	DMA
04E5h	0000	0100	1110	0101	R/W	DMA CH1 Stop Register Bits[15:8]	DMA

Table 4. ESC I/O Register Address Map (Continued)

Address (Hex)	Address (Bit)				Type	Name	Block
	FEDC	BA98	7654	3210			
04E6H	0000	0100	1110	0110	R/W	DMA CH1 Stop Register Bits[23:16]	DMA
04E7h	0000	0100	1110	0111	R/W	Reserved	DMA
04E8h	0000	0100	1110	1000	R/W	DMA CH2 Stop Register Bits[7:2]	DMA
04E9h	0000	0100	1110	1001	R/W	DMA CH2 Stop Register Bits[15:8]	DMA
04EAh	0000	0100	1110	1010	R/W	DMA CH2 Stop Register Bits[23:16]	DMA
04EBh	0000	0100	1110	1011	R/W	Reserved	DMA
04EC	0000	0100	1110	1100	R/W	DMA CH3 Stop Register Bits[7:2]	DMA
04EDh	0000	0100	1110	1101	R/W	DMA CH3 Stop Register Bits[15:8]	DMA
04EEh	0000	0100	1110	1110	R/W	DMA CH3 Stop Register Bits[23:16]	DMA
04EFh	0000	0100	1110	1111	R/W	Reserved	DMA
04F0h	0000	0100	1111	0000	R/W	Reserved	DMA
04F1h	0000	0100	1111	0001	R/W	Reserved	DMA
04F2h	0000	0100	1111	0010	R/W	Reserved	DMA
04F3h	0000	0100	1111	0011	R/W	Reserved	DMA
04F4h	0000	0100	1111	0100	R/W	DMA CH5 Stop Register Bits[7:2]	DMA
04F5h	0000	0100	1111	0101	R/W	DMA CH5 Stop Register Bits[15:8]	DMA
04F6h	0000	0100	1111	0110	R/W	DMA CH5 Stop Register Bits[23:16]	DMA
04F7h	0000	0100	1111	0111	R/W	Reserved	DMA
04F8h	0000	0100	1111	1000	R/W	DMA CH6 Stop Register Bits[7:2]	DMA
04F9h	0000	0100	1111	1001	R/W	DMA CH6 Stop Register Bits[15:8]	DMA
04FAh	0000	0100	1111	1010	R/W	DMA CH6 Stop Register Bits[23:16]	DMA

Table 4. ESC I/O Register Address Map (Continued)

Address (Hex)	Address (Bit)				Type	Name	Block
	FEDC	BA98	7654	3210			
04FBH	0000	0100	1111	1011	R/W	Reserved	DMA
04FC	0000	0100	1111	1100	R/W	DMA CH7 Stop Register Bits[7:2]	DMA
04FDh	0000	0100	1111	1101	R/W	DMA CH7 Stop Register Bits[15:8]	DMA
04FEh	0000	0100	1111	0111	R/W	DMA CH7 Stop Register Bits[23:16]	DMA
04FFh	0000	0100	1111	1111	R/W	Reserved	DMA
0C00h	0000	1100	0000	0000	R/W	Configuration RAM Page Register	Conf
0C80h	0000	1100	100	0000	RO	System Board ID Byte Lane 1 Bits[7:0]	Board ID
0C81h	0000	1100	100	0001	RO	System Board ID Byte Lane 2 Bits[15:8]	Board ID
0C82h	0000	1100	100	0010	RO	System Board ID Byte Lane 3 Bits[23:16]	Board ID
0C83h	0000	1100	1000	0011	RO	System Board ID Byte Lane 4 Bits[31:24]	Board ID

NOTES:
1. Port 70h resides in the ESC in addition the lower 7 bits of Port 70h reside in Real Time Clock also.
2. Bit 3 of ports 372h and 3F2h reside in the ESC while the other bits reside on the ISA bus.

4.3 Configuration Addresses

ESC configuration registers are accessed through I/O registers 22h and 23h. These I/O registers are used as index address register (22h) and index data register (23h). The index address register is used to write the configuration register address. The data (configuration register address) in register 22h is used to decode a configuration register. The selected configuration register can be read or written to by performing a read or a write operation to the index data register at I/O address 23h.

Table 5. Configuration Register Index Address

Configuration Offset	Abbreviation	Register Name
00–01h	—	Reserved
02h	ESCID	ESC ID
03–07h	—	Reserved
08h	RID	Revision ID
09–3Fh	—	Reserved

Table 5. Configuration Register Index Address (Continued)

Configuration Offset	Abbreviation	Register Name
40h	MS	Mode Select
41h	—	Reserved
42h	BIOSCSA	BIOS Chip Select A
43h	BIOSCSB	BIOS Chip Select B
44–4Ch	—	Reserved
4Dh	CLKDIV	BCLK Clock Divisor
4Eh	PCSA	Peripheral Chip Select A
4Fh	PCSB	Peripheral Chip Select B
50h	EISAID1	EISA ID Byte 1
51h	EISAID2	EISA ID Byte 2
52h	EISAID3	EISA ID Byte 3
53h	EISAID4	EISA ID Byte 4
54–56h	—	Reserved
57h	SGRBA	Scatter-Gather Relocate Base Address
58h	—	Reserved
59h	APICBA	APIC Base Address Relocation
60h	PIRQRC0	PIRQ0# Route Control
61h	PIRQRC1	PIRQ1# Route Control
62h	PIRQRC2	PIRQ2# Route Control
63h	PIRQRC3	PIRQ3# Route Control
64h	GPCSLA0	General Purpose Chip Select 0 Base Low Address
65h	GPCSHA0	General Purpose Chip Select 0 Base High Address
66h	GPCSM0	General Purpose Chip Select 0 Mask
67h	—	Reserved
68h	GPCSLA1	General Purpose Chip Select 1 Base Low Address
69h	GPCSHA1	General Purpose Chip Select 1 Base High Address
6Ah	GPCSM1	General Purpose Chip Select 1 Mask
6Bh	—	Reserved
6Ch	GPCSLA2	General Purpose Chip Select 2 Base Low Address
6Dh	GPCSHA2	General Purpose Chip Select 2 Base High Address

Table 5. Configuration Register Index Address (Continued)

Configuration Offset	Abbreviation	Register Name
6Eh	GPCSM2	General Purpose Chip Select 2 Mask
6Fh	GPXBC	General Purpose Peripheral X-Bus Control
70h	PACC	PIC/APIC Configuration Control
71–87h	—	Reserved
88h	TSTC	Test Control
89–9Fh	—	Reserved
A0h	SMICNTL	SMI Control
A2-A3h	SMIEN	SMI Enable
A4-A7h	SEE	System Event Enable
A8h	FTMR	Fast Off Timer
A9h	—	Reserved
AA-ABh	SMIREQ	SMI Request
ACh	CTLTMRL	Clock Scaling STPCLK# Low Timer
ADh	—	Reserved
AEh	CTLTMRH	Clock Scaling STPCLK# High Timer
AF-FFh	—	Reserved

4.4 X-Bus Peripherals

The ESC generates chip selects for certain functions that typically reside on the X-Bus. The ESC asserts the chip selects combinatorially from the LA addresses. The ESC generates chip select signals for the Keyboard Controller, Floppy Disk Controller, IDE, Parallel Port, Serial Port, and General Purpose peripherals. The ESC also generates read and write strobes for Real Time Clock and Configuration RAM. The read and write strobes are a function of LA addresses, the ISA read and write strobes (IORC# and IOWC#), and BCLK. All of the peripherals supported by the ESC are at fixed I/O addresses with the exception of the general purpose peripherals. The ESC support for these peripherals can be enabled or disabled through configuration registers Peripheral Chip Select A and Peripheral Chip Select B. The general purpose peripherals are mapped to I/O addresses by programming a set of configuration registers: General Purpose Chip Select x Base Low Address register, General Purpose Chip Select x Base High Address register, and General Purpose Chip Select x Mask register.

Table 6. X-Bus Chip Selects Decode

Address (Hex)	Address (Bit)				R/W	Name	Chip Select
	FEDC	BA98	7654	3210			
0060h	0000	0000	0110	00x0	R/W	Keyboard Controller	KYBDCS#
0064h	0000	0000	0110	01x0	R/W	Keyboard Controller	KYBDCS#
0070h	0000	0000	0111	0xx0	W	Real Time Clock	RTCALE
0071h	0000	0000	0111	0xx1	R/W	Real Time Clock	RTCWR#/ RTCRD#
0170h–0177h	0000	0001	0111	0xxx	R/W	IDE Controller 0-Secondary	ECS[2:0] = 011 (IDECS0#)
01F0h–01F7h	0000	0001	1111	0xxx	R/W	IDE Controller 0-Primary	ECS[2:0] = 011 (IDECS0#)
0278h–027Bh	0000	0010	0111	1000 to 1011	R/W	Parallel Port LPT3	ECS[2:0] = 010 (LPTCS#)
02F8h–02FFh	0000	0010	1111	xxxx	R/W	Serial Port COM2	ECS[2:0] = 00x (COMxCS#)
0370h–0375h	0000	0011	0111	0000 to 0101	R/W	Floppy Disk Controller-Secondary	FDCCS#
0376h	0000	0011	0111	0111	R/W	IDE Controller 1 Secondary	ECS[2:0] = 100 (IDECS1#)
0377h	0000	0011	0111	0110	R/W	IDE Controller 1 Secondary	ECS[2:0] = 100 (IDECS1#)
0377h	0000	0011	0111	0111	R/W	Floppy Disk Controller-Secondary	FDCCS#
0378h–037Bh	0000	0011	0111	1000 to 1011	R/W	Parallel Port LPT2	ECS[2:0] = 010 (LPTCS#)
03BCh–03BFh	0000	0011	1011	11xx	R/W	Parallel Port LPT 1	ECS[2:0] = 010 (LPTCS#)
03F0h–0375h	0000	0011	1111	0000 to 0101	R/W	Floppy Disk Controller-Primary	FDCCS#
03F6h	0000	0011	0111	0110	R/W	IDE Controller 1-Primary	ECS[2:0] = 100 (IDECS1#)

Table 6. X-Bus Chip Selects Decode (Continued)

Address (Hex)	Address (Bit)				R/W	Name	Chip Select
	FEDC	BA98	7654	3210			
03F7h	0000	0011	0111	0111	R/W	IDE Controller 1-Primary	ECS[2:0] = 100 (IDECS1#)
03F7h	0000	0011	0111	0111	R/W	Floppy Disk Controller-Primary	FDCCS#
03F8h–03FFh	0000	0011	1111	1000	R/W	Serial Port COM 1	ECS[2:0] = 00x (COMxCS#)
0800h–08FFh	0000	1000	xxxx	xxxx	W/R	Configuration RAM	CRAMWR#/ CRAMRD#

4.5 I/O APIC Registers

The APIC's registers are indirectly address through two 32 bit registers located in the CPU's memory space—the I/O Register Select (IOREGSEL) and I/O Window (IOWIN) Registers (Table 7). These registers can be relocated via the APIC Base Address Relocation Register and are aligned on 128 bit boundaries.

To access an I/O APIC register, the IOREGSEL Register is written with the address of the intended APIC register. Bits[7:0] of the IOREGSEL Register provide the address offset (Table 8). The IOWIN Register then becomes a 32-bit window pointing to the register selected by the IOREGSEL Register. Note that, for each redirection table register, there are two offset addresses (e.g., address offset 10h selects IOREDTBL0 bits[31:0] and 11h selects IOREDTBL0 bits[63:32].

Table 7. Memory Address For Accessing APIC Registers

Memory Address	Mnemonic	Register Name	Access
FEC0 x000h (82374EB) FEC0 xy00h (82374SB)	IOREGSEL	I/O Register Select	R/W
FEC0 x010h (82374EB) FEC0 xy10h (82374SB)	IOWIN	I/O Window	R/W

NOTE:
xy are determined by the x and y (82374SB only) fields in the APIC Base Address Relocation Register. Range for x = 0-Fh and the range for y = 0,4,8,Ch.

Table 8. I/O APIC Registers

Memory Address	Mnemonic	Register Name	Access
00h	IOAPICID	I/O APIC ID	R/W
01h	IOAPICVER	I/O APIC Version	RO
02h	IOAPICARB	I/O APIC Arbitration ID	RO
10-2Fh	IOREDTBL[0:15]	Redirection Table (Entries 0-15) (63 bits each)	R/W

NOTE:
Address Offset is determined by I/O Register Select Bits[7:0]

5.0 EISA CONTROLLER FUNCTIONAL DESCRIPTION

5.1 Overview

The EISA controller in the ESC provides Master/Slave EISA interface function for the ESC internal resources. In addition, the ESC acts as an EISA central resource for the system. As a system central resource, the EISA controller is responsible for generating the translation control signals necessary for bus-to-bus transfers. These translation includes transfer between devices on EISA Bus and ISA Bus and transfers between different size master device and slave device. The EISA controller generates the control signals for EISA Data Swap Buffers integrated in the PCEB. The ESC EISA interface generates cycles for DMA transfers, and refresh. The ESC internal registers are accessed through the EISA slave interface. The ESC is responsible for supporting the following:

Service EISA Master cycles to:

- EISA slaves devices.
- ISA slave devices.
- ESC internal registers.

Service ISA Master cycles to:

- EISA slave devices.
- ISA (mis-matched) slave devices.
- ESC internal registers.

Service DMA cycles :

- From/to DMA slave on the EISA bus to/from memory on the EISA/ISA bus.
- From/to DMA slave on the ISA bus to/from memory on the EISA/ISA bus.
- From/to DMA slave on the EISA/ISA bus to/from memory on the PCI bus.

Service Refresh Cycles

The EISA controller will service the refresh cycle by generating the appropriate address and command signals. These cycles are initiated by either the ESC internal refresh logic or by an external ISA-Bus Master.

Generates Data Swap Buffer Control

The EISA controller generates the control signals for the data bus swap control (assembly/disassembly) and swapping process to support data size mismatches of the devices on the EISA and ISA buses. The actual data steering and swapping is performed by the PCEB.

Generate Wait States

The wait state generator is responsible for generating the wait states based on the sampling of the EXRDY, CHRDY, NOWS# and the default wait states. The default wait state depends on the cycle type.

5.2 Clock Generation

The ESC is generates the EISA Bus clock. The ESC uses a divider circuit to generate the EISA Bus clock. The ESC supports PCI bus frequencies between 25 MHz and 33 MHz. The PCI clock is divided by 3 or 4 by the clock generation logic in the ESC. The EISA Clock Divisor register bits[2:0] select the divide value.

The ESC provides the EISA Bus clock as the BCLKOUT output. Although the ESC is capable of driving 240 pF load on the BCLKOUT pin, it is recommended that this signal be buffered to protect the EISA BCLK signal.

The ESC EISA control logic and EISA interface is synchronous to the BCLK input. A maximum delay of 15 ns is allowed between the BCLKOUT output and the BCLK input for proper device functionality.

Table 9. PCICLK and BCLK Frequency Relationship

PCICLK (MHz)	DIVISOR (Programmable)	BCLK (MHz)
25	3	8.33
30	4*	7.5
33.3	4*	8.33

NOTE:
The ESC wakes up after reset with a default divisor value of 4.

5.2.1 CLOCK STRETCHING

The ESC is capable of stretching EISA Bus clock (BCLKOUT) for PCEB generated EISA cycles. The ESC stretches the EISA Bus clock (BCLKOUT) in order to minimize the synchronization penalty between PCI clock and EISA clock for accesses to EISA Bus by PCI agents. The PCEB initiates an EISA cycle by asserting START# synchronous to PCICLK. The ESC ensures the START# minimum pulse width is met by stretching the EISA Bus clock low time.

The ESC samples START# on every PCICLK when the PCEB has the EISA Bus. After sampling START# asserted, the ESC delays the rising edge of BCLKOUT until the START# has met the 115 ns minimum pulse width specification.

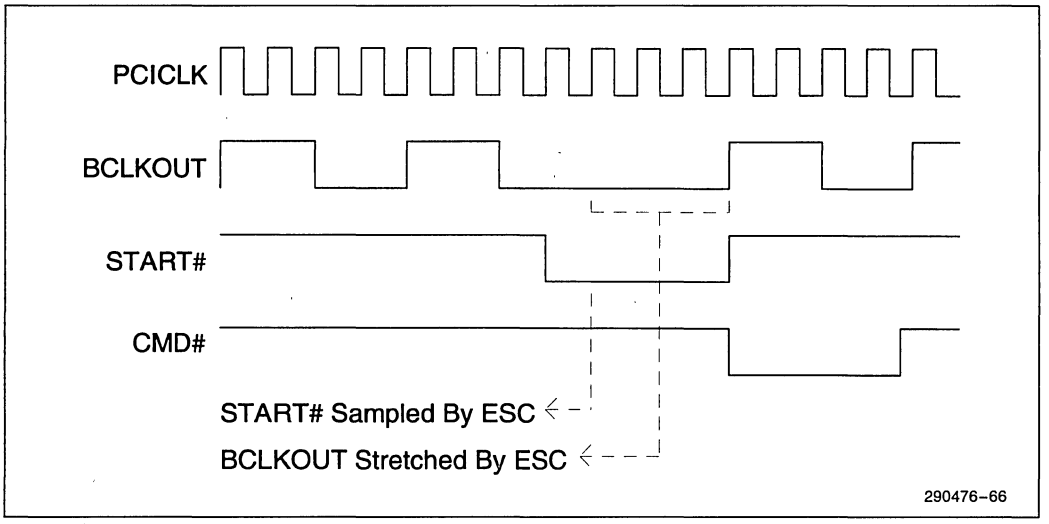


Figure 3. BCLK Stretching

5.3 EISA Master Cycles

EISA Master cycles are initiated on the EISA bus by an EISA Master (including PCEB for PCI agents). These cycles are accesses to the following resources:

- EISA slaves devices (including PCEB for PCI agents)
- ISA slave devices
- ESC internal registers (8-bit EISA Slave)

An EISA master gains control of the bus by asserting MREQx# (PEREQ# in case of PCEB) to the ESC. The ESC, after performing the necessary arbitration, asserts the corresponding MACKn# (negates EISAHOLD in case of the PCEB). Refer to Section 7.0 for arbitration protocol.

In response to receiving the acknowledge signal, the EISA Master starts the cycle by driving the bus with LA[31:02], BE[3:0], W/R, and M/IO. The EISA Master then asserts START# to indicate the beginning of the current cycle. A 16-bit EISA Master will also assert MASTER16# at this time. The ESC generates SBEH#, S1, and S0 signals from the BE[3:0]# signals.

5.3.1 EISA MASTER TO 32-BIT EISA SLAVE

An EISA slave after decoding it's address asserts EX32# or EX16#. The EISA master and the ESC use these signals to determine the EISA slave data size. The 32-bit or 16-bit EISA master continues with the cycles if EX32# or EX16# is asserted respectively. The ESC acts as a central resources for the EISA master and generates CMD# for the cycles. The ESC asserts CMD# on the same BCLK edge that START# is negated. The ESC monitors the EXRDY signal on the EISA bus to determine when to negate the CMD#. An EISA Slave can extend the cycle by negating EXRDY. EISA specification require that EXRDY not be held negated for more than 2.5 μs. A burstable EISA slave asserts SLBURST# signal the same time the slave decodes it's address. The EISA master will sample SLBURST# and assert MSBURST# if it is capable of bursting. The ESC keeps the CMD# asserted during a burst EISA transfer. The ESC deasserts CMD# to indicate the end of the burst transfer after the EISA master deasserts MSBURST#.

If EX16# is asserted, a 32-bit EISA master backs-off the bus by floating BE[3:0]# and START# (see Section 5.3.4). The ESC acts as a central resource for the EISA master in this case and takes over the mastership of the EISA bus by deriving START#, CMD#, and the appropriate byte enables. The ESC generates the necessary translation cycles for the EISA master and returns the bus ownership to the master by asserting EX32# and EX16#. The ESC monitors the EXRDY signal on the EISA bus to determine when to negate the CMD#. An EISA Slave can extend the cycle by negating EXRDY. EISA specification require that EXRDY not be held negated for more than 2.5 μs. A burstable EISA slave will assert the SLBURST# signal the same time when its address is decoded. The EISA master will sample SLBURST# and assert MSBURST# if it is capable of bursting. The ESC keeps the CMD# asserted during a burst EISA transfer. The ESC deasserts CMD# to indicate the end of the burst transfer after the EISA master deasserts MSBURST#.

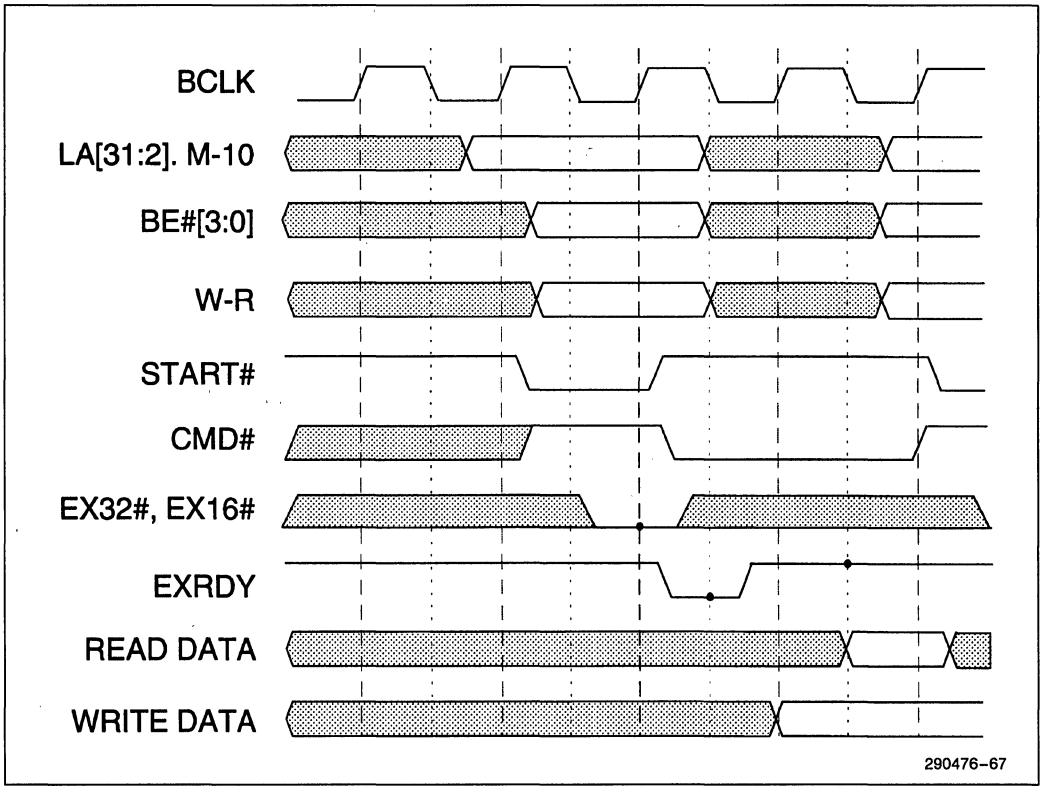


Figure 4. Standard EISA Master to EISA Slave Cycle

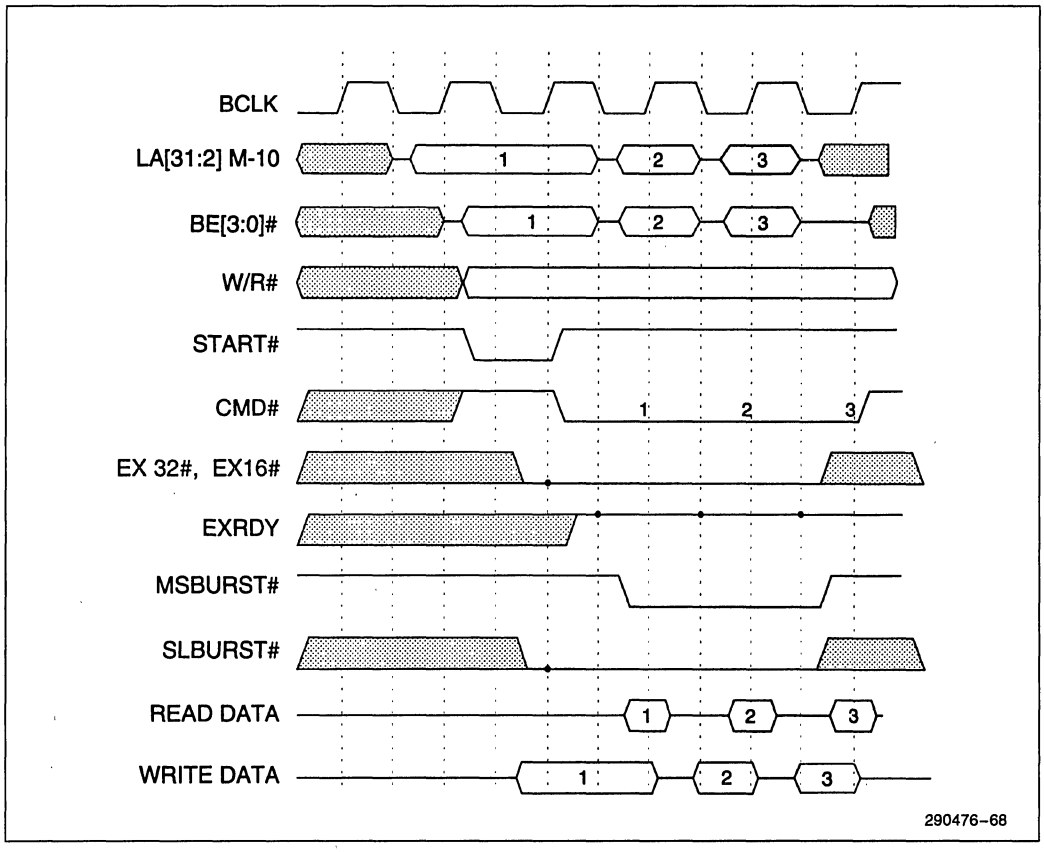


Figure 5. Burst EISA Master to EISA Slave Cycle

5.3.2 EISA MASTER TO 16-BIT ISA SLAVE

An ISA slave, after decoding it's address, asserts M16# or IO16#. The ESC monitors the EX32#, EX16#, M16#, and IO16# signals to determine the slave type. If EX32# and EX16# are negated and M16# or IO16# is asserted, the ESC performs ISA translation cycles for the EISA Bus master by generating BALE, MRDC#, MWRC#, IORC#, IOWC# signals as appropriate. The ISA slave can add wait states by negating CHRDY. The ESC samples CHRDY and translate it into EXRDY.

5.3.3 EISA MASTER TO 8-BIT EISA/ISA SLAVES

An 8-bit slave does not positively acknowledge it's selection by asserting any signal. The absence of an asserted EX32#, EX16#, M16#, and IO16# indicate to the ESC that an 8-bit device has been selected. The EISA master is backed-off the bus, and the ESC takes over mastership of the EISA/ISA bus. The ESC will run 8-bit translation cycles on the bus by deriving the EISA control signals and the ISA control signals. A slave can extend the cycles by negating EXRDY or CHRDY signals.

The ESC (Internal Registers) is accessed as an 8-bit slave.

5.3.4 EISA MASTER BACK-OFF

During EISA master transfer where the master and slave size is mis-matched, the EISA master is required to back-off the bus on the first falling edge of BCLK after START# is negated. The EISA master floats its START#, BE[3:0]#, and data lines at this time. This allows the ESC to performs translation cycle. The master must back-off the bus if a master/slave data size mis-match is determined, regardless if data size translation is performed.

At the end of the data size translation or transfer cycle control is transferred back to the bus master by the ESC by driving EX32# and EX16# active on the falling edge of BCLK, before the rising edge of BCLK that the last CMD# is negated. An additional BCLK is added at the end of the transfer to allow the exchanging of cycle control to occur.

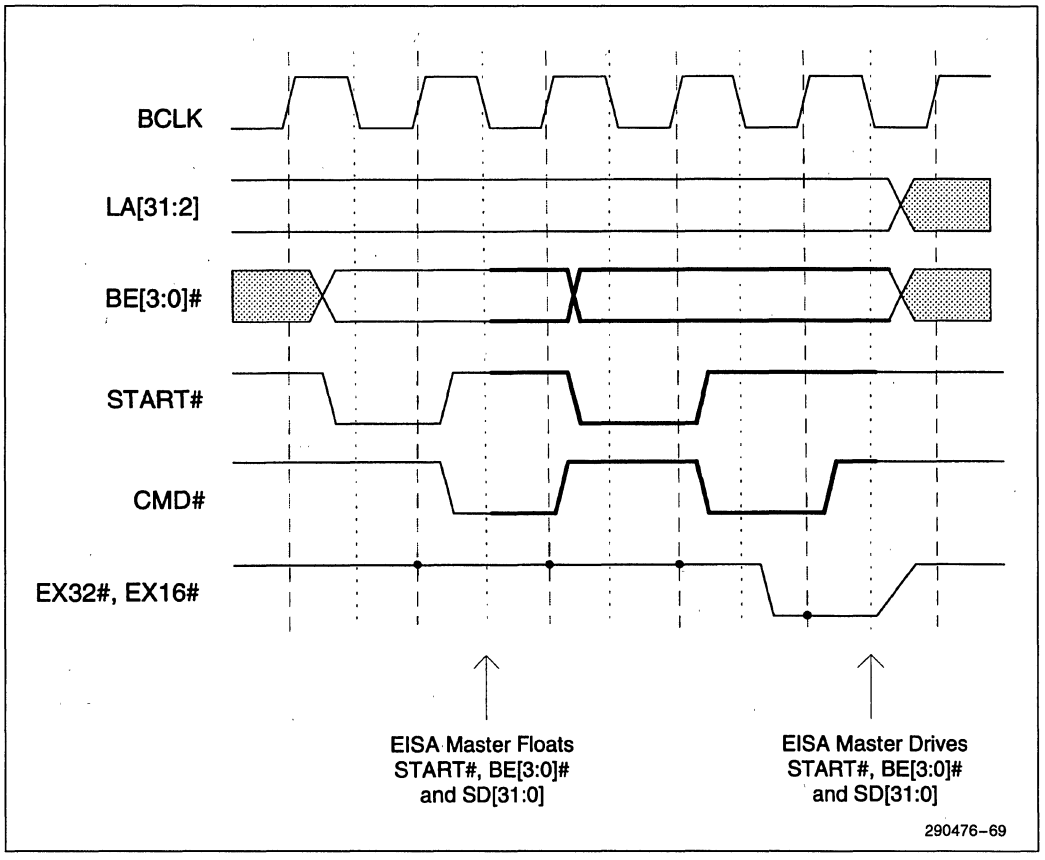


Figure 6. EISA Master Back Off Cycle

5.4 ISA Master Cycles

ISA cycles are initiated on the ISA bus by an ISA master. These cycles are accesses to the following system resources:

- EISA slaves devices (including PCEB for PCI agents).
- ISA slave devices.
- ESC internal registers (8-bit EISA Slave).

The ISA Master initiates such a cycle by asserting the DREQx# line to the ESC. The ESC, after performing the necessary arbitration, asserts the corresponding DACKx# line. Upon receiving an acknowledge from the ESC, the ISA master asserts the MASTER16# signal line to indicate that it has control of the ISA bus and a cycle on the ISA bus will take place. The ESC translates the ISA address signals SBHE#, SA1, and SA0 to EISA byte enables BE[3:0]#.

5.4.1 ISA MASTER TO 32-/16-BIT EISA SLAVE

An EISA slave will decode the address to determine if it has been selected. In response to a positive decode, the EISA slave will assert EX32# or EX16#. The ESC samples these signals to determine if an EISA Slave has been selected. If these signals are asserted, the ESC will perform ISA to EISA cycle translation by driving the EISA control signals.

The ISA Master asserts one of the ISA command signals MRDC#, MWTC#, IORC# or IOWC# depending on whether or not the access is to a memory, an I/O device or an I/O register. The ISA command signals will remain active until the end of the cycle. The ESC will generate the EISA translation by generating the EISA control signals; START#, CMD#, M/IO#, and W/R#.

The EISA slave can add wait states by negating EXRDY. The ESC samples EXRDY and translates it into CHRDY. The ESC will also generate the control signals to steer the data to the appropriate byte lanes for mismatched cycles.

5.4.2 ISA MASTER TO 16-BIT ISA SLAVE

An ISA Master initiates cycles to ISA slave devices. These cycles are either memory read/write or I/O read/write. The ISA bus Master is assumed to be 16-bit device, and it can access either 8- or 16-bit slave devices that reside on the ISA bus. A 16-bit ISA slave device will respond to a valid address by asserting M16# for memory cycles and IO16# for I/O cycles. The ESC is inactive during ISA Master cycles where either M16# or IO16# is sampled asserted.

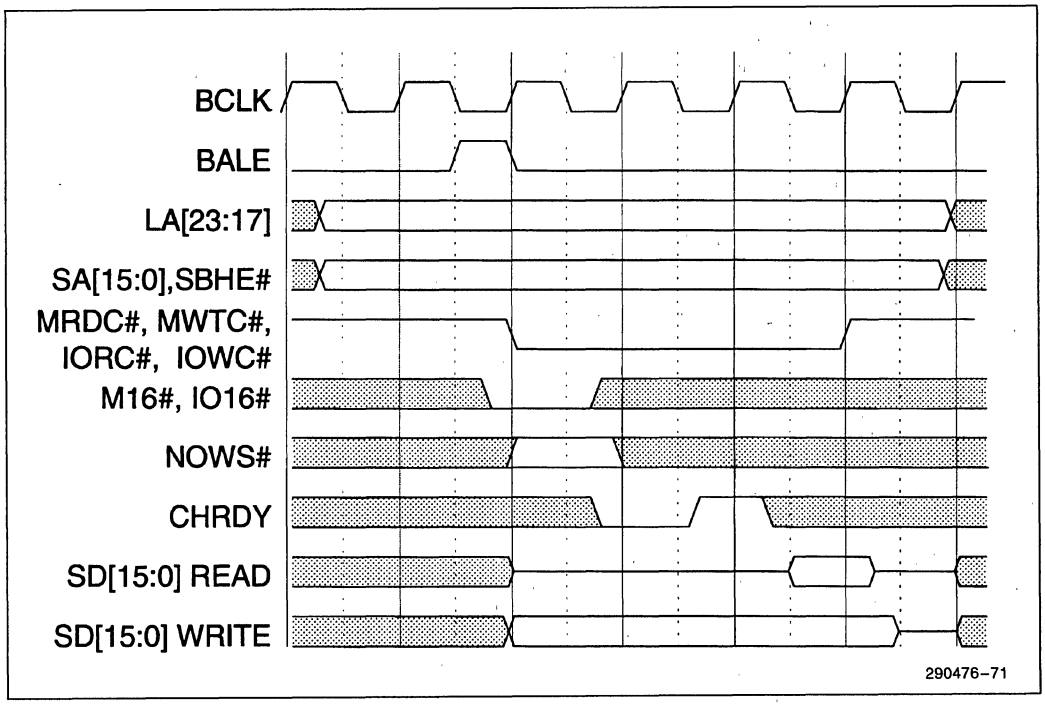


Figure 7. ISA Master to 16-Bit ISA Slave Cycles (3 BCLKs)

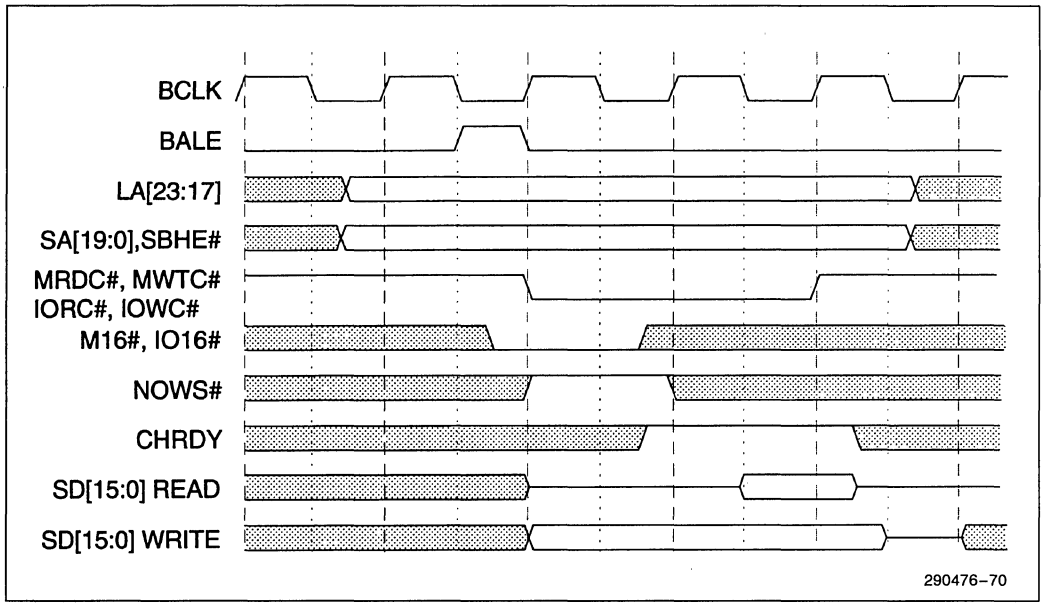


Figure 8. ISA Master to 16-Bit ISA Slave Extended Cycle (4 BCLKs)

5.4.3 ISA MASTER TO 8-BIT EISA/ISA SLAVE

An 8-bit slave does not positively acknowledge it's selection by asserting any signal. The absence of an asserted EX32#, EX16#, M16#, and IO16# indicate to the ESC that an 8-bit device has been selected. The EISA master is backed-off the bus, and the ESC takes over mastership of the EISA/ISA bus. The ESC will run 8-bit translation cycles on the bus by deriving the EISA control signals and the ISA control signals. A slave can extend the cycles by negating EXRDY or CHRDY signals. The ESC (Internal Registers) is accessed as an 8-bit slave.

5.4.4 ISA WAIT STATE GENERATION

There are three sources that can affect the generation of wait states for ISA cycles. The first is the default wait states, which determines the standard or default ISA bus cycle in the absence of any response from the slave. The second is cycle extension, which is indicated by the slave pulling the CHRDY signal line inactive (low). The CHRDY is high by default due to a pull-up resistor. Thus, the cycle will be extended until the CHRDY is returned to its active high value. The third way to change the number of wait states is when the slave asserts the NOWS# signal which makes the cycle shorter than the default or standard cycle.

ISA Memory slaves (8- and 16-bits) and ISA I/O slaves (only 8-bits) can shorten their default cycles by asserting the NOWS# signal lines. A 16-bit I/O slave cannot shorten its default cycles. When NOWS# is asserted at the same time the CHRDY is negated by the ISA slave device, NOWS# will be ignored and wait states will be added. (i.e.; CHRDY has precedence over NOWS#.)

DMA devices (I/O) cannot add wait states, but memory can. Table 10 shows the number of BCLKs for each cycle type (Memory, I/O, DMA), default, wait states added and with NOWS# asserted.

Table 10. Number of BCLKs for ISA Master Cycles

Cycle Type	Bus Size	No Wait State NOWS# = 0	Standard CHRDY = 1 NOWS# = 1	One Wait State CHRDY = 0
		Number of BCLKs		
Memory Read/Write	16	2	3	4
Memory Read/Write	8	4, 5	6	7
I/O Read/Write	16	3	3	4
I/O Read/Write	8	4, 5	6	7
DMA Compatible	8/16	8	8	10
DMA Type A(1)	8/16	NA	6	7
DMA Type B(1)	8/16	NA	4	5
DMA Type C(2)	8/16	NA	2	3

NOTES:
1. If ISA memory responds, the ESC will extend the cycle by 1 BCLK.
2. If ISA memory responds, the ESC will use DMA Type B read cycle timing.

5.5 Mis-Match Cycles

Data size translation is performed by the ESC for all mis-matched cycles. A mis-matched cycle is defined as a cycle in which the bus master and bus slave do not have equal data bus sizes (e.g., a 32-bit EISA master accessing a 16-bit ISA slave). The data size translation is performed in conjunction with the PCEB. The ESC generates the appropriate cycles and data steering control signals for mis-matched cycles. The PCEB uses the data steering control signals from the ESC to latch and redirect the data to the appropriate byte lanes. The ESC will perform one or more of the following operations depending on the master and slave type, transfer direction, and the number of byte enables active.

Table 11. Mis-Match Master Slave Combinations

Master Type	Cycle Type	Slave Type			
		32-Bit EISA	16-Bit EISA	16-Bit ISA	8-Bit EISA/ISA
32-bit EISA with 16-bit downshift	Standard Burst	match match	Mis-Match match	Mis-Match na	Mis-Match na
32-bit EISA	Standard Burst	match match	Mis-Match na	Mis-Match na	Mis-Match na
16-bit EISA	Standard Burst	Mis-Match Mis-Match	match match	Mis-Match na	Mis-Match na

NOTE:
na: Not Applicable. The cycle will never occur.

5.6 Data Swap Buffer Control Logic

For all mis-matched cycles, the ESC is responsible for performing data size translations. The ESC performs these data size translations by either becoming the master of the EISA/ISA Bus (see Section 5.3.4) or by directing the flow of data to the appropriate byte lanes. In both cases, the ESC generates Data Swap Buffer control signals to perform data size translation.

- SDCPYEN[13,3:1]
- SDCPYUP
- SDOE[2:0]#
- SDLE[3:0]#

The Data Swap Buffers are integrated in the PCEB (see PCEB data sheet Section 8.0 for Data Swap Buffer function description). The data size translation cycles consist of one or combinations of Assembly, Disassembly, Copy Up/Down, and Redrive.

ASSEMBLY

This occurs during reads when an EISA master data size is greater than the slave data size. ISA masters are required to perform assemble when accessing 8-bit slaves. Assembly consists of two, three, or four cycles depending on the master data size, slave data size, and number of active byte enables. During the assembly process, the data is latched in to the PCEB data latch/buffers. This data is driven or redriven on to the EISA bus during the last cycle. The master after initiating the cycle backs-off the bus (see the EISA master back-off section for details) when a mis-matched is detected. The ESC becomes the bus master and runs the appropriate number of cycles. At the end of the last cycle, the ESC transfer the control of bus back to the original master.

DISASSEMBLY

This occurs during writes when the EISA master data size is greater than the slave data size. ISA masters are required to perform disassemble when accessing 8-bit slaves. Disassembly consists of two, three, or four cycles depending on the master data size, slave data size, and number of active byte enables. During the disassembly process, the data is latched in the PCEB latch/buffers on the first cycle. This data is driven or redriven on to the EISA bus on subsequent cycles. The master after initiating the cycle backs-off the bus (see the EISA master back-off section for details) when a mis-matched is detected. The ESC becomes the bus master and runs the appropriate number of cycles. At the end of the last cycle, the ESC transfer the control of bus back to the original master.

COPY-UP

This occurs during reads when the master data size is greater than the slave data size and during writes when the master data size is smaller than the slave data size. The copy-up function is used for cycles with and without assembly/disassembly.

COPY-DOWN

This occurs during writes when the master data size is greater than the slave data size and during reads when the master data size is smaller than the slave data size. The copy-down function is used for cycles with and without assembly/disassembly.

RE-DRIVE

This occurs during reads and writes when both the master and slave are on the EISA/ISA bus and the PCEB is neither a master nor a slave. The re-drive function is always performed in conjunction with assembly/disassembly. During the assembly process, the last cycle is a re-drive cycle. During disassembly, all the cycles except the first cycle are re-drive cycles.

5.7 Servicing DMA Cycles

The ESC is responsible for performing DMA transfers. If the memory is determined (EX32# or EX16# asserted) to be on the EISA bus, the DMA cycle can be "A", "B", or "C" type. If the memory is determined to be on the ISA bus, then the DMA cycle will run as a compatible cycle. The DMA transfers are described in detail in Section 8.0.

5.8 Refresh Cycles

The ESC support refresh cycles on the EISA/ISA bus. The ESC asserts the REFRESH# signal to indicate when a refresh cycle is in progress. Refresh cycles are generated by two sources: the refresh unit inside the ESC or an external ISA bus masters. The EISA bus controller will enable the address lines LA[15:2] and the BE[3:0]#. The High and Low Page register contents will also be placed on the LA[31:16] bus during refresh. Memory slaves on the EISA/ISA bus must not drive any data onto the data bus during the refresh cycle. Slow memory slaves on the EISA/ISA may extend the refresh cycle by negating the EXRDY or CHRDY signal respectively. The refresh cycles are also described in Section 6.11.

5.9 EISA Slot Support

The ESC support up of 8 EISA slots. The ESC provides support for the 8 slots as follows:

- The ESC address and data output buffers directly drive 240 pF capacitive load on the Bus.
- The ESC generates slot specific AENx signals.
- The ESC supports EISA masters in all 8 slots.

The ESC generates encoded AENs and encoded Master Acknowledge signals for 8 slots and 8 masters. These signals must to decoded on the system board to generate the slot specific AENx signals and MACKx# signals. The ESC can be programmed through Mode Select register bit[1:0] to directly generate these signals for 4 slots and 4 masters.

5.9.1 AEN GENERATION

The ESC directly generates the slot specific AEN signals if the ESC is configured to support 4 AENx (Table 12). If the ESC is programmed to support more than 4 EISA AENx, the ESC will generate Encoded AEN signals. Discrete logic like a F138 is required to generate the slot specific AENs.

Table 12. AEN Generation

Cycle	A[15:12]	A[11:8]	A[7:4]	A[3:0]	AEN4	AEN3	AEN2	AEN1
DMA	xxxx	xxxx	xxxx	xxxx	1	1	1	1
IO	0000	xx00	xxxx	xxxx	1	1	1	1
IO	0001	xx00	xxxx	xxxx	1	1	1	0
IO	0010	xx00	xxxx	xxxx	1	1	0	1
IO	0011	xx00	xxxx	xxxx	1	0	1	1
IO	0100	xx00	xxxx	xxxx	0	1	1	1
IO	0101-1111	xx00	xxxx	xxxx	1	1	1	1
IO	xxxx	xx01	xxxx	xxxx	0	0	0	0
IO	xxxx	xx10	xxxx	xxxx	0	0	0	0
IO	xxxx	xx11	xxxx	xxxx	0	0	0	0
MEM	xxxx	xxxx	xxxx	xxxx	0	0	0	0

Table 13. Encoded AEN (AEN) Generation

Cycle	A[15:12]	A[11:8]	A[7:4]	A[3:0]	EAEN4	EAEN3	EAEN2	EAEN1
DMA	xxxx	xxxx	xxxx	xxxx	1	1	1	1
IO	0000	xx00	xxxx	xxxx	1	1	1	1
IO	0001	xx00	xxxx	xxxx	0	0	0	1
IO	0010	xx00	xxxx	xxxx	0	0	1	0
IO	0011	xx00	xxxx	xxxx	0	0	1	1
IO	0100	xx00	xxxx	xxxx	0	1	0	0
IO	0101	xx00	xxxx	xxxx	0	1	0	1
IO	0110	xx00	xxxx	xxxx	0	1	1	0
IO	0111	xx00	xxxx	xxxx	0	1	1	1
IO	1000	xx00	xxxx	xxxx	1	0	0	0
IO	1001-1111	xx00	xxxx	xxxx	1	1	1	1
IO	xxxx	xx01	xxxx	xxxx	0	0	0	0
IO	xxxx	xx10	xxxx	xxxx	0	0	0	0
IO	xxxx	xx11	xxxx	xxxx	0	0	0	0
MEM	xxxx	xxxx	xxxx	xxxx	0	0	0	0

NOTE:
EAEN[4:1] combinations not specified in the table are Reserved.

5.9.2 MACKX# GENERATION

The ESC generates the EISA Master Acknowledge signals if the ESC is configured for to directly support 4 masters through Mode Select register bit[1:0]. In this case the ESC generates MACKx#s for Master 0-3. If the ESC is programmed to support more than 4 EISA slots, the ESC will generate Encoded (E)MACKx#s. Discrete logic like a F138 is required to generate the MACKx#s for the Masters.

Table 14. Encoded MACK# (EMACKx#) Generation

EMACK [4:1]	MACK7#	MACK6#	MACK5#	MACK4#	MACK3#	MACK2#	MACK1#	MACK0#
0000	1	1	1	1	1	1	1	0
0001	1	1	1	1	1	1	0	1
0010	1	1	1	1	1	0	1	1
0011	1	1	1	1	0	1	1	1
0100	1	1	1	0	1	1	1	1
0101	1	1	0	1	1	1	1	1
0110	1	0	1	1	1	1	1	1
0111	0	1	1	1	1	1	1	1
1111	1	1	1	1	1	1	1	1

NOTE:
EMACK[4:1] combinations 1000–1110 are Reserved.

6.0 DMA CONTROLLER

6.1 DMA Controller Overview

The DMA circuitry incorporates the functionality of two 82C37 DMA controllers with seven independently programmable channels, (Channels 0-3 and Channels 5-7). DMA Channel 4 is used to cascade the two controllers together and will default to cascade mode in the Mode register. In addition to accepting requests from DMA slaves, the DMA also responds to requests that are initiated by software. Software may initiate a DMA service request by setting any DMA Channel Request register bit to a 1. The DMA controller for Channels 0-3 is referred to as "DMA-1" and the controller for Channels 4-7 is "DMA-2".

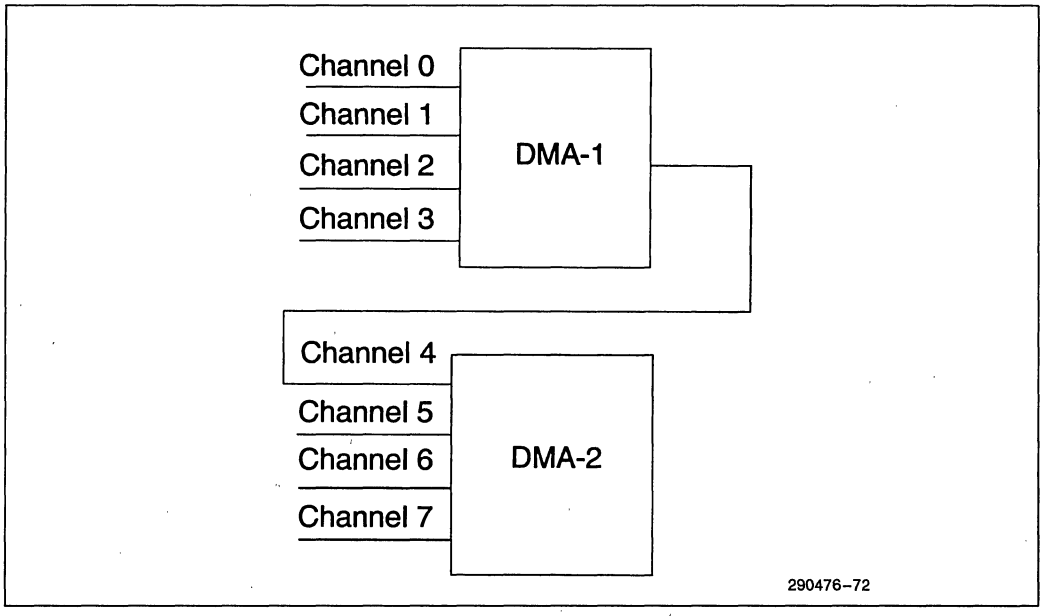


Figure 9. Internal DMA Controller

Each DMA channel can be programmed for 8- or 16-bit DMA device size. Each channel can also be programmed for compatibility, Type "A", Type "B", or Type "C"(burst transfer) timings. Each DMA channel defaults to the PC-AT compatible settings for DMA device size: channels [3:0] default to 8-bit, count-by-bytes transfers, while channels [7:5] default to 16-bit, count-by-words (address shifted) transfers. The ESC provides the timing control and data size translation necessary for DMA transfers between EISA/ISA agents of mismatched bus sizes.

The DMA Controller supports full 32-bit addressing. Each channel includes a 16-bit ISA compatible Current register which holds the 16 least-significant bits of the 32-bit address, and an ISA compatible Low Page register which contains the eight second most significant bits. An additional High Page register contains the eight most significant bits of the 32-bit address. The address counter can be programmed as either 16-bit compatible address counter or a full 32-bit address counter.

The channels can also be programmed for any of four transfer modes. The transfer modes include single, block, demand, or cascade. Each of the three active transfer modes (single, block, and demand), can perform three different types of transfers (read, write, or verify).

The DMA Controller also features refresh address generation, and auto-initialization following a DMA termination. EISA compatible buffer chaining is included as well as Stop registers to support ring buffer structures.

Scatter-Gather reduces CPU overhead by eliminating reprogramming of the DMA and I/O between buffers as well as reducing the number of interrupts.

The DMA Controller includes the EISA Bus arbiter which works with the PCEB's PCI bus arbiter. The arbiter determines which requester from among the requesting DMA slaves, EISA bus masters, the PCI bus, or Refresh should have the bus.

The DMA Controller is at any time either in master mode or slave mode. In master mode, the DMA controller is either servicing a DMA slave's request for DMA cycles, allowing an ISA master to use the bus via a cascaded DREQ signal, or granting the bus to an EISA master via MREQ#/MACK#. In slave mode, the ESC monitors both the EISA bus decoding and responding to I/O read and write commands that address its registers.

When the DMA is in master mode and servicing a DMA slave, it works in conjunction with the ESC EISA bus controller to create bus cycles on the EISA bus. The DMA places addresses onto the internal address bus and the bus controller informs the DMA when to place a new address on the internal bus.

6.2 DMA Transfer Modes

The channels can be programmed for any of four transfer modes. The transfer modes include single, block, demand, or cascade. Each of the three active transfer modes (single, block, and demand), can perform three different types of transfers (read, write, or verify). The ESC does not support memory to memory transfers.

6.2.1 SINGLE TRANSFER MODE

In Single Transfer mode the DMA is programmed to make one transfer only. The byte/word count will be decremented and the address decremented or incremented following each transfer. When the byte/word count "rolls over" from zero to FFFFFFh, or an external EOP is encountered, a Terminal Count (TC) will load a new buffer via Scatter-Gather, buffer chaining or autoinitialize if it is programmed to do so.

DREQ must be held active until DACK becomes active in order to be recognized. If DREQ is held active throughout the single transfer, the bus will be released to the CPU after a single transfer. With the DREQ asserted high, the DMA I/O device will rearbitrate for the bus. Upon winning the bus, another single transfer will be performed. This allows other bus masters a chance to arbitrate for, win, and execute cycles on the EISA Bus.

6.2.2 BLOCK TRANSFER MODE

In Block Transfer mode the DMA is activated by DREQ to continue making transfers during the service until a TC, caused by either a byte/word count going to FFFFFFh or an external EOP, is encountered. DREQ need only be held active until DACK becomes active. If the channel has been programmed for it, a new buffer will be loaded by buffer chaining or auto-initialization at the end of the service. In this mode, it is possible to lock out other devices for a period of time (including refresh) if the transfer count is programmed to a large number and Compatible timing is selected. Block mode can effectively be used with Type "A", Type "B", or Burst timing since the channel can be interrupted through the 4 μs timeout mechanism, and other devices (or Refresh) can arbitrate for and win the bus. See Section 7.0 on the EISA Bus Arbitration for a detailed description of the 4 μs timeout mechanism. Note that scatter-gather block mode is not supported.

6.2.3 DEMAND TRANSFER MODE

In Demand Transfer mode the DMA channel is programmed to continue making transfers until a TC (Terminal Count) is encountered or an external EOP is encountered, or until the DMA I/O device pulls DREQ inactive. Thus, transfers may continue until the I/O device has exhausted its data capacity. After the I/O device catches up, the DMA service is re-established when the DMA I/O device reasserts the channel's DREQ. During the time between services when the system is allowed to operate, the intermediate values of address and byte/word count are stored in the DMA controller Current Address and Current Byte/Word Count registers. A TC can cause a new buffer to be loaded via Scatter-Gather, buffer chaining or autoinitialize at the end of the service if the channel has been programmed for it.

6.2.4 CASCADE MODE

This mode is used to cascade more than one DMA controller together for simple system DMA requests for the additional device propagate through the priority network circuitry of the preceding device. The priority chain is preserved and the new device must wait for its turn to acknowledge requests. Within the ESC architecture, Channel 0 of DMA Controller two (DMA-2, Ch 4) is used to cascade DMA Controller one (DMA-1) to provide a total of seven DMA channels. Channel 0 on DMA-2 (labeled Ch 4 overall) connects the second half of the DMA system. This channel is not available for any other purpose.

In Cascade Mode, the DMA Controller will respond to DREQ with DACK, but the ESC will not drive the bus.

Cascade mode is also used to allow direct access of the system by 16-bit bus masters. These devices use the DREQ and DACK signals to arbitrate for the system bus and then they drive the address and command lines to control the bus. The ISA master asserts its ISA master request line (DREQx) to the DMA internal arbiter. If the ISA master wins the arbitration, the ESC responds with an ISA Master Acknowledge (DACKx) signal active. Upon sampling the DACKx line active, the ISA Master asserts MASTER16# signal and takes control of the EISA bus. The ISA Master has control of the EISA Bus, and the ISA Master may run cycles until it negates the MASTER16# signal.

6.3 DMA Transfer Types

Each of the three active transfer modes (Single, Block, or Demand) can perform three different types of transfers. These transfers are Read, Write and Verify.

Write Transfer

Write transfers move data from an EISA/ISA I/O device to memory located on EISA/ISA Bus or PCI Bus. The DMA indicates the transfer type to the EISA bus controller. The bus controller will activate IORC# and the appropriate EISA control signals (M/IO# and W/R#) to indicate a memory write.

Read Transfer

Read transfers move data from EISA/ISA or PCI memory to an EISA/ISA I/O device. The DMA indicates the transfer type to the EISA bus controller. The bus controller will activate IOWC# and the appropriate EISA control signals (M/IO# and W/R#) to indicate a memory read.

Verify Transfer

Verify transfers are pseudo transfers. The DMA controller operates as in Read or Write transfers, generating addresses and producing TC, etc. However, the ESC does not assert the memory and I/O control signals. Only the DACK signals are asserted. Internally the DMA controller will count BCLKs so that the DACK signals have a defined pulse width. This pulse width is nine BCLKs long. If Verify transfers are repeated during Block or Demand DMA requests, each additional pseudo transfer will add eight BCLKs. The DACK signals will not be toggled for repeated transfers.

6.4 DMA Timing

The ESC DMA provides four transfer timings. In addition to the compatible timings, the ESC DMA provides Type "A", Type "B", and Type "C" (burst) timings for I/O slave devices capable of running at faster speeds.

6.4.1 COMPATIBLE TIMINGS

Compatible timing is provided for DMA slave devices. Compatible timing runs at 9 BCLKs (1080 ns/single cycle) and 8 BCLKs (960 ns/cycle) during the repeated portion of a Block or Demand mode transfers.

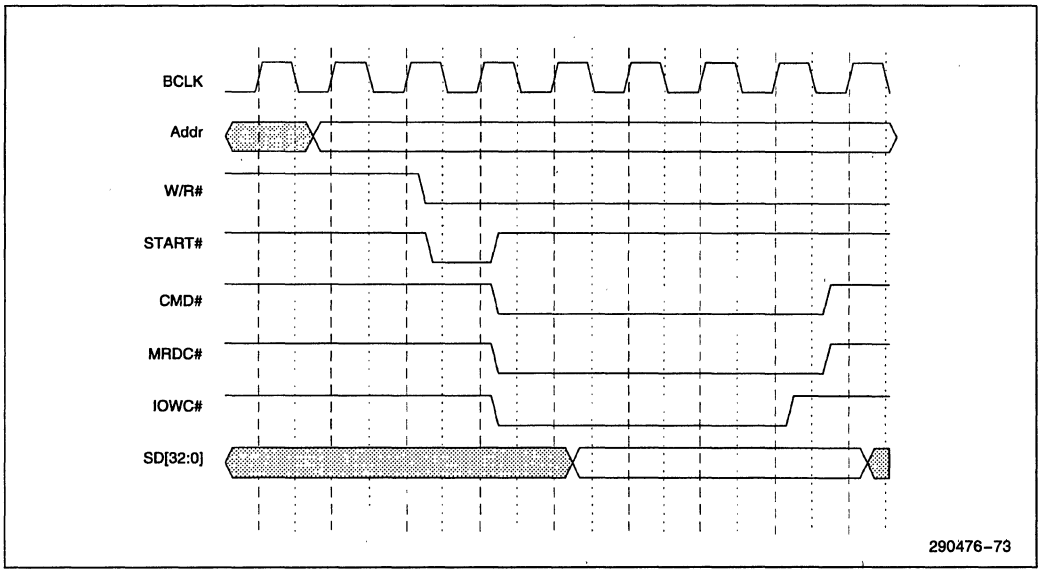


Figure 10. Compatible DMA Read Transfer (8 BCLKs)

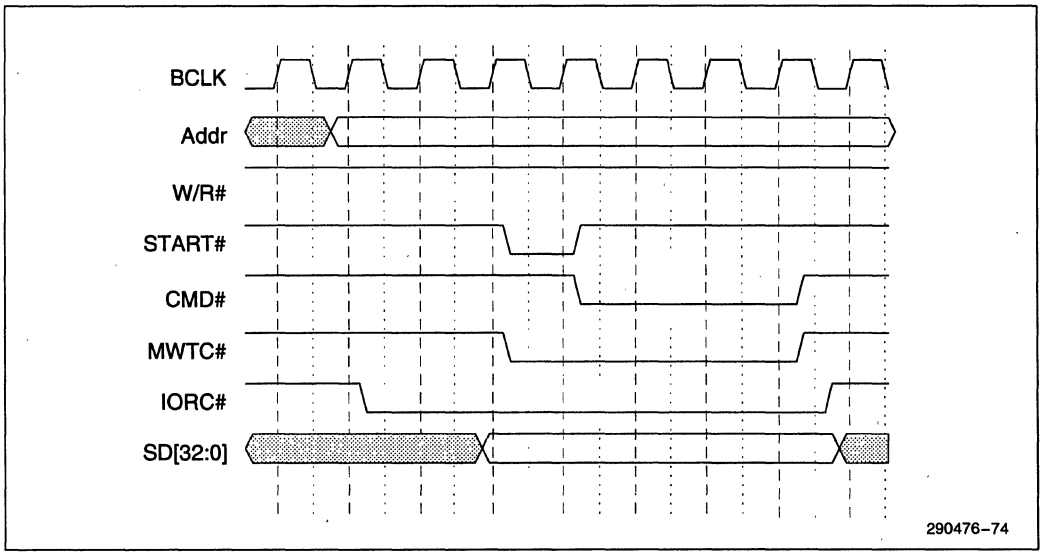


Figure 11. Compatible DMA Write Transfer (8 BCLKS)

6.4.2 TYPE "A" TIMING

Type "A" timing is provided to allow shorter cycles to EISA memory. (Note: Main memory behaves like EISA memory because the PCEB has an EISA slave interface.) Type "A" timing runs at 7 BCLKs (840 ns/single cycle) and 6 BCLKs (720 ns/cycle) during the repeated portion of a Block or Demand mode transfer. Type "A" timing varies from compatible timing primarily in shortening the memory operation to the minimum allowed by system memory. The I/O portion of the cycle (data setup on write, I/O read access time) is the same as with compatible cycles. The actual active command time is shorter, but it is expected that the DMA devices which provide the data access time or write data setup time should not require excess IORC# or IOWC# command active time. Because of this, most DMA devices should be able to use type "A" timing.

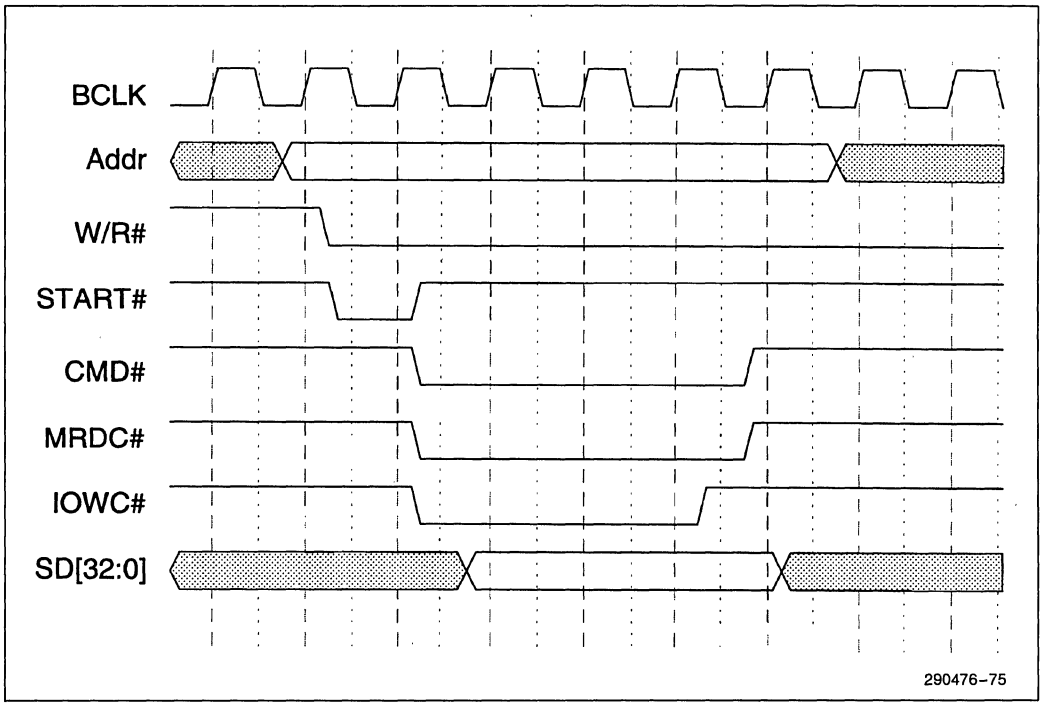


Figure 12. Type "A" DMA Read Transfers (6 BCLKS)

6.4.3 TYPE "B" TIMING

Type "B" timing is provided for 8-/16-bit DMA devices which can accept faster I/O timing. Type "B" only works with fast system memory. Type "B" timing runs at 6 BCLKs (720 ns/single cycle) and 4 BCLKs (480 ns/cycle) during the repeated portion of a block or demand mode transfer. Type "B" timing requires faster DMA slave devices than compatible timing in that the cycles are shortened so that the data setup time on I/O write cycles is shortened and the I/O read access time is required to be faster. Some of the current ISA devices should be able to support type "B" timing, but these will probably be more recent designs using relatively fast technology.

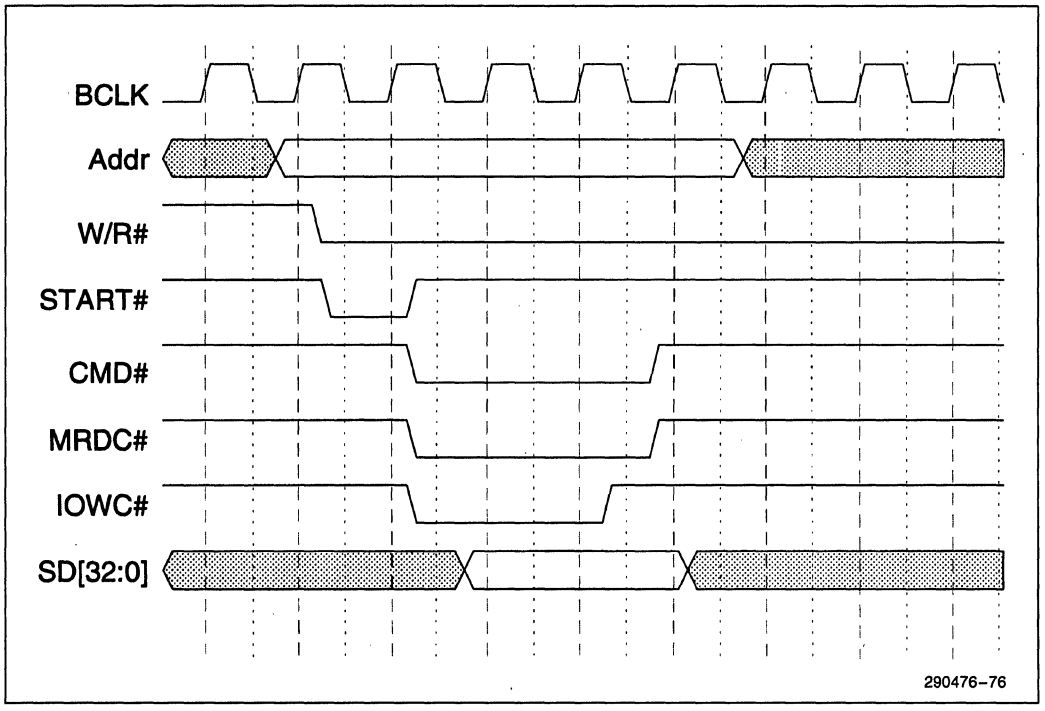


Figure 13. Type "B" DMA Read Transfer (4 BCLKS)

6.4.4 TYPE "C" (BURST) TIMING

Type "C" (burst) timing is provided for EISA DMA devices. The DMA slave device needs to monitor EXRDY and IORC# or IOWC# signals to determine when to change the data (on writes) or sample the data (on reads). This timing will allow up to 33 MBytes per second transfer rate with a 32-bit DMA device and 32-bit memory. Note that 8- or 16-bit DMA devices are supported (through the programmable DMA address increment) and that they use the "byte lanes" natural to their size for the data transfer. As with all bursts, the system will revert to two BCLK cycles if the memory does not support burst. When a DMA burst cycle accesses non-burst memory and the DMA cycle crosses a page boundary into burstable memory, the ESC will continue performing standard (non-burst) cycles. This will not cause a problem since the data is transferred correctly.

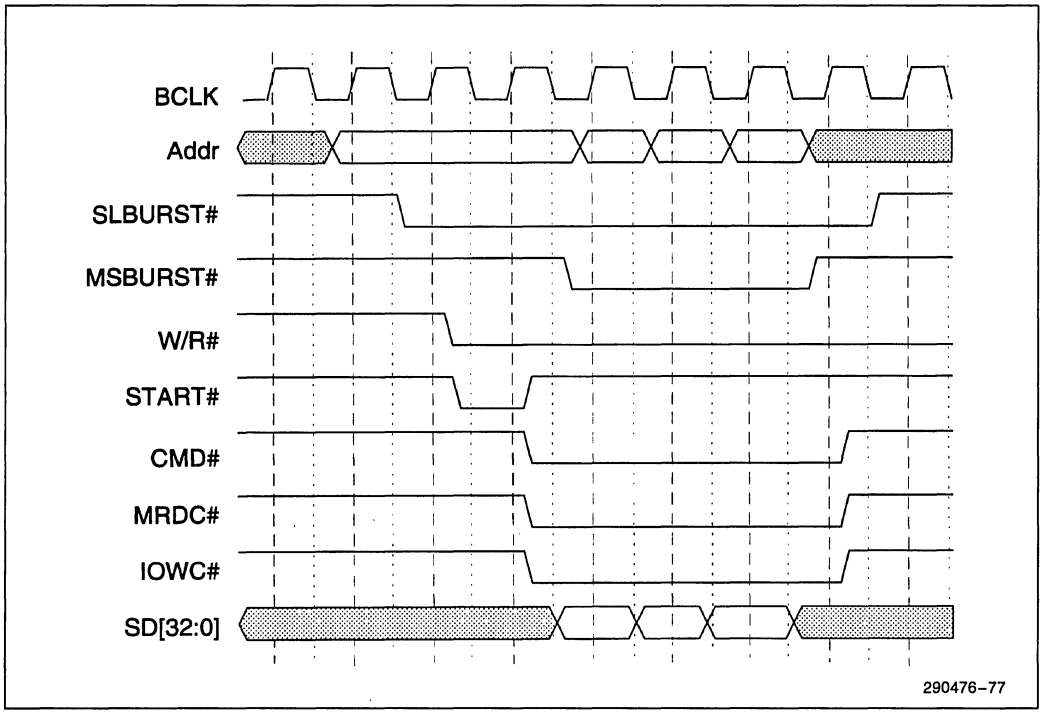


Figure 14. Type "C" (Burst) DMA Read Transfers (1 BCLK)

6.5 Channel Priority

For priority resolution the DMA consists of two logical channel groups-channels 0-3 and channels 4-6. Each group may be in either Fixed or Rotate mode, as determined by the Command register.

For arbitration purposes, the source of the DMA request is transparent. DMA I/O slaves normally assert their DREQ line to arbitrate for DMA service. However, a software request for DMA service can be presented through each channel's DMA Request register. A software request is subject to the same prioritization as any hardware request. Please see the detailed register description in Section 3.0 for Request Register programming information.

Fixed Priority

The initial fixed priority structure is as follows:

High priority	Low priority
(0, 1, 2, 3) 5, 6, 7	

The fixed priority ordering is 0, 1, 2, 3, 5, 6, and 7. In this scheme, Channel 0 has the highest priority, and Channel 7 has the lowest priority. Channels [3:0] of DMA-1 assume the priority position of Channel 4 in DMA-2, thus taking priority over Channels 5, 6, and 7.

Rotating Priority

Rotation allows for "fairness" in priority resolution. The priority chain rotates so that the last channel serviced is assigned the lowest priority in the Channel group (0-3, 5-7). Channels 0-3 rotate as a group of 4. They are always placed between Channel 5 and Channel 7 in the priority list. Channel 5-7 rotate as part of a group of 4. That is, Channels (5-7) form the first three partners in the rotation, while Channel group (0-3) comprises the fourth position in the arbitration. Table 15 demonstrates rotation priority:

Table 15. Rotating Priority Example

Programmed Mode	Action	Priority High .....Low
Group (0-3) is in rotation mode	1) Initial Setting	(0, 1, 2, 3), 5, 6, 7
Group (4-7) is in fixed mode.	2) After servicing channel 2	(3, 0, 1, 2), 5, 6, 7
	3) After servicing channel 3	(0, 1, 2, 3), 5, 6, 7
Group (0-3) in rotation mode	1) Initial Setting	(0, 1, 2 3), 5, 6, 7
Group (4-7) is in rotation mode	2) After servicing channel 0	5, 6, 7, (1, 2, 3, 0)
	3) After servicing channel 5	6, 7, (1, 2, 3, 0), 5
(note that the first servicing of	4) After servicing channel 6	7, (1, 2, 3, 0), 5, 6
channel 0 caused double rotation).	5) After servicing channel 7	(1, 2, 3, 0), 5, 6, 7

6.6 Scatter-Gather Functional Description

Scatter-Gather provides the capability of transferring multiple buffers between memory and I/O without CPU intervention. In Scatter-Gather, the DMA can read the memory address and word count from an array of buffer descriptors called the Scatter-Gather Descriptor (SGD) Table. This allows the DMA to sustain DMA transfers until all buffers in the Scatter-Gather Descriptor Table are transferred.

The Scatter-Gather Command register and Scatter-Gather Status register are used to control the operational aspect of Scatter-Gather transfers (see Section 3.2 for details of these registers). The Scatter-Gather Descriptor Next Link register holds the address of the next buffer descriptor in the Scatter-Gather Descriptor Table.

The next buffer descriptor is fetched from the Scatter-Gather Descriptor Table by a DMA read transfer. DACK# will not be asserted for this transfer because the I/O device is the DMA itself and the DACK is internal to the ESC. The ESC will assert IOWC# for these bus cycles like any other DMA transfer. The ESC will behave as an 8-bit I/O slave and will run type "B" timings for a Scatter-Gather buffer descriptor transfer. EOP will be asserted at the end of the transfer.

To initiate a typical Scatter-Gather transfer between memory and an I/O device the following steps are involved:

1. Software prepares a Scatter-Gather Descriptor (SGD) table in system memory. Each Scatter-Gather descriptor is 8 bytes long and consists of an address pointer to the starting address and the transfer count of the memory buffer to be transferred. In any given SGD table, two consecutive SGDs are offset by 8 bytes and are aligned on a 4-byte boundary.
2. Each Scatter-Gather Descriptor for the linked list must contain the following information:
   a. Memory Address (buffer start) 4 bytes
   b. Byte Count (buffer size) 3 bytes
   c. End of Link List 1 bit (MSB)

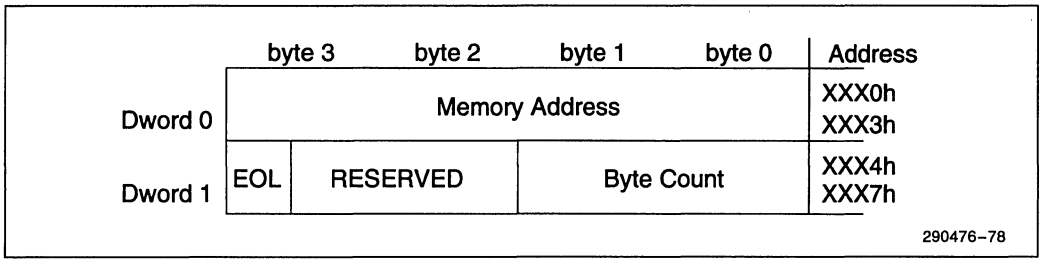


Figure 15. Scatter-Gather Descriptor Format

3. Initialize DMA Mode and Extended Mode registers with transfer specific information like 8-/16-bit I/O device, Transfer Mode, Transfer Type, etc.
4. Software provides the starting address of the Scatter-Gather Descriptor Table by loading the Scatter-Gather Descriptor Table Pointer register.
5. Engage the Scatter-Gather machine by writing a Start command to the Scatter-Gather Command register.
6. The Mask register should be cleared as last the last step of programming the DMA register set. This is to prevent DMA from starting a transfer with a partially loaded command description.
7. Once the register set is loaded and the channel is unmasked, the DMA will generate an internal request to fetch the first buffer from the Scatter Gather Descriptor Table.
8. The DMA will then respond to DREQ or software requests. The first transfer from the first buffer will move the memory address and word count from the Base register set to the Current register set. As long as Scatter-Gather is active and the Base register set is not loaded and the last buffer has not been fetched, the channel will generate a request to fetch a reserve buffer into the Base register set. The reserve buffer is loaded to minimize latency problems going from one buffer to another. Fetching a reserve buffer has a lower priority than completing DMA for the channel.
9. The DMA controller will terminate a Scatter-Gather cycle by detecting an End of List (EOL) bit in the SGD. After the EOL bit is detected, the channel will transfer the buffers in the Base and Current register sets if they are loaded. At Terminal Count the channel will assert EOP or IRQ13 depending on its programming and set the Terminate bit in the Scatter-Gather Status register. The Active bit in the Scatter-Gather Status register will be reset and the channel's Mask bit will be set.

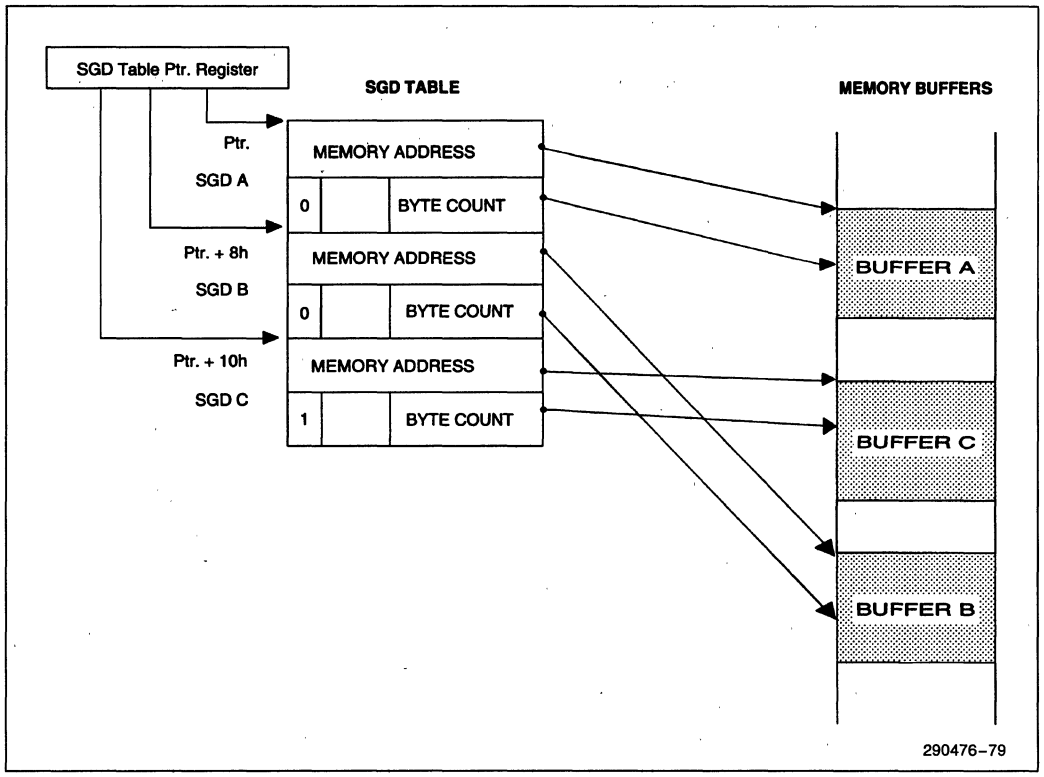


Figure 16. Link List Example

6.7 Register Functionality

See Section 3.2 for detailed information on register programming, bit definitions, and default values/functions after a reset.

DMA Channel 4 is used to cascade the two DMA controllers together and should not be programmed for any mode other than cascade. The Mode register for Channel 4 will default to cascade mode. Special attention should also be take when programming the Command and Mask registers as related to Channel 4 (refer to the Command and Mask register descriptions, Section 3.2).

6.7.1 ADDRESS COMPATIBILITY MODE

Whenever the DMA is operating in Address Compatibility mode, the addresses do not increment or decrement through the High and Low Page registers, and the high page register is set to 00h. This is compatible with the 82C37 and Low Page register implementation used in the PC AT. This mode is set when any of the lower three address bytes of a channel are programmed. If the upper byte of a channel's address is programmed last, the channel will go into Extended Address Mode. In this mode, the high byte may be any value and the address will increment or decrement through the entire 32-bit address.

After reset is negated all channels will be set to Address Compatibility Mode. The DMA Master Clear command will also reset the proper channels to Address Compatibility Mode. The Address Compatibility Mode bits are stored on a per channel basis.

6.7.2 SUMMARY OF THE DMA TRANSFER SIZES

Table 16 lists each of the DMA device transfer sizes. The column labeled "Word Count Register" indicates that the register contents represent either the number of bytes to transfer or the number of 16-bit words to transfer. The column labeled "Current Address Register Increment/Decrement" indicates the number added to or taken from the Current Address register after each DMA transfer cycle. The Mode Register determines if the Current Address register will be incremented or decremented.

Table 16. DMA Transfer Size

DMA Device Date Size And Word Count	Word Count Register	Current Address Increment/Decrement
8-bit I/O, Count By Bytes	Bytes	1
16-bit I/O, Count By Words (Address Shifted)	Words	1
16-bit I/O, Count By Bytes	Bytes	2
32-bit I/O, Count By Bytes	Bytes	4

6.7.3 ADDRESS SHIFTING WHEN PROGRAMMED FOR 16-BIT I/O COUNT BY WORDS

To maintain compatibility with the implementation of the DMA in the PC/AT which used the 82C37, the DMA will shift the addresses when the Extended Mode register is programmed for, or defaulted to, transfers to/from a 16-bit device count6.7.3-by-words. Note that the least significant bit of the Low Page register is dropped in 16-bit shifted mode. When programming the Current Address register while the DMA channel is in this mode, the Current Address must be programmed to an even address with the address value shifted right by one bit. The address shifting is as shown in Table 17.

Table 17. Address Shifting in 16-Bit I/O DMA Transfers

Output Address	8-Bit I/O Programmed Address	16-Bit I/O Programmed Address (Shifted)	16-Bit I/O Programmed Address (No Shift)	32-Bit I/O Programmed Address (No Shift)
A0 [16:1] A[31:17]	A0 A[16:1] A[31:17]	"0" A[15:0] A[31:17]	A0 A[16:01] A[31:17]	A0 A[16:01] A[31:17]

NOTE:
The least significant bit of the Low Register is dropped in 16-bit shifted mode.

6.7.4 STOP REGISTERS (RING BUFFER DATA STRUCTURE)

To support a common data communication data structure, (the ring buffer), a set of DMA registers have been provided. These registers are called Stop registers. Each channel has 22-bits of register location associated with it. The 22-bits are distributed between three different registers (one 8-bit and two 8-bit). The Stop registers can be enabled or disabled by writing to the channel's corresponding Extended Mode register.

The ring buffer data structure reserves a fixed portion of memory, on Dword boundaries, to be used for a DMA channel. Consecutively received frames or other data structures are stored sequentially within the boundaries of the ring buffer memory.

The beginning and end of the ring buffer area is defined in the Base Address register and the Base Address register + the Base Byte/Transfer Count. The incoming frames (data) are deposited in sequential locations of the ring buffer. When the DMA reaches the end of the ring buffer, indicating the byte count has expired, the DMA controller (if so programmed) will autoinitialize. Upon autoinitialization, the Current Address register will be restored from the Base Address register, taking the process back to the start of the ring buffer. The DMA will then be available to begin depositing the incoming bytes in the ring buffers sequential locations, providing that the CPU has read the data that was previously placed in those locations. The DMA determines that the CPU has read certain data by the value that the CPU writes into the Stop register.

Once the data of a frame is read by the CPU, the memory location it occupies becomes available for other incoming frames. The Stop register prevents the DMA from over writing data that has not yet been read by the CPU. After the CPU has read a frame from memory it will update the Stop register to point to the location that was last read. The DMA will not deposit data into any location beyond that pointed to by the Stop register. The last address transferred before the channel is masked is the first address that matches the Stop register.

For example, if the stop register = 00001Ch, the last three transfers are shown in Table 18.

Table 18. Stop Register Functionality Example

	By Bytes	By Words	By Words
Increment	XX00001Ah	XX000018h	XX000018h
	XX00001Bh	XX00001Ah	XX00001Ah
Decrement	XX00001Ch XX000021h	XX00001Ch XX000023h	XX00001Ch XX000023h
	XX000020h	XX000021h	XX000021h
	XX00001Fh	XX00001Fh	XX00001Fh

NOTE:
The Stop registers store values to compare against LA[23:2] only, so the size of the ring buffer is limited to 16 MBytes.

6.7.5 BUFFER CHAINING MODE AND STATUS REGISTERS

The Chaining Mode registers are used to implement the buffer chaining mode of a channel. The buffer chaining mode is useful when transferring data from a peripheral to several different areas of memory with one continuous transfer operation. Four registers are used to implement this function: the Chaining Mode register, the Chaining Mode Status Register, the Channel Interrupt Status register, and the Chain Buffer Expiration Control register.

The Chaining Mode register controls the buffer chaining initialization. Buffer chaining mode can be enabled or disabled. A Chaining Mode bit is used to indicate if Base register programming is complete and chaining can begin, or to hold off chaining because the Base registers still need programming. Another bit dictates the buffer expiration response by indicating whether an IRQ13 or EOP should be issued when the buffer needs reprogramming. The Chaining Mode Status Register indicates whether each channel's chaining mode is enabled or disabled.

The Channel Interrupt Status Register indicates the channel source of a DMA chaining interrupt on IRQ13. The CPU can read this register to determine which channel asserted IRQ13 following a buffer expiration. The Chain Buffer Expiration Control Register is a read only register that reflects the outcome after the expiration of a chain buffer. If a channel bit is set to 0, IRQ13 will be activated following the buffer expiration. If a channel bit is set to 1, EOP will be asserted following the buffer expiration.

6.7.6 AUTOINITIALIZE

By programming a bit in the Mode register, a channel may be set up as an autoinitialize channel. During Autoinitialization, the original values of the Current page, Current address and Current Byte/Word Count registers are automatically restored from the Base Address, and Word count registers of that channel following TC. The Base registers are loaded simultaneously with the Current registers by the microprocessor and remain unchanged throughout the DMA service. The mask bit is not set when the channel is in autoinitialize. Following autoinitialize the channel is ready to perform another DMA service, without CPU intervention, as soon as a valid DREQ is detected. (Note: Autoinitialize will not function if the channel is also programmed for Scatter-Gather or buffer chaining. Only one of these features should be enabled at a time.)

6.8 Software Commands

These are additional special software commands which can be executed in the Program Condition. They do not depend on any specific bit pattern on the data bus. The three software commands are:

1. Clear Byte Pointer Flip-Flop.
2. Master Clear.
3. Clear Mask Register.

6.8.1 CLEAR BYTE POINTER FLIP-FLOP

This command is executed prior to writing or reading new address or word count information to the DMA. This initializes the flip-flop to a known state so that subsequent accesses to register contents by the microprocessor will address upper and lower bytes in the correct sequence.

When the CPU is reading or writing DMA registers, two Byte Pointer Flip-Flops are used; one for Channels 0-3 and one for Channels 4-6. Both of these act independently. There are separate software commands for clearing each of them (0Ch for Channels 0-3, 0D8h for Channels 4-7).

An additional Byte Pointer Flip-Flop has been added for use when EISA masters are reading and writing DMA registers. (The arbiter state will be used to determine the current master of the bus.) This Flip-Flop is cleared when an EISA Master performs a write to either 00Ch or 0D8h. there is one Byte Pointer Flip Flop per eight DMA channels. This Byte Pointer was added to eliminate the problem of the CPU's byte pointer getting out of synchronization if an EISA Master takes the bus during the CPU's DMA programming.

6.8.2 DMA MASTER CLEAR

This software instruction has the same effect as the hardware Reset. The Command, Status, Request, and Internal First/Last Flip-Flop registers are cleared and the Mask register is set. The DMA Controller will enter the idle cycle.

There are two independent Master Clear Commands, 0Dh which acts on Channels 0-3, and 0DAh which acts on Channels 4-6.

6.8.3 CLEAR MASK REGISTER

This command clears the mask bits of all four channels, enabling them to accept DMA requests. I/O port 00Eh is used for Channels 0-3 and I/O port 0DCh is used for Channels 4-6.

6.9 Terminal Count/EOP Summary

Table 19 is a summary of the events that happen as a result of a terminal count or external EOP when running DMA in various modes.

Table 19. Terminal Count/EOP Summary Table

Conditions					
	AUTOINIT	No		Yes	
Event					
	Word Counter Expired	Yes	X	Yes	X
	EOP Input	X	Asserted	X	Asserted
Result					
	Status TC	set	set	set	set
	Mask	set	set	—	—
	SW Request	clr	clr	clr	clr
	Current Register	—	—	load	load

NOTES:
1. load = Load Current From Base
2. "-" = No Change
3. X = Don't Care
4. clr = Clear

6.10 Buffer Chaining

The buffer chaining mode of a channel is useful for transferring data from a peripheral to several different areas of memory within one transfer operation (from the DMA device's viewpoint). This is accomplished by causing the DMA to interrupt the CPU for more programming information while the previously programmed transfer is still in progress. Upon completion of the previous transfer, the DMA controller will then load the new transfer information automatically. In this way, the entire transfer can be completed without interrupting the operation of the DMA device. This mode is most useful for DMA single-cycle or demand modes where the transfer process allows time for the CPU to execute the interrupt routine.

The buffer chaining mode of a channel may be entered by programming the address and count of a transfer as usual. After the initial address and count is programmed, the Base registers are selected via the Chaining Mode register Chaining Mode Enabled bit. The address and count for the second transfer and both the Chaining Mode Enabled and the Program Complete bit of the Chaining Mode register should be programmed at this point, before starting the DMA process. When, during the DMA process, the Current Buffer is expired, the Base address, Page, and Count registers will be transferred to the Current registers and a signal that the buffer has been expired is sent to the programming master.

This signal will be an IRQ13 if the master is the CPU, or a TC if the programming master is an EISA Master device. The type of programming master is indicated in the DMA's Chaining Mode Register, bit 4. If the CPU is the programming master for the Channel, TC will be generated only if the Current buffer expires and there is no Next Buffer stored in the Base registers.

Upon the expiration of a Current Buffer, the new Base register contents should be programmed and both the Chaining Mode Enabled and Program complete bits of the Chaining Mode register should be set. This resets the interrupt, if the CPU was the programming master, and allows for the next Base register to Current register transfer. If the Program Complete bit is not set before the current transfer reaches TC, then the DMA controller will set the Mask bit and the TC bit in the Status register and stop transferring data. In this case, an over-run is likely to occur. To determine if this has, a read of either Status register or the Mask register can be done (the Mask register has been made readable). If the channel is masked or has registered a TC, the DMA channel has been stopped and the full address, count, and chaining mode must be programmed to return to normal operation.

Note that if the CPU is the programming master, an interrupt will only be generated if a Current Buffer expires and chaining mode is enabled. It will not occur during initial programming. The Channel Interrupt Status register will indicate pending interrupts only. That is, it will indicate an empty Base register with Chaining Mode enabled. When Chaining mode is enabled, only the Base registers are written by the processor, and only the Current registers can be read. The Current registers are only updated on a TC.

6.11 Refresh Unit

The ESC provides an EISA Bus compatible refresh unit that provides 14 bits of refresh address for EISA/ISA bus DRAMs that do not have their own local refresh units. The refresh system uses the combined functions of the Interval Timers, the DMA Arbiter, DMA address counter, and EISA Bus Controller. Functionally, the Refresh unit is a sub-section of the ESC DMA unit. The DMA Address Counter is used to increment the Refresh Address register following each refresh cycle. Interval Counter 1, Timer 1 generates an internal refresh request. The DMA Arbiter detects a Refresh signal from either the Counter/Timer or the REFRESH# input and determines when the refresh will be done. The DMA drives the refresh address out onto the LA address bus. The cycle is decoded and driven onto the EISA address bus by the EISA Bus Controller. The ESC EISA Bus Controller is responsible for generating the EISA cycle control signals. Timer 1 Counter 1 should be programmed to provide a refresh request about every 15 μs.

Requests for refresh cycles are generated by two sources: the ESC (Timer 1 Counter 1), and 16-bit masters that activate REFRESH# when they own the EISA bus.

If a 16-bit ISA bus master holds the bus longer than 15 μs, it must initiate memory refresh cycles. If the ISA Master initiates a Refresh cycle while it owns the bus, it floats the address lines and cycle control signals and asserts REFRESH# to the ESC. The ESC EISA Bus Controller generates the cycle control signals and the ESC DMA Refresh unit supplies the refresh address. The ISA Master must then wait one BCLK after MRDC# is negated before floating REFRESH# and driving the address lines and control signals.

Typically, the refresh cycle length is five BCLK's. The I/O slave can insert one wait state to extend the cycle to six BCLK's by asserting CHRDY. The ESC EISA Bus Controller, upon seeing REFRESH#, knows to run refresh cycles instead of DMA cycles.

7.0 EISA BUS ARBITRATION

The ESC receives requests for EISA Bus ownership from several different sources; from DMA devices, from the Refresh counter, from EISA masters and from PCI agents. PCI agents requesting the EISA Bus request the EISA Bus through the PCEB. Additionally, 16-bit ISA Masters may request the bus through a cascaded DMA channel (see the Cascade mode description in Section 6.2.4).

7.1 Arbitration Priority

At the top level of the arbiter, the ESC uses a three way rotating priority arbitration method. On a fully loaded bus, the order in which the devices are granted bus access is independent of the order in which they assert a bus request, since devices are serviced based on their position in the rotation. The arbitration scheme assures that DMA channels and EISA masters are able to access the bus with minimal latency.

The PCEB and EISA Masters share one of the slots in the three way rotating priority scheme. This sharing is a two way rotation between the CPU and EISA Masters as a group. In this arbitration scheme, the PCEB acts on behalf of the CPU and all other PCI masters.

EISA Masters have a rotating priority structure which can handle up to eight master requests.

The next position in the top level arbiter is occupied by the DMA. The DMA's DREQ lines can be placed in either fixed or rotating priority. The default mode is fixed and by programming the DMA Command registers, the priority can be modified to rotating priority mode.

7.2 Preemption

An EISA compatible arbiter ensures that minimum latencies are observed for both EISA DMA devices, and EISA Masters.

7.2.1 PCEB EISA BUS ACQUISITION AND PCEB PREEMPTION

EISA bus arbitration is intended to be optimized for CPU access the EISA bus. Since the CPU accesses to the EISA Bus through the PCEB, the PCEB is assumed to be the default owner of the EISA bus. The arbitration interface between the PCEB and the ESC is implemented as a HOLD/HLDA (EISAHOLD/ EISAHLDA) pair.

If a PCI cycle requires access to the EISA Bus while EISAHLDA signal is asserted (EISA Bus busy) the PCI cycle is retried, and the PCEB requests the EISA bus by asserting PEREQ#. The ESC, after sampling PEREQ# asserted, preempts the current owner of the EISA Bus. The ESC grants the EISA Bus by negating EISAHOLD signal.

The ESC asserts EISAHOLD to the PCEB when the ESC needs to acquire the ownership of the EISA bus. While EISAHOLD is asserted, the arbitration process is dynamic and may change (i.e. the ESC is still accepting EISA Bus requests). When the PCEB returns EISAHLDA, the arbiter freezes the arbitration process and determine the winner. If the new winner is an EISA Master or DMA channel, the ESC will assert NMFLUSH#. The ESC tri-states the NMFLUSH# output driver on the following clock. The PCEB holds NMFLUSH# asserted until all buffers are flushed. After all buffers are flushed, the PCEB negates NMFLUSH# and then tri-state the output buffer. After sampling NMFLUSH# negated, the ESC resumes driving NMFLUSH# on the next PCI clock. This way the ESC does not assert MACK# or DACK# until the PCEB acknowledges that all line buffers have been flushed.

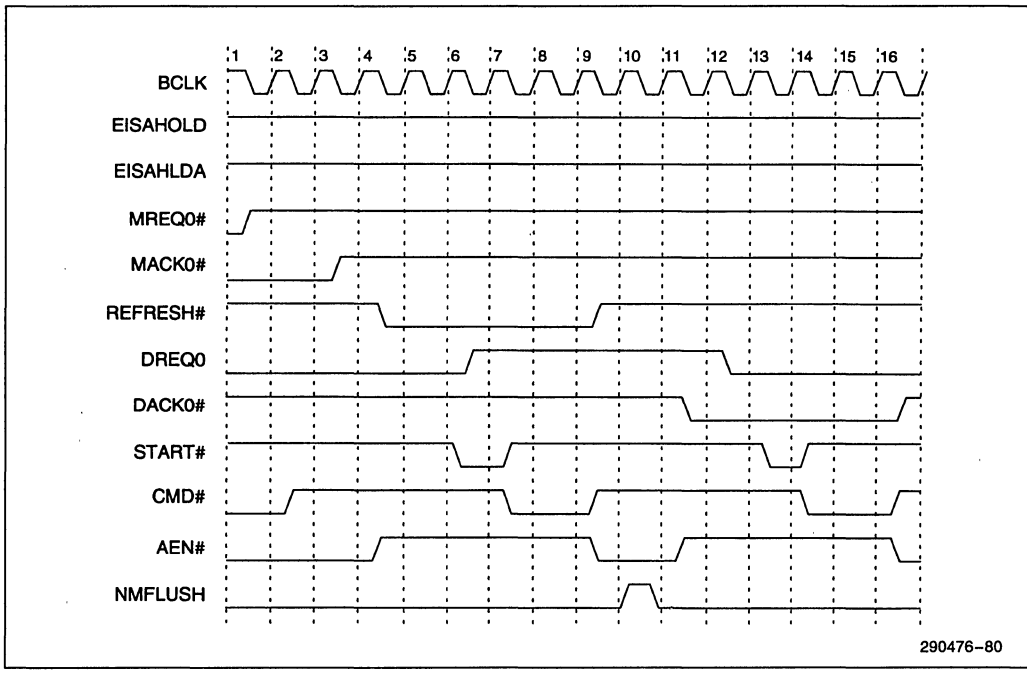


Figure 17. EISA Arbitration

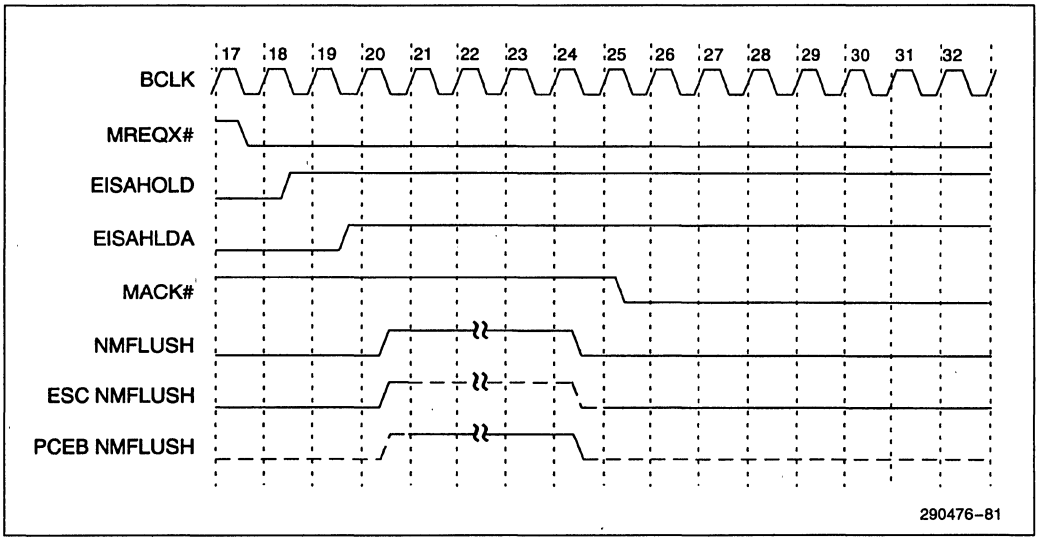


Figure 18. PCEB Preemption

7.2.2 EISA MASTER PREEMPTION

EISA specification requires that EISA Masters must release the bus within 64 BCLKs (8 μs) after the ESC negates MACKx#. If the bus master attempts to start a new bus cycle after this timeout period, a bus timeout (NMI) is generated and the RSTDRV is asserted to reset the offending bus master.

7.2.3 DMA PREEMPTION

A DMA slave device that is not programmed for compatible timing is preempted from the EISA Bus by another device that requests use of the bus. This will occur regardless of the priority of the pending request. For DMA devices not using compatible timing mode, the DMA controller stops the DMA transfer and releases the bus within 32 BCLK (4 μs) of a preemption request. Upon the expiration of the 4 μs timer, the DACK is negated after the current DMA cycle has completed. The EISA Bus then arbitrated for and granted to the highest priority requester. This feature allows flexibility in programming the DMA for long transfer sequences in a performance timing mode while guaranteeing that vital system services such as Refresh are allowed access to the expansion bus.

The 4 μs timer is not used in compatible timing mode. It is only used for DMA channels programmed for Type "A", Type "B", or Type "C" (Burst) timing. The 4 μs timer is also not used for 16-bit ISA masters cascaded through the DMA DREQ lines.

If the DMA channel that was preempted by the 4 μs timer is operating in Block mode, an internal bit will be set so that the channel will be arbitrated for again, independent of the state of DREQ.

7.3 Slave Timeouts

A slave which does not release EXRDY or CHRDY can cause the CMD# active time to exceed 256 BCLKs (32 μs). The ESC does not monitor EXRDY or CHRDY for this timeout. Typically this function is provided in a system through a third party add-in card. The add-in cards which monitor EXRDY or CHRDY assert IOCHK signal when the 256 BCLK count expires. The ESC in response asserts NMI.

The only way that a 16-bit ISA Master can be preempted from the EISA bus is if it exceeds the 256 BCLK

(32 μs) limit on CMD# active.

7.4 Arbitration During Non-Maskable Interrupts

If a non-maskable interrupt (NMI) is pending at the PCEB, and the PCEB is requesting the bus, the DMA and EISA Masters will be bypassed each time they come up for rotation. This gives the PCEB the EISA Bus bandwidth on behalf of the CPU to process the interrupt as fast as possible.

8.0 INTERVAL TIMERS

The ESC contains five counter/timers that are equivalent to those found in the 82C54 programmable interval timer. The five counters are contained in two separate ESC timer units, referred to as Timer-1 and Timer-2. The ESC uses the Timers to implement key EISA system functions. Timer-1 contains three counters, and Timer-2 contains two counters. EISA systems do not use the middle counter on Timer-2.

Interval Timer 1, Counter 0 is connected to the interrupt controller IRQ0 and provides a system timer interrupt for a time-of-day, diskette time-out, or other system timing functions. Counter 1 generates a refresh-request signal and Counter 2 generates the tone for the speaker.

Interval Timer 2, Counter 0 implements a fail safe timer. Counter 0 generates NMI at regular intervals, thus preventing the system from locking up. Counter 1 is not used. Counter 2 is used to slow down the CPU by means of pulse-width modulation. The output of Timer 2 Counter 2 is tied to the SLOWH# signal.

Table 20. Interval Timer Functions

Function	Counter 0 System Timer	Counter 0 Fail-Safe Timer
Gate	Always On	Always On
Clock In	1.193 MHz(OSC/12)	0.298 MHz(OSC/48)
Out	INT-1 IRQ0	NMI Interrupt
	Counter 1 Refresh Request	
Gate	Always On	
Clock In	1.193 MHz(0SC/12)	
Out	Refresh Request	
	Counter 2	**Counter 2**
Gate	Programmable	Refresh Request
Clock In	Port 61h	
Out	1.193 MHz(OSC/12)	8 MHz (BCLK)
	Speaker	CPU Speed Control (SLOWH#)

8.1 Interval Timer Address Map

Table 21 shows the I/O address map of the interval timer counters:

Table 21. Interval Timer I/O Address Map

I/O Port Address	Register Description
040h	Timer 1, System Timer (Counter 0)
041h	Timer 1, Refresh Request (Counter 1)
042h	Timer 1, Speaker Tone (Counter 2)
043h	Timer 1, Control Word Register
048h	Timer 2, Fail-Safe Timer (Counter 0)
049h	Timer 2, Reserved
04Ah	Timer 2, CPU Speed Control (Counter 2)
04Bh	Timer 2, Control Word Register

Timer 1—Counter 0, System Timer

This counter functions as the system timer by controlling the state of IRQ[0] and is typically programmed for Mode 3 operation. The counter produces a square wave with a period equal to the product of the counter period (838 ns) and the initial count value. The counter loads the initial count value one counter period after software writes the count value to the counter I/O address. The counter initially asserts IRQ[0] and decrements the count value by two each counter period. The counter negates IRQ[0] when the count value reaches 0. It then reloads the initial count value and again decrements the initial count value by two each counter period. The counter then asserts IRQ[0] when the count value reaches "0", reloads the initial count value, and repeats the cycle, alternately asserting and negating IRQ[0].

Timer 1—Counter 1, Refresh Request Signal

This counter provides the Refresh Request signal and is typically programmed for Mode 2 operation. The counter negates Refresh Request for one counter period (833 ns) during each count cycle. The initial count value is loaded one counter period after being written to the counter I/O address. The counter initially asserts Refresh Request, and negates it for 1 counter period when the count value reaches 1. The counter then asserts Refresh Request and continues counting from the initial count value.

Timer 1—Counter 2, Speaker Tone

This counter provides the speaker tone and is typically programmed for Mode 3 operation. The counter provides a speaker frequency equal to the counter clock frequency (1.193 MHz) divided by the initial count value. The speaker must be enabled by a write to port 061h (see Section 3.7 on the NMI Status and Control Ports).

Timer 2—Counter 0, Fail-Safe Timer

This counter functions as a fail-save timer by preventing the system from locking up. This counter generates an interrupt on the NMI line as the count expires by setting bit 7 on Port 0461h. Software routines can avoid the Fail-Safe NMI by resetting the counter before the timer count expires.

Timer 2—Counter 2, CPU Speed Control

This counter generates the SLOWH# to the CPU and is typically programmed for Mode 1 operation. The counter is triggered by the refresh request signal generated by Timer 1-Counter 1 only. If the counter is programmed, the counters SLOWH# output will stop the CPU for the programmed period of the one-shot every time a refresh request occurs. This counter is not configured or programmed until a speed reduction in the system is required.

8.2 Programming The Interval Timer

The counter/timers are programmed by I/O accesses and are addressed as though they are contained in two separate 82C54 interval timers. Timer 1 contains three counters and Timer 2 contains two counters. Each Timer is controlled by a separate Control Word register. Table 22 lists the six operating modes for the interval counters. Note that for the fail safe timer (timer 2, counter 0), only mode 0 is supported.

The interval timer is an I/O-mapped device. Several commands are available:

1. The Control Word Command specifies:
   — which counter to read or write
   — the operating mode
   — the count format (binary or BCD)
2. The Counter Latch Command latches the current count so that it can be read by the system. The count-down process continues.
3. The Read Back Command reads the count value, programmed mode, the current state of the OUT pins, and the state of the Null Count Flag of the selected counter.

The Read/Write Logic selects the Control Word register during an I/O write when address lines A1, A0 = 11. This condition occurs during an I/O write to port addresses 043h and 04Bh, the addresses for the Control Word Register on Timer 1 and Control Word Register on Timer 2 respectively. If the CPU writes to port 043h or port 04Bh, the data is stored in the respective Control Word Register and is interpreted as a Control Word used to define the operation of the Counters.

The Control Word Register is write-only. Counter Status information is available with the Read-Back Command.

Table 22. Counter Operating Modes

Mode	Function
0	Out signal on end of count (= 0)
1	Hardware retriggerable one-shot
2	Rate generator (divide by n counter)
3	Square wave output
4	Software triggered strobe
5	Hardware triggered strobe

Because the timer counters come up in an unknown state after power up, multiple refresh requests may be queued up. To avoid possible multiple refresh cycles after power up, program the timer counter immediately after power up.

Write Operations

Programming the interval timer is a simple process:

1. Write a control word.
2. Write an initial count for each counter.
3. Load the least and/or most significant bytes (as required by Control Word Bits 5, 4) of the 16-bit counter.

The programming procedure for the ESC timer units is very flexible. Only two conventions need to be observed. First, for each Counter, the Control Word must be written before the initial count is written. Second, the initial count must follow the count format specified in the Control Word (least significant byte only, most significant byte only, or least significant byte and then most significant byte).

Since the Control Word Register and the three Counters have separate addresses (selected by the A1, A0 inputs), and each Control Word specifies the Counter it applies to (SC0, SC1 bits), no special instruction sequence is required. Any programming sequence that follows the conventions above is acceptable.

A new initial count may be written to a Counter at any time without affecting the Counter's programmed Mode in any way. Counting will be effected as described in the Mode definitions. The new count must follow the programmed count format.

If a Counter is programmed to read/write two-byte counts, the following precaution applies: A program must not transfer control between writing the first and second byte to another routine which also writes into that same Counter. Otherwise, the Counter will be loaded with an incorrect count.

Interval Timer Control Word Format

The Control Word specifies the counter, the operating mode, the order and size of the COUNT value, and whether it counts down in a 16-bit or binary-coded decimal (BCD) format. After writing the control word, a new count may be written at any time. The new value will take effect according to the programmed mode.

If a counter is programmed to read/write two-byte counts, the following precaution applies: A program must not transfer control between writing the first and second byte to another routine which also writes into that same counter. Otherwise, the counter will be loaded with an incorrect count. The count must always be completely loaded with both bytes.

Read Operations

It is often desirable to read the value of a Counter without disturbing the count in progress. This is easily done in the ESC timer units.

There are three possible methods for reading the counters: a simple read operation, the Counter Latch Command, and the Read-Back Command.

Counter I/O Port Read

The first method is to perform a simple read operation. To read the Counter the CLK input of the selected Counter must be inhibited by using either the GATE input or external logic. Otherwise, the count may be in the process of changing when it is read, giving an undefined result. When reading the count value directly, follow the format programmed in the control register: read LSB, read MSB, or read LSB then MSB. Within the ESC timer unit, the GATE input on Timer 1 Counter 0, Counter 1 and Timer 2 Counter 0 are tied high. Therefore, the direct register read should not be used on these two counters. The GATE input of Timer 1 Counter 2 is controlled through I/O port 061h. If the GATE is disabled through this register, direct I/O reads of port 042h will return the current count value.

Counter Latch Command

The Counter Latch command latches the count at the time the command is received. This command is used to insure that the count read from the counter is accurate (particularly when reading a two-byte count). The count value is then read from each counter's Count register as was programmed by the Control register.

The selected Counter's output latch (OL) latches the count at the time the Counter Latch Command is received. This count is held in the latch until it is read by the CPU (or until the Counter is reprogrammed). The count is then unlatched automatically and the OL returns to "following" the counting element (CE). This allows reading the contents of the Counters "on the fly" without effecting counting in progress. Multiple Counter Latch Commands may be used to latch more than one Counter. Each latched Counter's OL holds its count until it is read. Counter Latch Commands do not effect the programmed Mode of the Counter in any way. The Counter Latch Command can be used for each counter in the ESC timer unit.

If a Counter is latched and then, some time later, latched again before the count is read, the second Counter Latch Command is ignored. The count read will be the count at the time the first Counter Latch Command was issued.

With either method, the count must be read according to the programmed format; specifically, if the Counter is programmed for two byte counts, two bytes must be read. The two bytes do not have to be read one right after the other; read, write, or programming operations for other Counters may be inserted between them.

Another feature of the ESC timer unit is that reads and writes of the same Counter may be interleaved. For example, if the Counter is programmed for two byte counts, the following sequence is valid:

1. Read least significant byte.
2. Write new least significant byte.
3. Read most significant byte.
4. Write new most significant byte.

One precaution is worth noting. If a Counter is programmed to read/write two-byte counts, a program must not transfer control between reading the first and second byte to another routine which also reads from that same Counter. Otherwise, an incorrect count will be read.

Read Back Command

The third method uses the Read-Back command. The Read-Back command is used to determine the count value, programmed mode, and current states of the OUT pin and Null Count flag of the selected counter or counters. The Read-Back command is written to the Control Word register, which causes the current states of the above mentioned variables to be latched. The value of the counter and its status may then be read by I/O access to the counter address.

The read-back command may be used to latch multiple counter output latches (OL) by setting the COUNT# bit D5=0 and selecting the desired counter(s). This single command is functionally equivalent to several counter latch commands, one for each counter latched. Each counter's latched count is held until it is read (or the counter is reprogrammed). Once read, a counter is automatically unlatched. The other counters remain latched until they are read. If multiple count read-back commands are issued to the same counter without reading the count, all but the first are ignored; i.e. the count which will be read is the count at the time the first read-back command was issued.

The read-back command may also be used to latch status information of selected counter(s) by setting STATUS# bit D4=0. Status must be latched to be read. The status of a counter is accessed by a read from that counter's I/O port address.

If multiple counter status latch operations are performed without reading the status, all but the first are ignored. The status returned from the read is the counter status at the time the first status read-back command was issued.

Both count and status of the selected counter(s) may be latched simultaneously by setting both the COUNT# and STATUS# bits[5:4]=00b. This is functionally the same as issuing two consecutive, separate read-back commands. The above discussions apply here also. Specifically, if multiple count and/or status read-back commands are issued to the same counter(s) without any intervening reads, all but the first are ignored.

If both count and status of a counter are latched, the first read operation from that counter will return the latched status, regardless of which was latched first. The next one or two reads (depending on whether the counter is programmed for one or two type counts) return the latched count. Subsequent reads return unlatched count.

9.0 INTERRUPT CONTROLLER

The ESC provides an EISA compatible interrupt controller which incorporates the functionality of two 82C59 interrupt controllers. The two controllers are cascaded so that 14 external and two internal interrupts are possible. The master interrupt controller provides IRQ[7:0] and the slave interrupt controller provides IRQ[15:8] (Figure 19). The two internal interrupts are used for internal functions only and are not available at the chip periphery. IRQ2 is used to cascade the two controllers together and IRQ0 is used as a system timer interrupt and is tied to Interval Timer 1, Counter 0. The remaining 14 interrupt lines (IRQ1, IRQ3-IRQ15) are available for external system interrupts. Edge or level sense selection is programmable on a by-controller basis.

The Interrupt Controller consists of two separate 82C59 cores. Interrupt Controller 1 (CNTRL-1) and Interrupt Controller 2 (CNTRL-2) are initialized separately, and can be programmed to operate in different modes. The default settings are: 80x86 Mode, Edge Sensitive (IRQ0-15) Detection, Normal EOI, Non-Buffered Mode, Special Fully Nested Mode disabled, and Cascade Mode. CNTRL-1 is connected as the Master Interrupt Controller and CNTRL-2 is connected as the Slave Interrupt Controller.

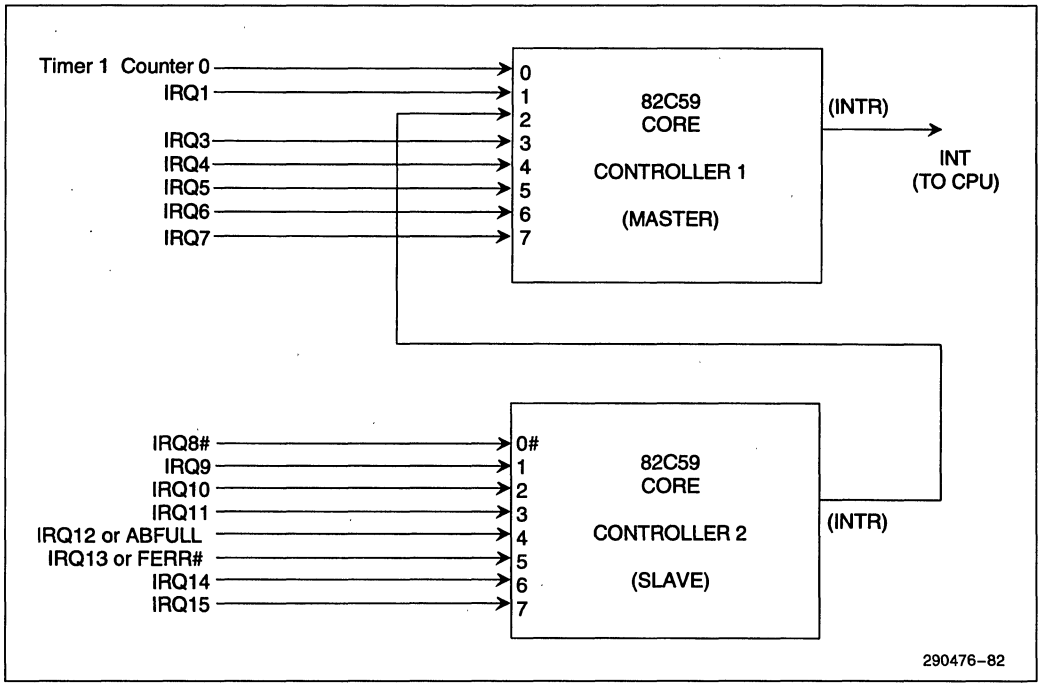


Figure 19. Block Diagram of the Interrupt Controller

Table 23 lists the I/O port address map for the interrupt registers.

Table 23. I/O Address Map

Interrupts	I/O Address	# of bits	Register
IRQ[7:0]	0020h	8	CNTRL-1 Control Register
IRQ[7:0]	0021h	8	CNTRL-1 Mask Register
IRQ[7:0]	04D0h	8	CNTRL-1 Edge/Level Control Register
IRQ[15:8]	00A0h	8	CNTRL-2 Control Register
IRQ[15:8]	00A1h	8	CNTRL-2 Mask Register
IRQ[15:8]	04D1h	8	CNTRL-2 Edge/Level Control Register

IRQ0, and IRQ2 are connected to the interrupt controllers internally. The other interrupts are always generated externally. IRQ12 and IRQ13 may be generated internally through the ABFULL and FERR# signals, respectively.

Table 24. Typical Interrupt Functions

Priority	Label	Controller	Typical Interrupt Source
1	IRQ0	1	Interval Timer 1, Counter 0 OUT
2	IRQ1	1	Keyboard
3-10	IRQ2	1	Interrupt from Controller 2
3	IRQ8#	2	Real Time Clock
4	IRQ9	2	Expansion Bus Pin B04
5	IRQ10	2	Expansion Bus Pin D03
6	IRQ11	2	Expansion Bus Pin D04
7	IRQ12	2	Expansion Bus Pin D05
8	IRQ13	2	Coprocessor Error, Chaining
9	IRQ14	2	Fixed Disk Drive Controller Expansion Bus Pin D07
10	IRQ15	2	Expansion Bus Pin D06
11	IRQ3	1	Serial Port 2, Expansion Bus B25
12	IRQ4	1	Serial Port 1, Expansion Bus B24
13	IRQ5	1	Parallel Port 2, Expansion Bus B23
14	IRQ6	1	Diskette Controller, Expansion Bus B22
15	IRQ7	1	Parallel Port 1, Expansion Bus B21

9.1 Interrupt Controller Internal Registers

Several registers are contained internally within each 82C59. The interrupts at the IRQ input lines are handled by two registers in cascade, the Interrupt Request Register (IRR) and the In-Service Register (ISR). The IRR is used to store all the interrupt levels which are requesting service and the ISR is used to store all the interrupt levels which are being serviced.

Internal circuitry determines the priorities of the bits set in the IRR. The highest priority is selected and strobed into the corresponding bit of the ISR during Interrupt Acknowledge Cycles.

The Interrupt Mask Register (IMR) stores the bits which mask the incoming interrupt lines. The IMR operates on the IRR. Masking of a higher priority input will not effect the interrupt request lines of lower priority inputs.

9.2 Interrupt Sequence

The powerful features of the Interrupt Controller in a microcomputer system are its programmability and the interrupt routine addressing capability. The latter allows direct or indirect jumping to the specific interrupt routine requested without any polling of the interrupting devices. The following shows the interrupt sequence for an x86 type system (the 8080 mode of the interrupt controller must never be selected when programming the ESC).

Note that externally, the interrupt acknowledge cycle sequence appears different than in a traditional discrete 82C59 implementation. However, the traditional interrupt acknowledge sequence is generated within the ESC and it is an EISA compatible implementation.

1. One or more of the Interrupt Request (IRQ[x]) lines are raised high, setting the corresponding IRR bit(s).
2. The Interrupt Controller evaluates these requests, and sends an INTR to the CPU, if appropriate.
3. The CPU acknowledges the INTR and responds with an interrupt acknowledge cycle. This cycle is translated into a PCI bus command. This PCI command is broadcast over the PCI bus as a single cycle as opposed to the two cycle method typically used.
4. Upon receiving an interrupt acknowledge cycle from the CPU over the PCI, the PCEB converts the single cycle into an INTA# pulse to the ESC. The ESC uses the INTA# pulse to generate the two cycles that the internal 8259 pair can respond to with the expected interrupt vector. The cycle conversion is performed by a functional block in the ESC Interrupt Controller Unit. The internally generated interrupt acknowledge cycle is completed as soon as possible as the PCI bus is held in wait states until the interrupt vector data is returned. Each cycle appears as an interrupt acknowledge pulse on the INTA# pin of the cascaded interrupt controllers. These two pulses are not observable at the ESC periphery.
5. Upon receiving the first internally generated interrupt acknowledge, the highest priority ISR bit is set and the corresponding IRR bit is reset. The Interrupt Controller does not drive the Data Bus during this cycle. On the trailing edge of the first cycle pulse, a slave identification code is broadcast by the master to the slave on a private, internal three bit wide bus. The slave controller uses these bits to determine if it must respond with an interrupt vector during the second INTA# cycle.
6. Upon receiving the second internally generated interrupt acknowledge, the Interrupt Controller releases an 8-bit pointer (the interrupt vector) onto the Data Bus where it is read by the CPU.
7. This completes the interrupt cycle. In the AEOI mode the ISR bit is reset at the end of the second interrupt acknowledge cycle pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt subroutine.

If no interrupt request is present at step four of either sequence (i.e., the request was too short in duration) the Interrupt Controller will issue an interrupt level 7.

9.3 80x86 Mode

When initializing the control registers of the 82C59, an option exists in Initialization Control Word Four (ICW4) to select either an 80x86 or an MSC-85 microprocessor based system. The interrupt acknowledge cycle is different in an MSC-85 based system than in the 80x86 based system: the interrupt acknowledge takes three INTA# pulses with the MSC-85, rather than the two pulses with the 80x86. The ESC is used only in an 80x86 based system. You must program each interrupt controller's ICW4 bit 0 to a "1" to indicate that the interrupt controller is operating in an 80x86 based system. This setting ensures proper operation during an interrupt acknowledge.

9.3.1 ESC INTERRUPT ACKNOWLEDGE CYCLE

As discussed, the CPU generates an interrupt acknowledge cycle that is translated into a single PCI command and broadcast across the PCI bus to the PCEB. The PCEB pulses the INTA# signal to the ESC. The ESC Interrupt Unit translates the INTA# signal into the two INTA# pulses expected by the interrupt controller subsystem. The Interrupt Controller uses the first interrupt acknowledge cycle to internally freeze the state of the interrupts for priority resolution. The first controller (CNTRL-1), as a master, issues a three bit interrupt code on the cascade lines to CNTRL-2 (internal to the ESC) at the end of the INTA# pulse. On this first cycle the interrupt controller block does not issue any data to the processor and leaves its data bus buffers disabled. CNTRL-2 decodes the information on the cascade lines, compares the code to the byte stored in Initialization

Command Word Three (ICW3), and determines if it will have to broadcast the interrupt vector during the second interrupt acknowledge cycle. On the second interrupt acknowledge cycle, the master (CNTRL-1) or slave (CNTRL-2), will send a byte of data to the processor with the acknowledged interrupt code composed as follows:

Table 25. Content of Interrupt Vector Byte for 80x86 System Mode

Interrupt	D7	D6	D5	D4	D3	D2	D1	D0
IRQ7,15	T7	T6	T5	T4	T3	1	1	1
IRQ6,14	T7	T6	T5	T4	T3	1	1	0
IRQ5,13	T7	T6	T5	T4	T3	1	0	1
IRQ4,12	T7	T6	T5	T4	T3	1	0	0
IRQ3,11	T7	T6	T5	T4	T3	0	1	1
IRQ2,10	T7	T6	T5	T4	T3	0	1	0
IRQ1,9	T7	T6	T5	T4	T3	0	0	1
IRQ0,8	T7	T6	T5	T4	T3	0	0	0

NOTE:
T7–T3 represent the interrupt vector address (refer Register Description section).

The byte of data released by the interrupt unit onto the data bus is referred to as the "interrupt vector". The format for this data is illustrated on a per-interrupt basis in Table 25.

9.4 Programming The Interrupt Controller

The Interrupt Controller accepts two types of command words generated by the CPU or bus master:

1. **Initialization Command Words (ICWs):** Before normal operation can begin, each Interrupt Controller in the system must be initialized. In the 82C59, this is a two to four byte sequence. However, for the ESC, each controller must be initialized with a four byte sequence. This four byte sequence is required to configure the interrupt controller correctly for the ESC implementation. This implementation is EISA-compatible.

   The four initialization command words are referred to by their acronyms: ICW1, ICW2, ICW3, and ICW4.

   The base address for each interrupt controller is a fixed location in the I/O memory space, at 0020h for CNTRL-1 and at 00A0h for CNTRL-2.

   An I/O write to the CNTRL-1 or CNTRL-2 base address with data bit 4 equal to 1 is interpreted as ICW1. For ESC-based EISA systems, three I/O writes to "base address + 1" (021h for CNTRL-1 and 0A0h for CNTRL-2) must follow the ICW1. The first write to "base address + 1" (021h/0A0h) performs ICW2, the second write performs ICW3, and the third write performs ICW4.

ICW1 starts the initialization sequence during which the following automatically occur:

a. Following initialization, an interrupt request (IRQ) input must make a low-to-high transition to generate an interrupt.

b. The Interrupt Mask Register is cleared.

c. IRQ7 input is assigned priority 7.

d. The slave mode address is set to 7.

e. Special Mask Mode is cleared and Status Read is set to IRR.

ICW2 is programmed to provide bits[7:3] of the interrupt vector that will be released onto the data bus by the interrupt controller during an interrupt acknowledge. A different base [7:3] is selected for each interrupt controller. Suggested values for a typical EISA system are listed in Table 26.

ICW3 is programmed differently for CNTRL-1 and CNTRL-2, and has a different meaning for each controller.

For CNTRL-1, the master controller, ICW3 is used to indicate which IRQx input line is used to cascade CNTRL-2, the slave controller. Within the ESC interrupt unit, IRQ2 on CNTRL-1 is used to cascade the INTR output of CNTRL-2. Consequently, bit 2 of ICW3 on CNTRL-1 is set to a 1, and the other bits are set to 0's.

For CNTRL-2, ICW3 is the slave identification code used during an interrupt acknowledge cycle. CNTRL-1 broadcasts a code to CNTRL-2 over three internal cascade lines if an IRQ[x] line from CNTRL-2 won the priority arbitration on the master controller and was granted an interrupt acknowledge by the CPU. CNTRL-2 compares this identification code to the value stored in ICW3, and if the code is equal to bits[2:0] of ICW3, CNTRL-2 assumes responsibility for broadcasting the interrupt vector during the second interrupt acknowledge cycle pulse.

ICW4 must be programmed on both controllers. At the very least, bit 0 must be set to a 1 to indicate that the controllers are operating in an 80x86 system.

2. **Operation Command Words (OCWs):** These are the command words which dynamically reprogram the Interrupt Controller to operate in various interrupt modes.

   Any interrupt lines can be masked by writing an OCW1. A 1 written in any bit of this command word will mask incoming interrupt requests on the corresponding IRQx line.

   OCW2 is used to control the rotation of interrupt priorities when operating in the rotating priority mode and to control the End of Interrupt (EOI) function of the controller.

   OCW3 is used to set up reads of the ISR and IRR, to enable or disable the Special Mask Mode (SMM), and to set up the interrupt controller in polled interrupt mode.

   The OCWs can be written into the Interrupt Controller any time after initialization. Table 26 shows an example of typical values programmed by the BIOS at power-up for the ESC interrupt controller.

Table 26. Suggested Default Values For Interrupt Controller Registers

Port	Value	Description of Contents
020h	11h	CNTLR-1, ICW1
021h	08h	CNTLR-1, ICW2 Vector Address for 000020h
021h	04h	CNTLR-1, ICW3 Indicates Slave Connection
021h	01h	CNTLR-1, ICW4 8086 Mode
021h	B8h	CNTLR-1, Interrupt Mask (may vary)
4D0h	00h	CNTLR-1, Edge/Level Control Register
0A0h	11h	CNTLR-2, ICW1
0A1h	70h	CNTLR-2, ICW2 Vector Address for 0001C0h
0A1h	02h	CNTLR-2, ICW3 Indicates Slave ID
0A1h	01h	CNTLR-2, ICW4 8086 Mode
4D1h	00h	CNTLR-2, Edge/Level Control Register
0A1h	BDh	CNTLR-2, Interrupt Mask (may vary)

Figure 20 illustrates the sequence software must follow to load the interrupt controller Initialization Command Words (ICWs). The sequence must be executed for CNTRL-1 and CNTRL-2. After writing ICW1, ICW2, ICW3, and ICW4 must be written in order. Any divergence from this sequence, such as an attempt to program an OCW, will result in improper initialization of the interrupt controller and unexpected, erratic system behavior. It is suggested that CNTRL-2 be initialized first, followed by CNTRL-1.

In the ESC, it is required that all four Initialization Command Words (ICWs) be initialized. As shown in Figure 20, all ICWs must be programmed prior to programming the OCWs.

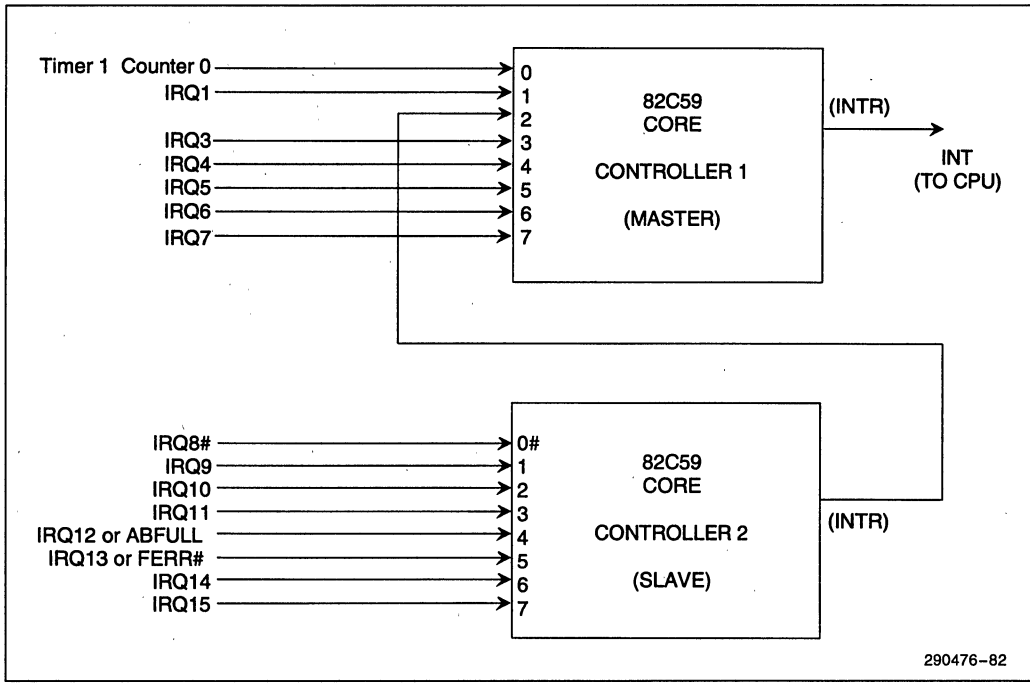


Figure 20. Initialization Sequence for ESC Initialization Command Words (ICWs)

9.5 End-Of-Interrupt Operation

9.5.1 END OF INTERRUPT (EOI)

The In Service (IS) bit can be reset either automatically following the trailing edge of the second internal INTA# pulse (when AEOI bit in ICW1 is set) or by a command word that must be issued to the Interrupt Controller before returning from a service routine (EOI command). An EOI command must be issued twice with this cascaded interrupt controller configuration, once for master and once for the slave.

There are two forms of EOI commands: Specific and Non-Specific. When the Interrupt Controller is operated in modes which preserve the fully nested structure, it can determine which IS bit to reset on EOI. When a Non-Specific EOI command is issued, the Interrupt Controller will automatically reset the highest IS bit of those that are set, since in the fully nested mode the highest IS level was necessarily the last level acknowledged and serviced. A non-specific EOI can be issued with OCW2 (EOI = 1, SL = 0, R = 0).

When a mode is used which may disturb the fully nested structure, the Interrupt Controller may no longer be able to determine the last level acknowledged. In this case a Specific End of Interrupt must be issued which includes as part of the command the IS level to be reset. A specific EOI can be issued with OCW2 (EOI = 1, SL = 1, R = 0, and L0-L2 is the binary level of the IS bit to be reset).

It should be noted that an IS bit that is masked by an IMR bit will not be cleared by a non-specific EOI if the Interrupt Controller is in the Special Mask Mode.

9.5.2 AUTOMATIC END OF INTERRUPT (AEOI) MODE

If AEOI = 1 in ICW4, then the Interrupt Controller will operate in AEOI mode continuously until reprogrammed by ICW4. Note that reprogramming ICW4 implies that ICW1, ICW2, and ICW3 must be reprogrammed first, in sequence. In this mode the Interrupt Controller will automatically perform a non-specific EOI operation at the trailing edge of the last interrupt acknowledge pulse. Note that from a system standpoint, this mode should be used only when a nested multilevel interrupt structure is not required within a single Interrupt Controller. The AEOI mode can only be used in a master Interrupt Controller and not a slave (on CNTRL-1 but not CNTRL-2).

9.6 Modes Of Operation

9.6.1 FULLY NESTED MODE

This mode is entered after initialization unless another mode is programmed. The interrupt requests are ordered in priority from 0 through 7 (0 being the highest). When an interrupt is acknowledged the highest priority request is determined and its vector placed on the bus. Additionally, a bit of the Interrupt Service register (IS[0:7]) is set. This IS bit remains set until the microprocessor issues an End of Interrupt (EOI) command immediately before returning from the service routine. Or, if the AEOI (Automatic End of Interrupt) bit is set, this IS bit remains set until the trailing edge of the second internal INTA#. While the IS bit is set, all further interrupts of the same or lower priority are inhibited, while higher levels will generate an interrupt (which will be acknowledged only if the microprocessor internal Interrupt enable flip-flop has been re-enabled through software).

After the initialization sequence, IRQ0 has the highest priority and IRQ7 the lowest. Priorities can be changed, as will be explained, in the rotating priority mode.

9.6.2 THE SPECIAL FULLY NESTED MODE

This mode will be used in the case of a big system where cascading is used, and the priority has to be conserved within each slave. In this case the special fully nested mode will be programmed to the master (using ICW4). This mode is similar to the normal nested mode with the following exceptions:

1. When an interrupt request from a certain slave is in service, this slave is not locked out from the master's priority logic and further interrupt requests from higher priority IRQ's within the slave will be recognized by the master and will initiate interrupts to the processor. (In the normal nested mode a slave is masked out when its request is in service and no higher requests from the same slave can be serviced.)
2. When exiting the Interrupt Service routine the software has to check whether the interrupt serviced was the only one from that slave. This is done by sending a non-specific End of Interrupt (EOI) command to the slave and then reading its In-Service register and checking for zero. If it is empty, a non-specific EOI can be sent to the master too. If not, no EOI should be sent.

9.6.3 AUTOMATIC ROTATION (EQUAL PRIORITY DEVICES)

In some applications there are a number of interrupting devices of equal priority. Automatic rotation mode provides for a sequential 8-way rotation. In this mode a device receives the lowest priority after being serviced. In the worst case, a device requesting an interrupt will have to wait until each of seven other devices are serviced at most once. Figure 21 shows an example of automatic rotation.

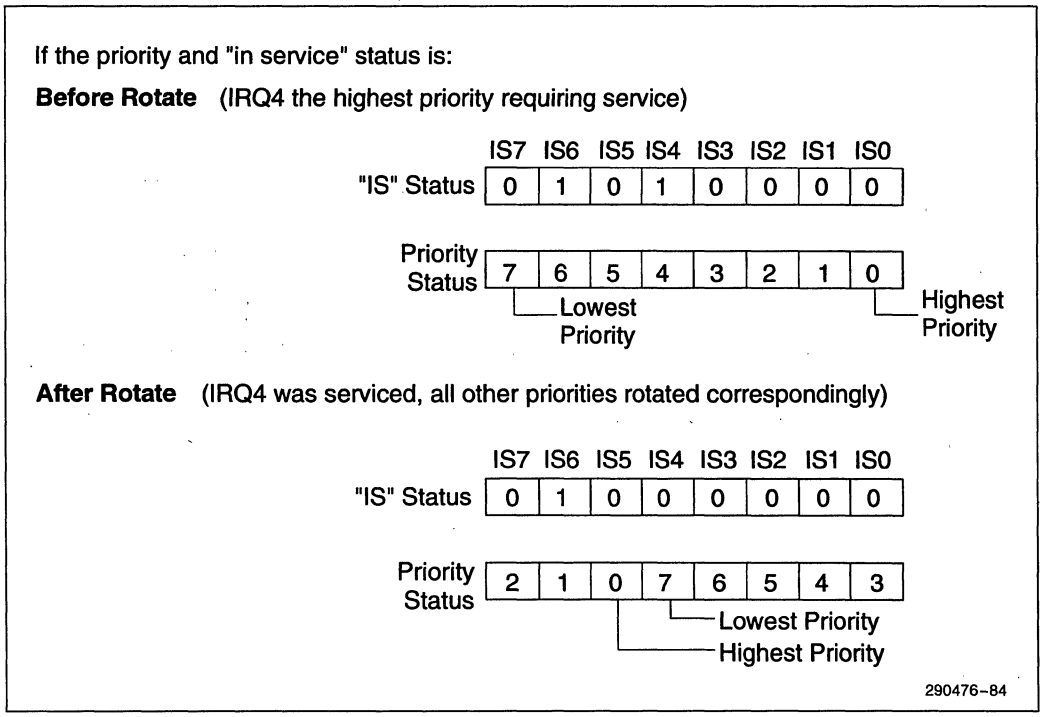


Figure 21. Automatic Rotation Mode Example

There are two ways to accomplish Automatic Rotation using OCW2, the Rotation on Non-Specific EOI Command (R=1, SL=0, EOI=1) and the Rotate in Automatic EOI Mode which is set by (R=1, SL=0, EOI=0) and cleared by (R=0, SL=0, EOI=0).

9.6.4 SPECIFIC ROTATION (SPECIFIC PRIORITY)

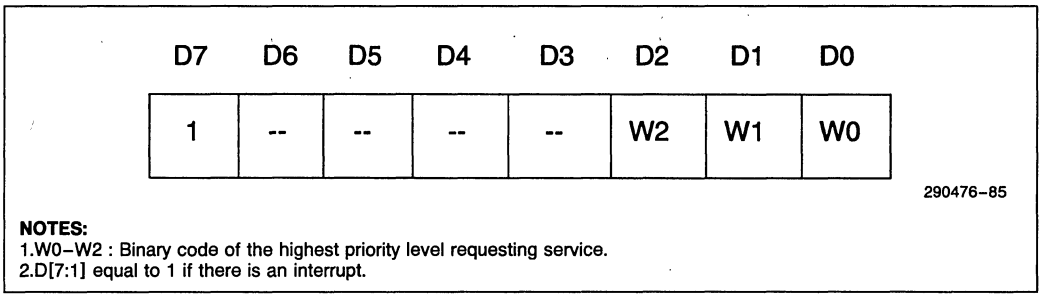
The programmer can change priorities by programming the bottom priority and thus fixing all other priorities. For example, if IRQ5 is programmed as the bottom priority device, then IRQ6 will be the highest priority device. The Set Priority command is issued in OCW2 where: R=1, SL=1; L0–L2 is the binary priority level code of the bottom priority device. See the register description for the bit definitions.

Note that in this mode internal status is updated by software control during OCW2. However, it is independent of the End of Interrupt (EOI) command (also executed by OCW2). Priority changes can be executed during an EOI command by using the Rotate on Specific EOI command in OCW2 (R=1, SL=1, EOI=1 and L0–L2=IRQ level to receive bottom priority).

9.6.5 POLL COMMAND

The Polled Mode can be used to conserve space in the interrupt vector table. Multiple interrupts that can be serviced by one interrupt service routine do not need separate vectors if the service routine uses the poll command. The Polled Mode can also be used to expand the number of interrupts. The polling interrupt service routine can call the appropriate service routine, instead of providing the interrupt vectors in the vector Table. In this mode the INTR output is not used and the microprocessor internal Interrupt Enable flip-flop is reset, disabling its interrupt input. Service to devices is achieved by software using a Poll command.

The Poll command is issued by setting P=1 in OCW3. The Interrupt Controller treats the next I/O read pulse to the Interrupt Controller as an interrupt acknowledge, sets the appropriate IS bit if there is a request, and reads the priority level. Interrupts are frozen from the I/O write to the I/O read. The word enabled onto the data bus during I/O read is shown in Figure 22.



NOTES:
1.W0–W2 : Binary code of the highest priority level requesting service.
2.D[7:1] equal to 1 if there is an interrupt.

Figure 22. Polled Mode

This mode is useful if there is a routine command common to several levels so that the INTA# sequence is not needed (saves ROM space).

9.6.6 CASCADE MODE

The Interrupt Controllers in the ESC system are interconnected in a cascade configuration with one master and one slave. This configuration can handle up to 15 separate priority levels.

The master controls the slaves through a three line internal cascade bus. When the master drives 010b on the cascade bus, this bus acts like a chip select to the slave controller.

In a cascade configuration, the slave interrupt outputs are connected to the master interrupt request inputs. When a slave request line is activated and afterwards acknowledged, the master will enable the corresponding slave to release the interrupt vector address during the second INTA# cycle of the interrupt acknowledge sequence.

Each Interrupt Controller in the cascaded system must follow a separate initialization sequence and can be programmed to work in a different mode. An EOI command must be issued twice: once for the master and once for the slave.

9.6.7 EDGE AND LEVEL TRIGGERED MODES

There are two ELCR registers, one for each 82C59 bank. They are located at I/O ports 04D0h (for the Master Bank, IRQ[0:1,3:7]) and 04D1h (for the Slave Bank, IRQ[8#:15]). They allow the edge and level sense selection to be made on an interrupt by interrupt basis instead of on a complete bank. Only the interrupts that connect to the EISA bus may be programmed for level sensitivity. That is IRQ (0,1,2,8#,13) must be programmed for edge sensitive operation. The LTIM bit is disabled in the ESC. The default programming is equivalent to programming the LTIM bit (ICW1 bit 3) to a 0.

If an ELCR bit is equal to "0", an interrupt request will be recognized by a low to high transition on the corresponding IRQ input. The IRQ input can remain high without generating another interrupt.

If an ELCR bit is equal to "1", an interrupt request will be recognized by a "low" level on the corresponding IRQ input, and there is no need for an edge detection. For level triggered interrupt mode, the interrupt request signal must be removed before the EOI command is issued or the CPU interrupt must be disabled. This is necessary to prevent a second interrupt from occurring.

In both the edge and level triggered modes the IRQ inputs must remain active until after the falling edge of the first INTA#. If the IRQ input goes inactive before this time a DEFAULT IRQ7 will occur when the CPU acknowledges the interrupt. This can be a useful safeguard for detecting interrupts caused by spurious noise glitches on the IRQ inputs. To implement this feature the IRQ7 routine is used for "clean up" simply executing a return instruction, thus ignoring the interrupt. If IRQ7 is needed for other purposes a default IRQ7 can still be detected by reading the ISR. A normal IRQ7 interrupt will set the corresponding ISR bit, a default IRQ7 won't. If a default IRQ7 routine occurs during a normal IRQ7 routine, however, the ISR will remain set. In this case it is necessary to keep track of whether or not the IRQ7 routine was previously entered. If another IRQ7 occurs it is a default.

IRQ13 still appears externally to be an edge sensitive interrupt even though it is shared internally with the Chaining interrupt. The Chaining interrupt is ORed after the edge sense logic.

9.7 Register Functionality

For a detailed description of the Interrupt Controller register set, please see Section 3.4, Interrupt Controller Register.

9.7.1 INITIALIZATION COMMAND WORDS

Four initialization command words (ICWs) are used to initialize each interrupt controller. Each controller is initialized separately. Following this initialization sequence, the interrupt controller is ready to accept interrupts.

9.7.2 OPERATION CONTROL WORDS (OCWS)

After the Initialization Command Words (ICWs) are programmed into the Interrupt Controller, the chip is ready to accept interrupt requests at its input lines. However, Interrupt Controller operation can be dynamically modified to fit specific software/hardware expectations. Different modes of operation are dynamically selected following initialization through the use of Operation Command Words (OCWs).

9.8 Interrupt Masks

9.8.1 MASKING ON AN INDIVIDUAL INTERRUPT REQUEST BASIS

Each Interrupt Request input can be masked individually by the Interrupt Mask register (IMR). This register is programmed through OCW1. Each bit in the IMR masks one interrupt channel if it is set to a "1". Bit 0 masks IRQ0, bit 1 masks IRQ1 and so forth. Masking an IRQ channel does not effect the other channel's operation, with one notable exception. Masking IRQ[2] on CNTRL-1 will mask off all requests for service from CNTRL-2. The CNTRL-2 INTR output is physically connected to the CNTRL-1 IRQ[2] input.

9.8.2 SPECIAL MASK MODE

Some applications may require an interrupt service routine to dynamically alter the system priority structure during its execution under software control. For example, the routine may wish to inhibit lower priority requests for a portion of its execution but enable some of them for another portion.

The difficulty here is that if an Interrupt Request is acknowledged and an End of Interrupt command did not reset its IS bit (i.e., while executing a service routine), the Interrupt Controller would have inhibited all lower priority requests with no easy way for the routine to enable them.

The Special Mask Mode enables all interrupts not masked by a bit set in the Mask Register. Interrupt service routines that require dynamic alteration of interrupt priorities can take advantage of the Special Mask Mode. For example, a service routine can inhibit lower priority requests during a part of the interrupt service, then enable some of them during another part.

In the Special Mask Mode, when a mask bit is set in OCW1, it inhibits further interrupts at that level and enables interrupts from all other levels (lower as well as higher) that are not masked.

Thus, any interrupts may be selectively enabled by loading the Mask register with the appropriate pattern.

Without Special Mask Mode, if an interrupt service routine acknowledges an interrupt without issuing an EOI to clear the IS bit, the interrupt controller inhibits all lower priority requests. The Special Mask Mode provides an easy way for the interrupt service routine to selectively enable only the interrupts needed by loading the Mask register.

The special Mask Mode is set by OCW3 where: SSMM = 1, SMM = 1, and cleared where SSMM = 1, SMM = 0.

9.9 Reading The Interrupt Controller Status

The input status of several internal registers can be read to update the user information on the system. The Interrupt Request Register (IRR) and In-Service Register (ISR) can be read via OCW3, as discussed in Section 3.7. The Interrupt Mask Register (IMR) is read via a read of OCW1, as discussed in Section 3.7. Here are brief descriptions of the ISR, the IRR, and the IMR.

Interrupt Request Register (IRR): 8-bit register which contains the status of each interrupt request line. Bits that are clear indicate interrupts that have not requested service. The Interrupt Controller clears the IRR's highest priority bit during an interrupt acknowledge cycle. (Not effected by IMR).

In-Service Register (ISR): 8-bit register indicating the priority levels currently receiving service. Bits that are set indicate interrupts that have been acknowledged and their interrupt service routine started. Bits that are cleared indicate interrupt requests that have not been acknowledged, or interrupt request lines that have not been asserted. Only the highest priority interrupt service routine executes at any time. The lower priority interrupt services are suspended while higher priority interrupts are serviced. The ISR is updated when an End of Interrupt Command is issued.

Interrupt Mask Register (IMR): 8-bit register indicating which interrupt request lines are masked.

The IRR can be read when, prior to the I/O read cycle, a Read Register Command is issued with OCW3 (RR=1, RIS=0).

The ISR can be read when, prior to the I/O read cycle, a Read Register Command is issued with OCW3 (RR=1, RIS=1).

The interrupt controller retains the ISR/IRR status read selection following each write to OCW3. Therefore, there is no need to write an OCW3 before every status read operation, as long as the current status read corresponds to the previously selected register. For example, if the ISR is selected for status read by an OCW3 write, the ISR can be read over and over again without writing to OCW3 again. However, to read the IRR, OCW3 will have to be reprogrammed for this status read prior to the OCW3 read to check the IRR. This is not true when poll mode is used. Polling Mode overrides status read when P=1, RR=1 in OCW3.

After initialization, the Interrupt Controller is set to read the IRR.

As stated, OCW1 is used for reading the IMR. The output data bus will contain the IMR status whenever I/O read is active. The address is 021h or 061h (OCW1).

9.10 Non-Maskable Interrupt (NMI)

An NMI is an interrupt requiring immediate attention and has priority over the normal interrupt lines (IRQx). The ESC indicates error conditions by generating a non-maskable interrupt.

The ESC generates NMI interrupts based on the following Hardware and Software events.

Hardware Events:

1. **Motherboard Parity Errors:** Memory parity errors for the motherboard memory. These errors are reported to the ESC through the PERR# signal line.
2. **System Errors:** System error on the motherboard. The system board uses the SERR# signal to indicate system errors to the ESC.
3. **Add-In Board Parity Errors:** Parity errors on the add-in memory boards on the EISA expansion bus. IOCHK# signal on the EISA bus is driven low by the add-in board logic when this error occurs.
4. **Fail-Safe Timer Timeout:** Fail-Safe Timer (Timer 2, Counter 0) count expires. If this counter has been set and enabled, and the count expires before a software routine can reset the counter.
5. **Bus Timeout:** An EISA bus Master or Slave exceeds the allocated time on the bus. A bus timeout occurs if an EISA Master does not relinquish the bus (MREQ# negated) within 64 BCLKS after it has been preempted (MACK# negated). A bus timeout also occurs if a memory slave extends the cycle (CHRDY negated) long enough to keep CMD# asserted for more than 256 BCLKS. The DMA controller does not cause a bus timeout. The ESC asserts RESDRV when a bus timeout occurs.

Software Events:

1. **Software Generated NMI: If an I/O Write access to Port 0462h occurs. The data value for this write is a don't care.**

The NMI logic incorporates four different 8-bit registers. These registers are used by the CPU to determine the source of the interrupt and to enable or disable/clear the interrupts. See Section 3.4, Interrupt Controller Registers, for the register details.

Table 27. NMI Register I/O Address Map

I/O Port Address	Register Description
0061h	NMI Status Register
0070h	NMI Enable Register
0461h	Extended NMI Register
0462h	Software NMI Register

Table 28. NMI Source Enable/Disable And Status Port Bits

NMI Source	IO Port Bit for Status Reads	IO Port Bit for Enable/Disable
PERR#	Port 0061h, Bit 7	Port 0061h, Bit 2
IOCHK#	Port 0061h, Bit 6	Port 0061h, Bit 3
Fail-Safe	Port 0461h, Bit 7	Port 0461h, Bit 2
Bus Timeout	Port 0461h, Bit 6	Port 0461h, Bit 3
Write to Port 0462h	Port 0461h, Bit 5	Port 0461h, Bit 1

The individual enable/disable bits clear the NMI detect flip-flops when disabled.

All NMI sources can be enabled or disabled by setting Port 070h bit[7]. This disable function does not clear the NMI detect Flip-Flops. This means, if NMI is disabled then enabled via Port 070h, then an NMI will occur when Port 070h is re-enabled if one of the NMI detect Flip-Flops had been previously set.

To ensure that all NMI requests are serviced, the NMI service routine software needs to incorporate a few very specific requirements. These requirements are due to the edge detect circuitry of the host microprocessor, 80386 or 80486. The software flow would need to be the following:

1. NMI is detected by the processor on the rising edge of the NMI input.
2. The processor will read the status stored in port 061h and 0461h to determine what sources caused the NMI. The processor may then reset the register bits controlling the sources that it has determined to be active. Between the time the processor reads the NMI sources and resets them, an NMI may have been generated by another source. The level of NMI will then remain active. This new NMI source will not be recognized by the processor because there was no edge on NMI.
3. The processor must then disable all NMI's by writing bit[7] of port 070H high and then enable all NMI's by writing bit[7] of port 070H low. This will cause the NMI output to transition low then high if there are any pending NMI sources. The CPU's NMI input logic will then register a new NMI.

10.0 ADVANCED PROGRAMMABLE INTERRUPT CONTROLLER (APIC)

In addition to the standard EISA compatible interrupt controller described in the previous section, the ESC incorporates the Advanced Programmable Interrupt Controller (APIC). While the standard interrupt controller is intended for use in a uni-processor system, APIC can be used in either a uni-processor or multi-processor system. APIC provides multi-processor interrupt management and incorporates both static and dynamic symmetric interrupt distribution across all processors. In systems with multiple I/O subsystems, each subsystem can have its own set of interrupts.

In a uni-processor system, APIC's dedicated interrupt bus can reduce interrupt latency over the standard interrupt controller (i.e., the latency associated with the propagation of the interrupt acknowledge cycle across multiple busses using the standard interrupt controller approach). Interrupts can be controlled by the standard EISA compatible interrupt controller unit, the I/O APIC unit, or mixed mode where both the standard and I/O APIC are used. The selection of which controller responds to an interrupt is determined by how the interrupt controllers are programmed. Note that it is the programmer's responsibility to make sure that the same interrupt input signal is not handled by both interrupt controllers.

At the system level, APIC consists of two parts (Figure 23)—one residing in the I/O subsystem (called the I/O APIC) and the other in the CPU (called the Local APIC). The ESC contains an I/O APIC unit. The local APIC and the I/O APIC communicate over a dedicated APIC bus. The ESC's I/O APIC bus interface consists of two bi-directional data signals (APICD[1:0]) and a clock input (APICCLK).

The CPU's Local APIC Unit contains the necessary intelligence to determine whether or not its processor should accept interrupts broadcast on the APIC bus. The Local Unit also provides local pending of interrupts, nesting and masking of interrupts, and handles all interactions with its local processor (e.g., the INTR/INTA/EOI protocol). The Local Unit further provides inter-processor interrupts and a timer, to its local processor. The register level interface of a processor to its local APIC is identical for every processor.

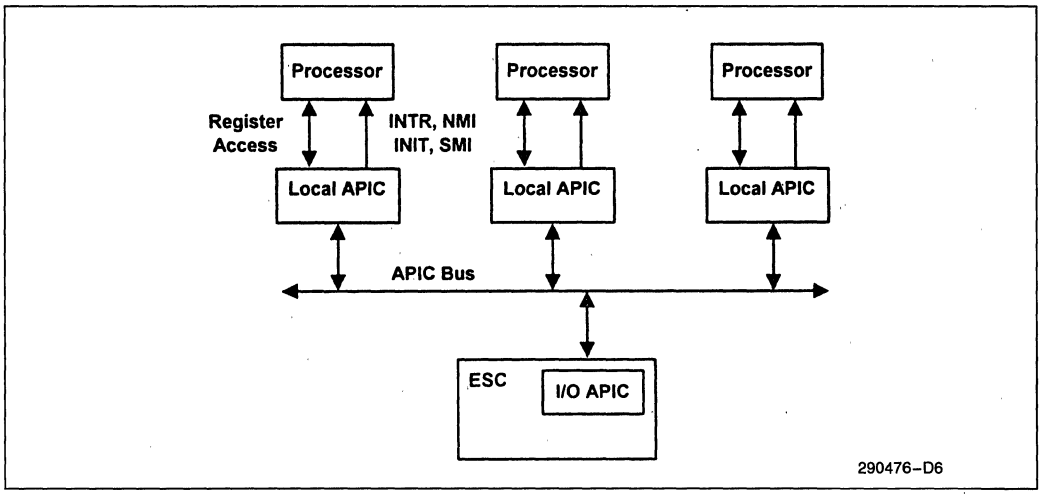


Figure 23. APIC System Structure

The ESC's I/O APIC Unit consists of a set of interrupt input signals, a 16-entry Interrupt Redirection Table, programmable registers, and a message unit for sending and receiving APIC messages over the APIC bus (Figure 24). I/O devices inject interrupts into the system by asserting one of the interrupt lines to the I/O APIC (Figure 25). The I/O APIC selects the corresponding entry in the Redirection Table and uses the information in that entry to format an interrupt request message. Each entry in the Redirection Table can be individually programmed to indicate edge/level sensitive interrupt signals, the interrupt vector and priority, the destination processor, and how the processor is selected (statically or dynamically). The information in the table is used to transmit a message to other APIC units (via the APIC bus).

The ESC's I/O APIC contains a set of programmable registers. Two of the registers (I/O Register Select and I/O Window Registers) are located in the CPU's memory space and are used to indirectly access the other APIC registers as described in Section 3.0, Register Description. The Version Register provides the implementation version of the I/O APIC. The I/O APIC ID Register is programmed with an ID value that serves as a physical name of the I/O APIC. This ID is loaded into the ARB ID Register when the I/O APIC ID Register is written and is used during bus arbitration.

NOTE:

1. When the ESC's I/O APIC receives an interrupt request, the ESC instructs the PCEB to flush its buffers and to request all system buffers pointing to PCI to be flushed (via the AFLUSH# signal). The APIC does not send the interrupt message over the APIC bus until the ESC receives confirmation from the PCEB (via the AFLUSH# signal) that all buffers have been flushed and temporarily disabled.
2. The interrupt number or the vector does not imply a particular priority for being sent. The I/O APIC continually polls the 16 interrupts in a rotating fashion, one at a time. The pending interrupt polled first is the one sent.

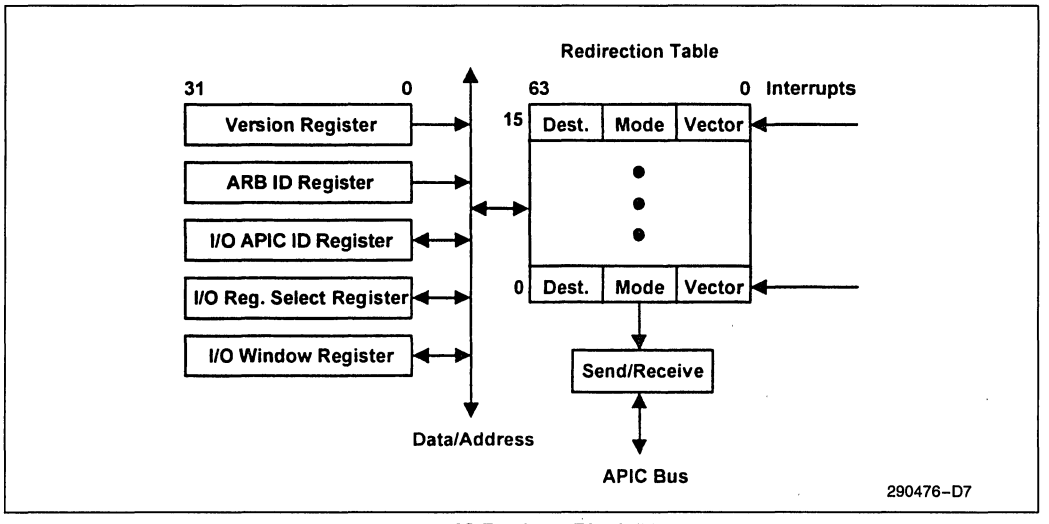


Figure 24. APIC Register Block Diagram

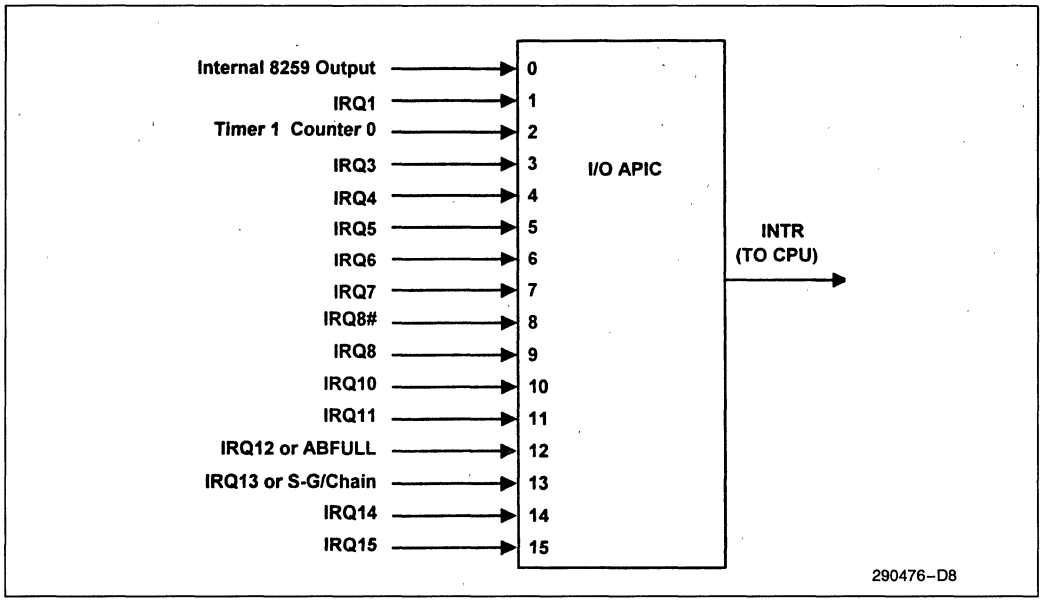


Figure 25. I/O APIC Interrupt Mapping

10.1 Physical Characteristics Of APIC Bus

The APIC bus is a 3-wire synchronous bus connecting all APICs (all I/O units and all local units). Two of these wires are used for data transmission, and one wire is a clock. For bus arbitration, the APIC uses only one of the data wires. The bus is logically a wire-OR and electrically an open-drain connection providing for both message transmission and arbitration for lowest priority. All the values mentioned in the protocol description are logical values (i.e. "Bus Driven" is logical 1 and "Bus Not Driven" is logical 0). The electrical values are 0 for logical one and 1 for logical zero.

10.2 Arbitration For APIC Bus

The APIC uses one wire arbitration to win the bus ownership. A rotating priority scheme is used for arbitration. The winner of the arbitration becomes the lowest priority agent and assumes an arbitration ID of 0. All other agents, except the agent whose arbitration ID is 15, increment their arbitration IDs by one. The agent whose ID was 15 takes the winner's arbitration ID and increments it by one. Arbitration IDs are changed (incremented or assumed) only for messages that are transmitted successfully. For lowest priority messages, the arbitration ID is updated before the final status cycle, which ultimately decides if the message is successful. A message is transmitted successfully if no CS error or acceptance error is reported for that message.

An APIC agent can acquire the bus using two different priority schemes; normal, or EOI (End Of Interrupt). EOI has the highest priority. EOI priority is used to send EOI messages for level interrupts from local APIC to the I/O APIC. When an agent requests the bus with EOI priority, all others requesting the bus with normal priorities back off.

A bus arbitration cycle starts by the agent driving a start cycle (bit 1 = EOI, bit 0 = 1) on the APIC bus (Table 29). Bit 1 = 1 indicates "EOI" priority and bit 1 = 0 indicates normal priority. Bit 0 should be 1.

In cycles 2 through 5, the agent drives the arbitration ID on bit 1 of the bus. High-order ID bits are driven first with successive cycles proceeding to the low bits of the ID. All arbitration losers in a given cycle drop off the bus, using every subsequent cycle as a tie breaker for the previous cycle. When all arbitration cycles are completed, there will be only one agent left driving the bus.

Table 29. Bus Arbitration Cycles

Cycle	Bit 1	Bit 0	Comments
1	EOI	1	0 1 = normal, 1 1 = EOI
2	ArbID3	0	Arbitration ID bits 3 through 0
3	ArbID2	0	
4	ArbID1	0	
5	ArbID0	0	

10.3 Bus Message Formats

After bus arbitration the winner is granted exclusive use of the bus and drives its message on the bus. APIC messages come in four formats—14 cycle EOI Message, 21 cycles Short Message, 33 cycles Lowest Priority Message, and 39 cycles Remote Read Message. All APICs on the APIC bus know the length of an interrupt message by checking the appropriate fields in the message.

EOI Message For Level Triggered Interrupts

The EOI Message is used to send an EOI cycle occurring for a level triggered interrupt from local APIC to the I/O APIC. This cycle contains the priority vector (V[7:0]) of the interrupt. When this message is received, the I/O APIC resets the Remote IRR bit for that interrupt. If the interrupt signal is still active after the RIRR bit is reset, the I/O APIC will treat it as a new interrupt.

Table 30. EOI Message

Cycle	Bit 1	Bit 0	Comments
1	EOI	1	0 1 = normal, 1 1 = EOI
2	ArbID3	0	Arbitration ID bits 3 through 0
3	ArbID2	0	
4	ArbID1	0	
5	ArbID0	0	
6	V7	V6	Interrupt vector V7–V0
7	V5	V4	
8	V3	V2	
9	V1	V0	

Table 30. EOI Message (Continued)

Cycle	Bit 1	Bit 0	Comments
10	C	C	Check Sum
11	0	0	Postamble
12	A	A	Status Cycle0
13	A1	A1	Status Cycle1
14	0	0	Idle

Short Message

Short Messages are used for the delivery of Fixed, NMI, SMI (82374SB only), Reset, ExtINT and Lowest Priority with Focus processor interrupts. The delivery mode bits (M[2:0]) specify the message. All short messages take 21 cycles, including the idle cycle.

Cycle 1 is the start cycle (Table 31). Cycles 2 through 5 are for bus arbitration as described earlier. APIC ID bits are sent on the bus one bit at a time (Only one data bus bit is used). The other bit should be zero. Cycles 6 and 7 provide destination mode and delivery mode bits. Cycle 8 provides level and trigger mode information. Cycles 10 through 13 are the 8-bit interrupt vector. The vector is only defined for delivery modes fixed, and lowest-priority. For delivery mode of "Remote Read", the vector field contains the address of the register to be read remotely.

If Destination Mode (DM) is 0 (physical mode), then cycles 15 and 16 are the APIC ID and cycles 13 and 14 are zero. If DM is 1 (logical mode), then cycles 13 through 16 are the 8-bit destination field. The interpretation of the logical mode 8-bit destination field is performed by the local units using the Destination Format Register. Shorthands of "all-incl-self" and "all-excl-self" both use physical destination mode and a destination field containing APIC ID value of all ones. The sending APIC knows whether it should (incl) or should not (excl) respond to its own message.

Cycle 17 is a checksum for the data in cycles 6 through 16. The (single) APIC driving the message provides this checksum in cycle 17.

Cycle 18 is a postamble cycle driven as 00 by all APICs to perform various internal computations based on the information contained in the received message. One of the computations takes the computed checksum of the data received in cycles 6 through 16 and compares it against the value in cycle 17. If any APIC computes a different checksum than the one passed in cycle 17, then that APIC signals an error on the APIC bus in cycle 19 by driving it as 11. If this happens, all APICs assume the message was never sent and the sender must try sending the message again, which includes re-arbitrating for the APIC bus. In lowest priority delivery when the interrupt has a focus processor, the focus processor signals this by driving 10 during cycle 19. This tells all the other APICs that the interrupt has been accepted, the LP arbitration is preempted, and short message format is used. Cycle 19 and 20 indicates the status of the message (i.e. accepted, check sum error, retry or error). Table 32 shows the status signals combinations and their meanings for all delivery modes.

The checksum is calculated iteratively on each cycle by adding the following terms:

1. The two least significant bits from the last cycle's checksum.
2. The current two data bits.
3. The carry bit from the last cycle's checksum shifted to the least significant bit.

Note that, at the beginning of the calculation, the three bits composing the previous cycle's checksum (two lower bits and carry) are zero.

Table 31. Short Message

Cycle	Bit 1	Bit 0	Comments
1	0	1	0 1 = Normal, 1 1 = EOI
2	ArbID3	0	Arbitration ID bits 3 through 0
3	ArbID2	0	
4	ArbID1	0	
5	ArbID0	0	
6	DM	M2	DM = Destination Mode
7	M1	M0	M2–M0 = Delivery Mode
8	L	TM	L = Level, TM = Trigger Mode
9	V7	V6	V7–V0 = Interrupt Vector
10	V5	V4	
11	V3	V2	
12	V1	V0	
13	D7	D6	Destination
14	D5	D4	
15	D3	D2	
16	D1	D0	
17	C	C	Checksum for Cycles 6–16
18	0	0	Postamble
19	A	A	Status Cycle 0
20	A1	A1	Status Cycle 1
21	0	0	Idle

Table 32. APIC Bus Message Status Information

Delivery Mode	Focus Processor	Status: A A	Comments	Status: A1 A1	Comments
Fixed, EOI	N/A	00	CS is OK	10	Accepted
Fixed, EOI	N/A	00	CS is OK	11	Retry
Fixed, EOI	N/A	00	CS is OK	0X	Error
Fixed, EOI	N/A	11	CS Error	XX	
Fixed, EOI	N/A	10	Error	XX	
Fixed, EOI	N/A	01	Error	XX	
NMI, SMM, Reset, ExtINT	N/A	00	CS is OK	10	Accepted
NMI, SMM, Reset, ExtINT	N/A	00	CS is OK	11	Error
NMI, SMM, Reset, ExtINT	N/A	00	CS is OK	0X	Error
NMI, SMM, Reset, ExtINT	N/A	11	CS Error	XX	
NMI, SMM, Reset, ExtINT	N/A	10	CS Error	XX	
NMI, SMM, Reset, ExtINT	N/A	01	CS Error	XX	
Lowest Priority	No	00	CS is OK. No Focus Processor	11	Go for LP Arb.
Lowest Priority	No	00	CS is OK. No Focus Processor	10	End and Retry
Lowest Priority	No	00	CS is OK. No Focus Processor	0X	Error
Lowest Priority	Yes	10	CS is OK. Focus Processor	XX	
Lowest Priority	Yes	11	CS Error	XX	
Lowest Priority	Yes	01	Error	XX	
Remote Read	N/A	00	CS is OK	XX	
Remote Read	N/A	11	CS Error	XX	
Remote Read	N/A	01	Error	XX	
Remote Read	N/A	10	Error	XX	

NOTE:
CS = Check Sum

Lowest Priority (LP) without Focus Processor (FP) Message

This message format is used to deliver an interrupt in the lowest priority mode in which it does not have a Focus Processor. Cycles 1 through 20 for this message are the same as for the Short Message discussed above. Status cycle 19 identifies if there is not a Focus processor (00) and a status value of 11 in cycle 20 indicates the need for lowest priority arbitration.

Cycle 21 through 28 are used to arbitrate for the lowest priority processor. The processors that take part in the arbitration drive their processor priority on the bus. Only the local APICs that have "free interrupt slots" participate in the lowest priority arbitration.

Cycle 29 through 32 are used to break a tie in case two or more processors have lowest priority. The bus arbitration IDs are used to break the tie.

Cycle 33 is an additional status cycle driven by the accepting local APIC. By receiving 10 during this cycle, the sending I/O APIC knows that there was a local APIC left after LP arbitration. Otherwise, the message is retried. Cycle 34 is an idle cycle.

Table 33. Lowest Priority without Focus Processor Message

Cycle	Bit 1	Bit 0	
1	0	1	0 1 = normal, 1 1 = EOI
2	ArbID3	0	Arbitration ID bits 3 through 0
3	ArbID2	0	
4	ArbID1	0	
5	ArbID0	0	
6	DM	M2	DM = Destination mode
7	M1	M0	M2–M0 = Delivery mode
8	L	TM	L = Level, TM = Trigger Mode
9	V7	V6	V7–V0 = Interrupt Vector
10	V5	V4	
11	V3	V2	
12	V1	V0	
13	D7	D6	Destination
14	D5	D4	
15	D3	D2	
16	D1	D0	
17	C	C	Checksum for cycles 6-16
18	0	0	Postamble
19	A	A	Status cycle 0
20	A1	A1	Status cycle 1

Table 33. Lowest Priority without Focus Processor Message (Continued)

Cycle	Bit 1	Bit 0	
21	P7	0	Inverted Processor Priority P7–P0
22	P6	0	
23	P5	0	
24	P4	0	
25	P3	0	
26	P2	0	
27	P1	0	
28	P0	0	
29	ArbID3	0	Arbitration ID 3 -0
30	ArbID2	0	
31	ArbID1	0	
32	ArbID0	0	
33	S	S	Status (10 means status OK; all other values indicate an error)
34	0	0	Idle

Remote Read Message

The Remote Read Message (Table 34) is used for a local APIC to read the register in another local APIC. The message format is the same as the Short Message for the first 20 cycles.

Cycles 21 through 36 contain the Remote Register data. The status information in cycle 37 specifies if the data is good or not. The Remote Read Message is always successful in that it is never retried (although the data may be valid or invalid). The reason is that the Remote Read Message is a debug feature, and a "hung" remote APIC that is unable to respond should not cause the debugger to hang up.

Table 34. Remote Read Message

Cycle	Bit 1	Bit 0	
1	0	1	0 1 = normal, 1 1 = EOI
2	ArbID3	0	Arbitration ID bits 3 through 0
3	ArbID2	0	
4	ArbID1	0	
5	ArbID0	0	
6	DM	M2	DM = Destination mode
7	M1	M0	M2–M0 = Delivery mode
8	L	TM	L = Level, TM = Trigger Mode

Table 34. Remote Read Message (Continued)

Cycle	Bit 1	Bit 0	
9	V7	V6	V7–V0 = Interrupt Vector
10	V5	V4	
11	V3	V2	
12	V1	V0	
13	D7	D6	Destination
14	D5	D4	
15	D3	D2	
16	D1	D0	
17	C	C	Checksum for cycles 6-16
18	0	0	Postamble
19	A	A	Status cycle 0
20	A1	A1	Status cycle 1
21	d31	d30	Remote register data 31-0
22	d29	d28	
23	d27	d26	
24	d25	d24	
25	d23	d22	
26	d21	d20	
27	d19	d18	
28	d17	d16	
29	d15	d14	
30	d13	d12	
31	d11	d10	
32	d09	d08	
33	d07	d06	
34	d05	d04	
35	d03	d02	
36	d01	d00	
37	S	S	Data Status: 11 = valid, 00 = invalid
38	C	C	Check Sum for data d[31:00]
39	0	0	Idle

11.0 PCEB/ESC INTERFACE

The PCEB/ESC interface (Figure 26) provides the inter-chip communications between the PCEB and ESC. The interface provides control information between the two components for PCI/EISA arbitration, data size translations (controlling the PCEB's EISA data swap logic), and interrupt acknowledge cycles.

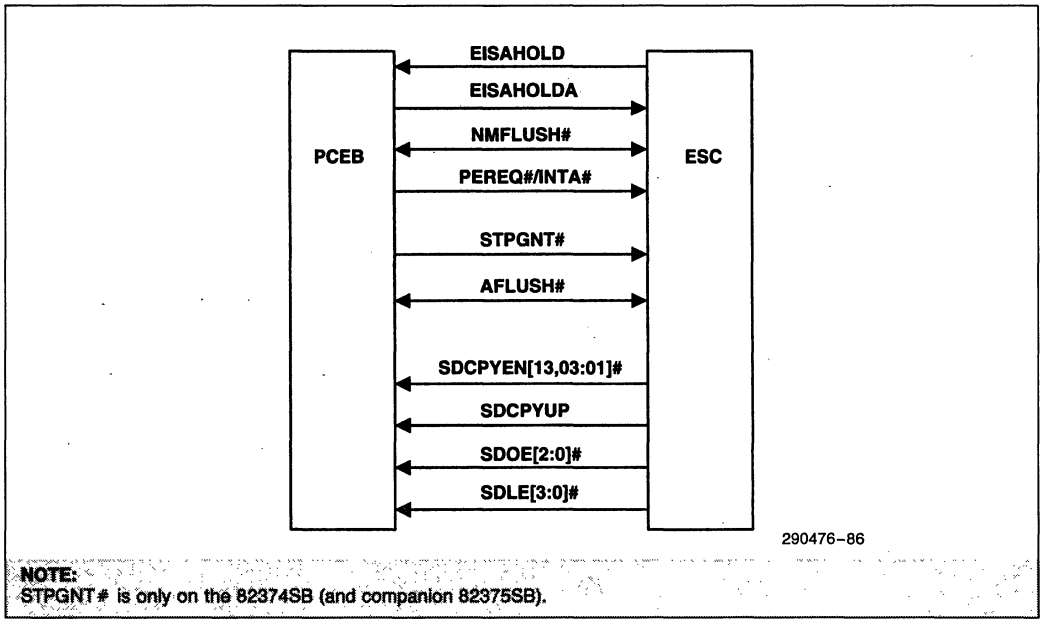


Figure 26. PCEB/ESC Interface Signals

11.1 Arbitration Control Signals

The PCEB contains the arbitration circuitry for the PCI Bus and the ESC contains the arbitration circuitry for the EISA Bus. The PCEB/ESC Interface contains a set of arbitration control signals (EISAHOLD, EISAHOLDA, NMFLUSH#, and PEREQ#/INTA#) that synchronize bus arbitration and ownership changes between the two bus environments. The signals also force PCI device data buffer flushing, if needed, to maintain data coherency during EISA Bus ownership changes.

The PCEB is the default owner of the EISA Bus. If another EISA/ISA master or DMA wants to use the bus, the ESC asserts EISAHOLD to instruct the PCEB to relinquish EISA Bus ownership. The PCEB completes any current EISA Bus transaction, tri-states its EISA Bus signals, and asserts EISAHOLDA to inform the ESC that the PCEB is off the bus.

For ownership changes, other than for a refresh cycle, the ESC asserts the NMFLUSH# signal to the PCEB (for one PCICLK) to instruct the PCEB to flush its Line Buffers pointing to the PCI Bus. The assertion of NMFLUSH# also instructs the PCEB to initiate flushing and to temporarily disable system buffers on the PCI Bus (via MEMREQ#, MEMACK, and FLSHREQ#). The buffer flushing maintains data coherency, in the event that the new EISA Bus master wants to access the PCI Bus. Buffer flushing also prevents dead-lock conditions between the PCI Bus and EISA Bus. Since the ESC/PCEB does not know ahead of time, whether the new master is going to access the PCI Bus or a device on the EISA Bus, buffers pointing to the PCI Bus are always flushed when there is a change of EISA Bus ownership, except for refresh cycles. For refresh cycles, the ESC

controls the cycle and, thus, knows that the cycle is not an access to the PCI Bus and does not initiate a flush request to the PCEB. After a refresh cycle, the ESC always surrenders control of the EISA Bus back to the PCEB.

NMFLUSH# is a bi-directional signal that is negated by the ESC when buffer flushing is not being requested. The ESC asserts NMFLUSH# to request buffer flushing. When the PCEB samples NMFLUSH# asserted, it starts driving the signal in the asserted state and begins the buffer flushing process. (The ESC tristates NMFLUSH# after asserting if for the initial 1 PCICLK period.) The PCEB keeps NMFLUSH# asserted until all buffers are flushed and then it negates the signal for 1 PCICLK. When the ESC samples NMFLUSH# negated, it starts driving the signal in the negated state, completing the handshake. When the ESC samples NMFLUSH# negated, it grants ownership to the winner of the EISA Bus arbitration (at the time NMFLUSH# was negated). Note that for a refresh cycle, NMFLUSH# is not asserted by the ESC.

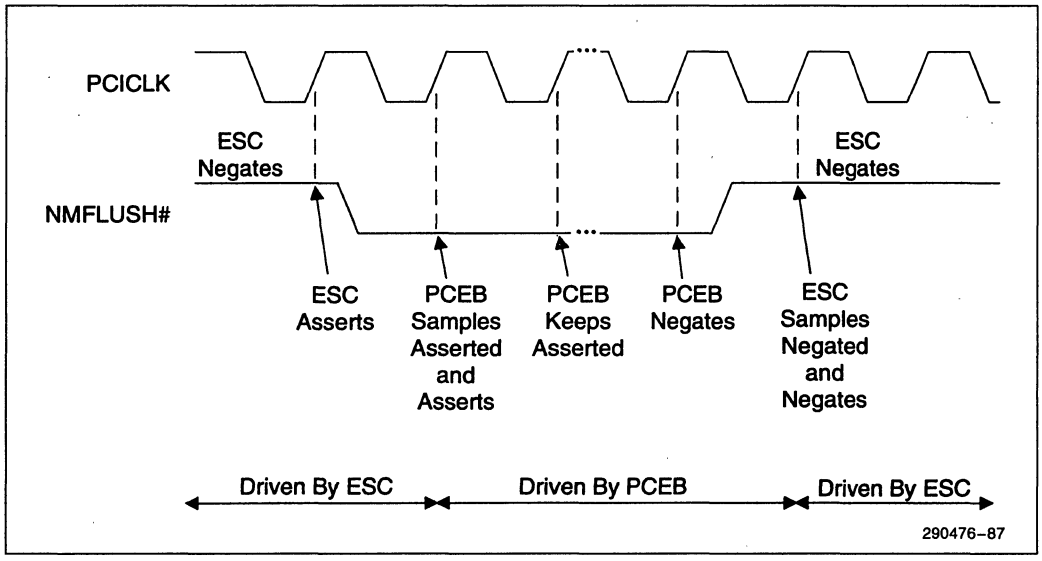


Figure 27. NMFLUSH# Protocol

When the EISA master completes its transfer and gets off the bus (i.e., removes its request to the ESC), the ESC negates EISAHOLD and the PCEB, in turn, negates EISAHOLDA. At this point, the PCEB resumes its default ownership of the EISA Bus.

If a PCI master requests access to the EISA Bus while the bus is owned by a master other than the PCEB, the PCEB retries the PCI cycle and requests ownership of the EISA Bus by asserting PEREQ#/INTA# to the ESC. PEREQ#/INTA# is a dual function signal that is a PCEB request for the EISA Bus (PEREQ# function) when EISAHOLDA is asserted. In response to the PCEB request for EISA Bus ownership, the ESC removes the grant to the EISA master. When the EISA master completes its current transactions and relinquishes the bus (removes its bus request), the ESC negates EISAHOLD and the PCEB, in turn, negates EISAHOLDA. At this point, a grant can be given to the PCI device for a transfer to the EISA Bus. Note that the INTA# function of the PEREQ#/INTA# signal is described in Section 11.5, Interrupt Acknowledge Control.

11.2 System Buffer Coherency Control—APIC

During an interrupt sequence, the system buffers must be flushed before the ESC's I/O APIC can send an interrupt message to the local APIC (CPU's APIC). The ESC and PCEB maintain buffer coherency when the ESC receives an interrupt request for its I/O APIC using the AFLUSH# signal.

11.3 Power Management

In response to the ESC's STPCLK# assertion, the CPU sends out a stop grant bus cycle to indicate that it has entered the stop grant state. The PCEB informs the ESC of the stop grant cycle using the STPGNT# signal.

11.4 EISA Data Swap Buffer Control Signals

The cycles in the EISA environment may require data size translations before the data can be transferred to its intermediate or final destination. As an example, a 32-bit EISA master write cycle to a 16-bit EISA slave requires a disassembly of a 32-bit Dword into 16 bit Words. Similarly, a 32-bit EISA master read cycle to a 16-bit slave requires an assembly of two 16-bit Words into a 32-bit Dword. The PCEB contains EISA data swap buffers to support data size translations on the EISA Bus. The operation of the data swap logic is described in the PCEB data sheet. The ESC controls the operation of the PCEB's data swap logic with the following PCEB/ESC interface signals. These signals are outputs from the ESC and inputs to the PCEB.

- SDCPYEN[13,03:01]#
- SDCPYUP
- SDOE[2:0]#
- SDLE[3:0]#

Copy Enable Outputs (SDCPYEN[13,3:1]#)

These signals enable the byte copy operations between data byte lanes 0, 1, 2 and 3 as shown in the Table 35. ISA master cycles do not perform assembly/disassembly operations. Thus, these cycles use SDCPYEN[13,03:01]# to perform the byte routing and byte copying between lanes. EISA master cycles however, can have assembly/disassembly operations. These cycles use SDCPYEN[13,03:01]# in conjunction with SDCPYUP and SDLE[3:0]#.

Table 35. Byte Copy Operations

Signal	Copy Between Byte Lanes
SDCPYEN01#	Byte 0 (bits[7:0]) and Byte 1 (bits[15:8])
SDCPYEN02#	Byte 0 (bits[7:0]) and Byte 2 (bits[23:16])
SDCPYEN03#	Byte 0 (bits[7:0]) and Byte 3 (bits[31:24])
SDCPYEN13#	Byte 1 (bits[15:8]) and Byte 3 (bits[31:24])

System Data Copy Up (SDCPYUP)

SDCPYUP controls the direction of the byte copy operations. When SDCPYUP is asserted (high), active lower bytes are copied onto the higher bytes. The direction is reversed when SDCPYUP is negated (low).

System Data Output Enable (SDOE[2:0]#)

These signals enable the output of the data swap buffers onto the EISA Bus (Table 36). SDOE[2:0] are re-drive signals in case of mis-matched cycles between EISA to EISA, EISA to ISA, ISA to ISA and the DMA cycles between the devices on EISA.

Table 36. Output Enable Operations

Signal	Byte Lane
SDOE0#	Applies to Byte 0 (bits[7:0])
SDOE1#	Applies to Byte 1 (bits[15:8])
SDOE2#	Applies to Byte 2 and Byte 3 (bits[31:16])

System Data to Internal (PCEB) Data Latch Enables (SDLE[3:0]#)

These signals latch the data from the EISA Bus into the data swap latches. The data is then either sent to the PCI Bus via the PCEB or re-driven onto the EISA Bus. SDLE[3:0]# latch the data from the corresponding EISA Bus byte lanes during PCI reads from EISA, EISA writes to PCI, DMA cycles between an EISA device and the PCEB. These signals also latch data during mismatched cycles between EISA to EISA, EISA to ISA, ISA to ISA, the DMA cycles between the devices on EISA, and any cycles that require copying of bytes, as opposed to copying and assembly/disassembly.

11.5 Interrupt Acknowledge Control

PEREQ#/INTA# (PCI to EISA Request or Interrupt Acknowledge) is a dual function signal and the selected function depends on the status of EISAHLDA. When EISAHLDA is negated, this signal is an interrupt acknowledge (INTA#) and supports interrupt processing. If interrupt acknowledge is enabled via the PCEB's PCICON Register and EISAHOLDA is negated, the PCEB asserts PEREQ#/INTA# when a PCI interrupt acknowledge cycle is being serviced. This informs the ESC that the forwarded EISA I/O read from location 04h is an interrupt acknowledge cycle. Thus, the ESC uses this signal to distinguish between a request for the interrupt vector and a read of the ESC's DMA register located at 04h. The ESC responds to the read request by placing the interrupt vector on SD[7:0].

12.0 INTEGRATED SUPPORT LOGIC

The ESC integrates support logic for assorted functions for a typical EISA system board. The following functions are directly supported by the ESC.

- EISA Address Buffer Control
- Coprocessor Interface
- BIOS Interface
- Keyboard Controller Interface
- Real Time Clock Interface
- Floppy Disk Controller Interface
- Configuration RAM Interface
- X-Bus and IDE Decode

NOTE:
The ESC directly supports X-Bus Floppy Disk Controller and IDE. If the IDE resides on another bus (e.g., ISA or PCI Bus), additional hardware (to modify the X-Bus signals) is required to support the X-Bus Floppy Disk Controller.

12.1 EISA Address Buffer Control

The EISA Bus consists of unlatched addresses (LA[31:2]) and latched addresses (SA[19:2]). EISA devices generate or monitor LA addresses, and ISA devices generate or monitor SA addresses. Three Discrete F543s are used to generate the SA address from LA and LA addresses from SA addresses (Figure 28). The ESC generates the control signals SALE#, LASAOE#, and SALAOE# for the F543s. These signals control the direction of the address flow. For EISA master, DMA, and Refresh cycles the, the LA addresses are generated by the master device, and the SA addresses are driven by the F543s. For ISA master devices, the SA addresses are generated by the master device, and the LA addresses by driven by the F543s.

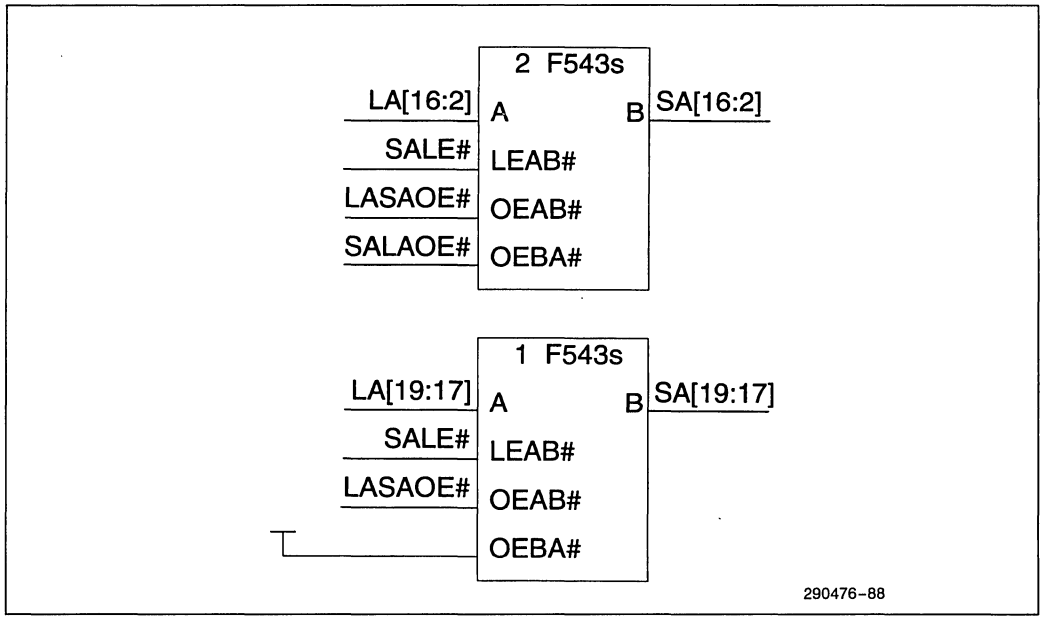


Figure 28. EISA Address Buffers

Table 37. EISA Address Buffer Control Function

Signal		Cycle	Type	
	EISA Master	**ISA Master**	**DMA**	**Refresh**
SALE#	Pulses	Low	Pulses	Pulses
LASAOE#	Low	High	Low	Low
SALAOE#	High	Low	High	High

12.2 Coprocessor Interface

The numeric coprocessor interface is designed to support PC/AT compatible numeric coprocessor exception handling. The EISA Clock Divider configuration register bit 5 needs to be set to a 1 in order to enable the coprocessor error support in the ESC. The coprocessor interface consists of FERR# signal and IGNNE# signal. The FERR# signal and IGNNE# signals are connected directly to the Floating Point Error pin and Ignore Floating Point Error pin of the CPU respectively.

Whenever an error during computation is detected, the CPU asserts the FERR# signal to the ESC. The ESC internally generates an interrupt on the IRQ13 line of the integrated Interrupt Controller. The result is a asserted INTR signal to the CPU.

When the ESC detects an I/O write to the internal port 00F0h, the ESC deasserts the internal IRQ13 line to the integrated Interrupt Controller. At the same time the ESC asserts the IGNNE# signal. The ESC keeps the IGNNE# signal asserted until the FERR# signal is negated by the CPU.

If the coprocessor error support is enabled in the EISA Clock Divider configuration register then the ESC IRQ13 pins cannot be used, and this pin should be tied to ground.

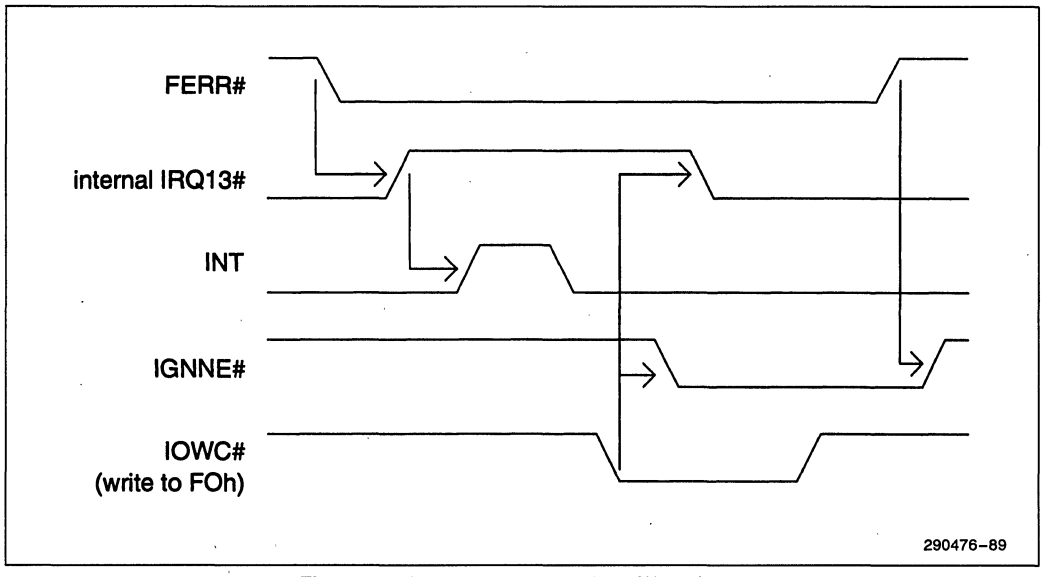


Figure 29. Coprocessor Interface Waveform

12.3 BIOS Interface

The ESC supports a total of 512 KBytes of BIOS memory. The ESC asserts the LBIOSCS# signal for EISA or ISA memory cycles decoded to be in the BIOS space. The 512 KBytes of BIOS includes the conventional 128 KBytes of BIOS and 384 KBytes of enlarged BIOS. The 128 KBytes of conventional BIOS is divided into multiple regions. Each region can be independently enabled or disabled by setting the appropriate bits in the BIOS Chip Select A register and BIOS Chip Select B register. The 128 KBytes of conventional BIOS is also aliased at different locations within the memory space. Refer to Section 4.1, BIOS Memory Space, for details.

The ESC generates the LBIOSCS# signal by internally latching the output of the BIOS address decode with BALE signal. The ESC asserts the LBIOSCS# for all read cycles in the enabled BIOS memory space. The ESC will assert LBIOSCS# signal for write cycles in the enabled BIOS memory space only if the BIOS Chip Select B register bit 3 is set to 1 (BIOS write enable).

12.4 Keyboard Controller Interface

The ESC provides a complete interface to a glueless interface to a 8x42 Keyboard Controller. The ESC Keyboard Controller interface consists of Keyboard Controller Chip Select (KYBDCS#) signal, Mouse interrupt (ABFULL) signal. The ESC also supports the fast Keyboard commands for CPU reset (ALTRST#) and address A20 enable (ALTA20) by integrating Port 92h.

The ESC asserts the KYBDCS# signal for I/O cycles to addresses 60h (82374EB/SB), 62h (82374EB only), 64h (82374EB/SB), and 66h (82374EB only) if the Peripheral Chip Select A register bit 1 is set to 1. The ESC uses the ABFULL signal to internally generate an interrupt request to the integrated Interrupt Controller on the IRQ12 line if EISA Clock Divisor register bit 4 is set to 1 (Mouse Interrupt Enable). A low to high transition on the ABFULL signal is internally latched by the ESC. The high level on this latch remains until a write to I/O port 60h is detected or the ESC is reset.

The ALTRST# is used to reset the CPU under software control. The ESC ALTRST# signal needs to be AND'ed externally with the reset signal from the keyboard controller. A write to the System Control Register (092h) bit 0 to set the bit to a 1 from a 0 causes the ESC to pulse the ALTRST# signal. ALTRST# is asserted for approximately 4 BCLKs. The ESC will not pulse the ALTRST# signal if bit 0 has previously been set to a 1.

12.5 Real Time Clock

The ESC provides a glueless interface for the Real Time Clock in the system. The ESC provides a Real Time Clock Address Latch Enable signal (RTCALE), a Real Time Clock read Strobe(RTCRD#), and a Real Time Clock Write strobe (RTCWR#). The ESC pulses the RTCALE signal asserted for one and a half BCLKs when an I/O write to address 70h is detected. The ESC asserts RTCRD# signal and RTCWR# signal for I/O read and write accesses to address 71h respectively.

The ESC also supports the power on password protection through the Real Time Clock. The power on password protection is enabled by setting the System Control register 092h bit 3 to a 1. The ESC does not assert RTCRD# signal or RTCWR# signal for I/O cycles to 71h if the access are addressed to Real Time Clock addresses (write to 70h) 36h to 3Fh if the power on password protection is enabled.

12.6 Floppy Disk Control Interface

The ESC supports interface to the 82077(SL) floppy disk controller chip. The ESC provides a Floppy Disk Controller Chip Select signal (FDCCS#). The ESC also provides a buffered Drive Interface (DSKCHG#) signal. In addition, the ESC generates the control for the disk light.

The ESC supports both the primary address range (03F0h–03F7h) and secondary address range (0370h–0377h) of the Floppy Disk Controller. The state of Peripheral Chip Select A register bit 5 determines which address range is decoded by the ESC as access to Floppy Disk Controller. If bit 5 is set to 0, the ESC will decode the primary Floppy Disk Controller address range. If bit 5 is set to 1, the ESC will decode the secondary Floppy Disk Controller address range.

The ESC supports the Drive Interface signal. During I/O accesses to address 03F7h (primary) or 0377h (secondary), the ESC drives the inverted state of the DSKCHG# signal on to the SD7 data line. The ESC uses the DSKCHG# signal to determine if the Floppy Disk Controller is present on the X-Bus. If the DSKCHG# signal is samples low during reset, the ESC will disable Floppy Disk Controller support.

The ESC also supports the Disk Light function by generating the DLIGHT# signal. If System Control 092h register bit 6 or bit 7 is set to a 1, the ESC will assert the DLIGHT# signal.

12.7 Configuration RAM Interface

The ESC provides the control signals for 8 Kbytes of external configuration RAM. The configuration RAM is used for storing EISA configuration system parameters. The configuration RAM is I/O mapped between location 0800h–08FFh. Due to the I/O address constraint (256 byte addresses for 8 Kbyte of RAM) , the configuration RAM is organized in 32 pages of 256 bytes each. The I/O port 0C00h is used to store the configuration RAM page address. The ESC integrates this port as Configuration RAM Page register. During a read or a write to the configuration RAM address space 0800h–08FFh, the ESC drives the configuration RAM page address by placing the content of the Configuration RAM Page Address register bits[4:0] on the EISA Address line LA[31:27]#. The ESC will also assert the CRAMRD# signal or the CRAMWR# signal for I/O read and write accesses to I/O address 0800h–08FFh. The ESC will only generate the configuration RAM page address and assert the CRAMRD# signal and CRAMWR# signal if the Peripheral Chip Select B register bit 7 is set to 1.

12.8 General Purpose Peripherals, IDE, Parallel Port, and Serial Port Interface

The ESC provides three dual function pins (GPCS[2:0]#.ECS[2:0]). The functionality of these pins is selected through the configuration Mode Select register bit 4. If Mode Select register bit 4 is set to 0 the general purpose chip select functionality is selected. If Mode Select register bit 4 is set to 1, the encoded chip select functionality is selected.

In general purpose chip select mode, the ESC generates three general purpose chip selects (GPCS[2:0]#). The decode for each general purpose chip selects is programmed through a set of three configuration registers; General Purpose Chip Select x Base Low Address register, General Purpose Chip Select x Base High Address register, and General Purpose Chip Select x Mask register. Each General Purpose Peripheral can be mapped anywhere in the 64 Kbytes of I/O address. The general purpose peripheral address range is programmable from 1 byte to 256 bytes with 2^n granularity.

In encoded chip select mode (ESC[2:0]), in addition to decoding the general purpose chip select 0 address and general purpose chip select 1 address, the ESC also decodes IDE, Parallel Ports, and Serial Ports addresses. The encoded chip select mode requires an external decoder like a F138 to generate the device chip selects from the ESC[2:0] signals.

The ESC generates encoded chip selects for two Serial Ports, COMACS# (ECS[2:0] = 000) and COMBCS# (ESC[2:0] = 001). The ESC supports Serial Port COM1 and Serial Port COM2. Accesses to Serial Port COM1 or Serial Port COM2 are individually programmed through Peripheral Chip Select B register bits[0:3] to generate a encoded chip select for COMACS# or COMBCS#.

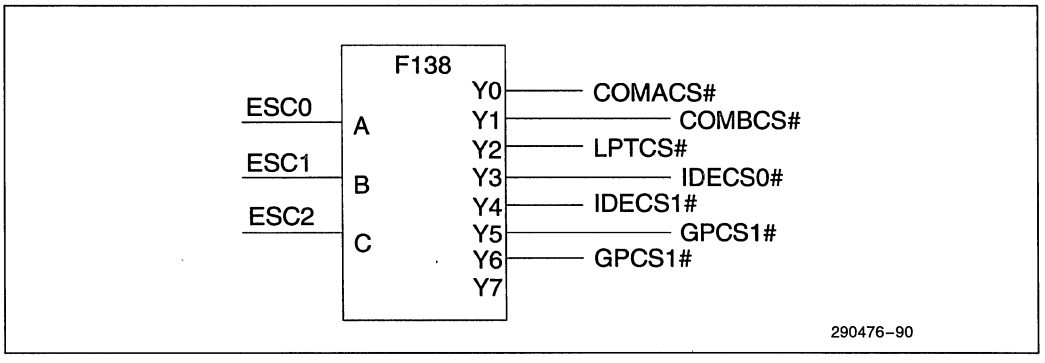


Figure 30. Encoded Chip Select Decoder Logic

Table 38. Encoded Chip Select Decode

ESC2	ESC1	ESC0	PERIPHERAL CS
0	0	0	COMACS#
0	0	1	COMBCS#
0	1	0	LPTCS#
0	1	1	IDECS0#
1	0	0	IDECS1#
1	0	1	GPCS0#
1	1	0	GPCS1#
1	1	1	idle state

NOTE:
Refer to Section 4.5 for the address decode of the peripheral chip selects.

12.9 X-Bus Control And General Purpose Decode

The X-Bus is a secondary data bus buffered from the EISA Bus. The X-Bus is used to interface with peripheral devices that do not require a high speed interface. Typically a discrete buffer device like a F245 is used to buffer the EISA Bus from the X-Bus. The ESC provides two control signals, XBUST/R# and XBUSOE#, for the discrete F245 buffer.

NOTE:
The ESC directly supports X-Bus floppy disk controller and X-Bus IDE hard disk controller. If IDE resides elsewhere (e.g., ISA Bus or PCI Bus), additional hardware is required to support the X-Bus floppy. The additional hardware is used to modify the X-Bus control signals. Figure 31 shows the logic needed to support an X-Bus floppy disk with IDE on the ISA or PCI bus.

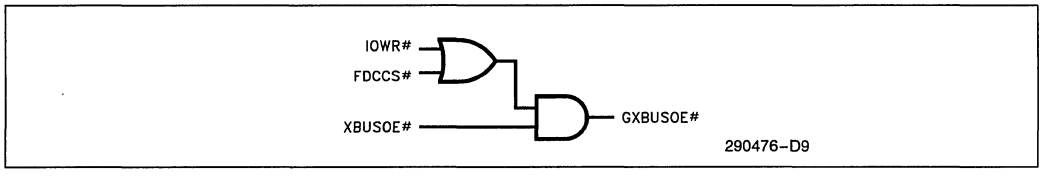


Figure 31

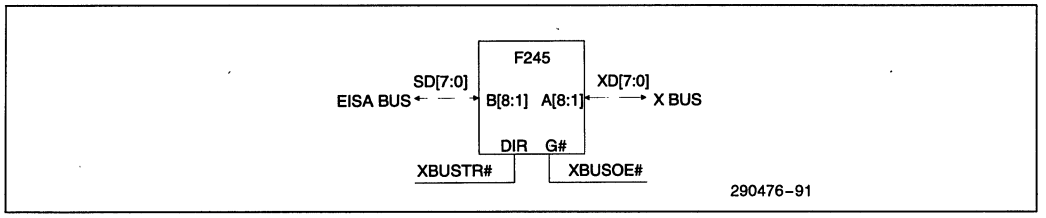


Figure 32. X-Bus Data Buffer

The XBUST/R# signal controls the direction of the data flow of the F245. When the XBUST/R# signal is high, the data direction of the F245 buffer is from the XD[7:0] bus to the SD[7:0] bus. The ESC drives the XBUST/R# signal high during EISA master I/O read cycles, ISA master I/O read cycles, DMA write cycles (write to memory), and memory read cycles decoded to be in the X-Bus BIOS address space. The ESC also drives the XBUST/R# signal high for DMA reads (reads from memory/writes to I/O) from the X-Bus BIOS address space. The X-Bus BIOS address space is defined as the enabled regions and enabled aliases of the BIOS memory space. See Section 4.1, BIOS Memory Space, for detailed description of the BIOS memory map and the configuration bits.

The XBUSOE# signal controls outputs of the F245. When the XBUSOE# signal is asserted, the F245 drives its A buffers or B buffers depending on the state of the XBUST/R# signal. The ESC asserts the XBUSOE# signal for I/O cycles decoded to be in the address range of the peripherals supported by the ESC if these peripherals are enabled in the Peripheral Chip Select A register and Peripheral Chip Select B register.

13.0 POWER MANAGEMENT (82374SB)

The ESC has extensive power management capability permitting a system to operate in a low power state without being powered down. In a typical desktop personal computer there are two states—Power On and Power-Off. Leaving a system powered on when not in use wastes power. The ESC provides a Fast On/Off feature that creates a third state called Fast Off (Figure 33). When in the Fast Off state, the system consumes less power than the Power-On state.

The ESC's power management architecture is based on three functionsSystem Management Mode (SMM), Clock Control, and Advanced Power Management (APM). Software (called SMM code) controls the transitions between the Power On state and the Fast Off state. The ESC invokes this software by generating an SMI to the CPU (asserting the SMI# signal). A variety of programmable events are provided that can generate an SMI. The SMM code places the system in either the Power On state or the Fast Off state.

A Fast On event is an event that instructs the computer (via an SMI to the CPU) to enter the Power-On state in anticipation of system activity by the user. Fast On events are programmable and include moving the mouse, pressing a key on the keyboard, an external hardware event, an incoming call to a system FAX/Modem, a RTC alarm, or the operating system.

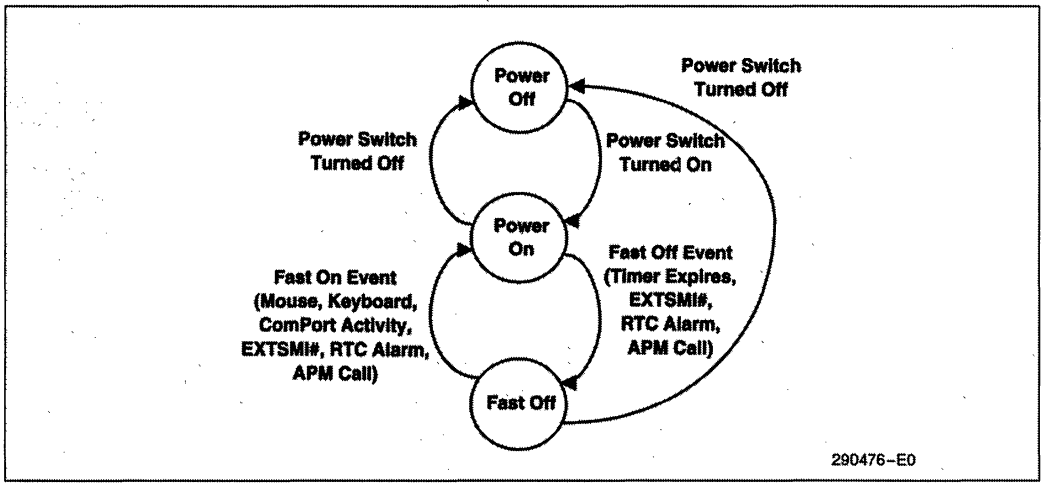


Figure 33. Fast On/Off Flow

13.1 SMM Mode

SMM mode is invoked by asserting the SMI# signal to the CPU. The ESC provides a variety of programmable events that can generate an SMI. When the CPU receives an SMI, it enters SMM mode and executes SMM code out of SMRAM. The SMM code places the system in either the Power On state or the Fast Off state. In the Power On state, the computer system operates normally. In this state one of the four programmable events listed below can trigger an SMI.

1. A global idle timer called the Fast Off timer expires (an indication that the end user has not used the computer for a programmed period of time).
2. The EXTSMI# pin is asserted.
3. An RTC alarm interrupt is detected.
4. The operating system issues an APM call.

13.2 SMI Sources

The SMI# signal can be asserted by hardware interrupt events, the Fast Off Timer, an external SMI event (EXTSMI#), and software events (via the APMC and APMS Registers). Enable/disable bits (in the SMIEN Register) permit each event to be individually masked from generating an SMI. In addition, the SMI# signal can be globally enabled/disabled in the SMICNTL Register. Status of the individual events causing an SMI is provided in the SMIREQ Register. For detailed information on the SMI control/status registers, refer to Section 3.0, Register Description.

Hardware Interrupt Events

Hardware events (IRQ[12,8#,4,3,1] and the Fast Off Timer) are enabled/disabled from generating an SMI in the SMIEN Register. When enabled, the occurrence of the corresponding hardware event generates an SMI (asserts the SMI# signal), regardless of the current power state of the system.

Fast Off Timer

The Fast Off Timer is used to indicate (through an SMI) that the system has been idle for a programmed period of time. The timer counts down from a programmed start value and when the count reaches 00h, can generate an SMI. The timer decrement rate is 1 count every minute and is re-loaded each time a System Event occurs. This counter should NOT be programmed to 00h.

System events are programmable events that can keep the system in the Power On state when there is system activity (Figure 34). These events are indicated by the assertion of IRQ[15:9,8#,7:3,1:0], NMI, or SMI signals. The system event prevents the system from entering the Fast Off state by re-loading the Fast Off Timer.

In addition to system events, *break events cause the system to transition from a Fast Off state to the Power On state. System events (and break events) are enabled/disabled in the SEE Register. When enabled and the associated hardware event occurs (signal is asserted), the Fast Off Timer is re-loaded with its initial count.*

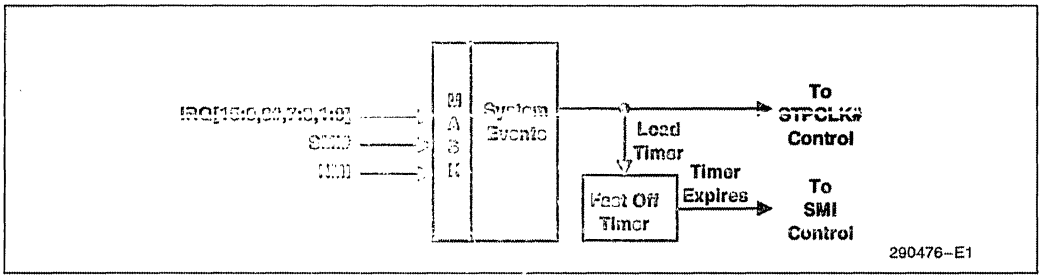


Figure 34. System Events and the Fast Off Timer

EXTSMI#

The EXTSMI# input pin provides the system designer the capability to invoke SMM with external hardware. For example, the EXTSMI# input could be connected to a "green button" permitting the user to enter the Fast Off state by depressing a button. The EXTSMI# generation of an SMI is enabled/disabled in the SMIEN Register.

Software Events

Software events (accessing the APMx Registers) indicate that the OS is passing power management information to the SMI handler. There are two Advanced Power Management (APM) registers—APM Control (APMC) and APM Status (APMS) Registers. These registers permit software to generate an SMI; by writing to the APMC Register. For example, the APMC can be used to pass an APM command between APM OS and BIOS and the APMS Register could be used to pass data between the OS and the SMI handler. For detailed descriptions of these registers, see Section 3.0, Register Description.

The two APM Registers are located in normal I/O space. The PCEB subtractively decodes PCI accesses to these registers and forwards the accesses to the EISA Bus. The APM Registers are not accessible by EISA masters. Note that the remaining power management registers are located in PCI configuration space.

13.3 SMI# And INIT Interaction

The SMI# input to the CPU is an edge sensitive signal. When an S-series processor is reset (INIT asserted), the processor resets the SMI# edge detect logic. After INIT is negated, it takes two clocks before the edge detect circuit can catch an edge. The ESC only asserts SMI# when INIT is negated. If the ESC asserts SMI# and then the INIT signal is sampled asserted, the ESC negates SMI#.

13.3.1 CLOCK CONTROL

The CPU can be put in a low power state by asserting the STPCLK# signal. STPCLK# is an interrupt to the CPU. However, for this type of interrupt, the CPU does not generate an interrupt acknowledge cycle. Once the STPCLK# interrupt is executed, the CPU enters the stop grant state. In this state, the CPU's internal clocks are disabled and instruction execution is stopped. The stop grant state is exited when the STPCLK# signal is negated.

Software can assert STPCLK#, if enabled via the SMICNTL Register, by a read of the APMC Register. Note that STPCLK# can also be periodically asserted by using clock scaling as described below.

The ESC automatically negates STPCLK# when a break event occurs (if enabled in the SEE Register) and the CPU stop grant special cycle has been received. Software can negate STPCLK# by disabling STPCLK# in the SMICNTL Register or by a write to the APMC Register.

NOTE:

1. INIT is always enabled as a break event. Otherwise, INIT acts exactly as other break events:
   - — If STPCLK# is negated when INIT is asserted, the STPCLK high timer is reloaded.
   - — If INIT is asserted when STPCLK# is asserted but before the stop grant bus cycle, STPCLK# negation waits until after the stop grant bus cycle. This happens after the CPU is reset when it samples STPCLK# still asserted.
   - — If INIT is asserted when STPCLK# is asserted and after the stop grant bus cycle, STPCLK# is negated immediately. This guarantees that STPCLK# will be negated after the CPU is reset.
2. While the STPCLK# signal is asserted, the external interrupts (NMI, SMI# and INT) may be asserted to the CPU. If INTR is asserted, it will remain asserted until the CPU INTA cycle is detected. If SMI# (or NMI) is asserted, it remains asserted until the SMI (or NMI) handler clears the ESC's CSMIGATE (or sets the ESC's NMIMASK bit). Thus, SMI#, NMI and INTR can be applied to the CPU independent of the STPCLK# signal state. Note that when SMI#, NMI, and IRQx are enabled as break events, the occurrence of the break event negates STPCLK#.

Clock Scaling (Emulating Clock Division)

Clock scaling permits the ESC to periodically place the CPU in a low power state. This emulates clock division. When clock scaling is enabled, the CPU runs at full frequency for a pre-defined time period and then is stopped for a pre-defined time period. The run/stop time interval ratio emulates the clock division effect from a power/performance point of view. However, clock scaling is more effective than dividing the CPU frequency. For example, if the CPU is in the stop grant state and a break event occurs, the CPU clock returns to full frequency. In addition, there is no recovery time latency to start the clock.

Two programmable 8-bit clock scale timer control registers set the STPCLK# high (negate) and low (assert) times the CTLTMRH and CTLTMRL Registers. The timer is clocked by a 32 µs internal clock. This allows a programmable timer interval for both the STPCLK# high and low times of 0-8 ms. When enabled via the SMICNTL Register, the STPCLK# Timer operates as follows:

- When STPCLK# is negated, the timer is loaded with the value in the CTLTMRH Register and starts counting down. When the timer reaches 00h, STPCLK# is asserted. Since the timer is re-loaded with the contents of the CTLTMRH Register every time STPCLK# is negated (for break events or clock throttling), the STPCLK# minimum inactive time is guaranteed.
- When STPCLK# is asserted, the timer is loaded with the value in the CTLTMRL Register. The timer does not begin to count until the Stop Grant Special Cycle is received. When the timer reaches 00h, STPCLK# is negated. Note that a break event also negates STPCLK#.

NOTE:

If STPCLK# is negated and a break event occurs, the STPCLK# Timer is loaded with the value in the CTLTMRH Register.

13.4 Stop Grant Special Cycle

The Host-Bridge (e.g. PCMC) translates the CPU's stop grant cycle into a PCI special cycle. The PCEB recognizes the stop grant PCI special cycle and asserts STPGNT# low to ESC for one PCICLK. The ESC does not start the STPCLK low timer until STPGNT# is asserted.

During Halt or Autohalt state, the P54C does not respond to STPCLK# assertion with a stop grant cycle. However, during this state an INTR, SMI# or NMI# assertion causes the CPU to exit the halted state, and eventually recognize the STPCLK# assertion with a stop grant cycle. The system design must guarantee that INTR, SMI# and NMI# assertion is not blocked outside of the chipset while STPCLK# is asserted. Otherwise, a potential deadlock situation will exist.

13.5 Dual-Processor Power Management Support

Figure 35 depicts the power management support for dual-processor (DP) or P54CT upgrade processor configuration. The input signals of SMI#, STPCLK#, and NMI of both OEM and upgrade sockets are tied together.

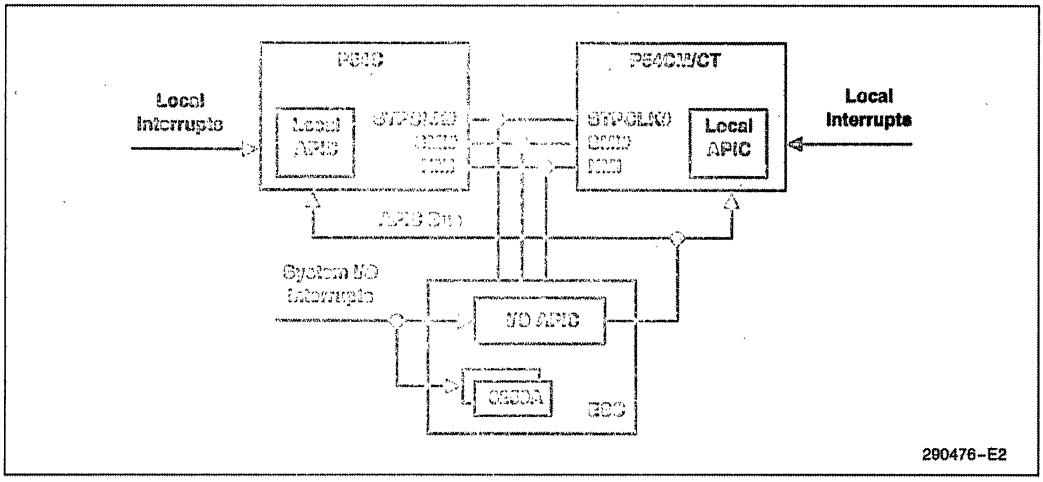


Figure 35. Dual Processor System Configuration

13.5.1 SMI# DELIVERY MECHANISM

For Uni or CT upgrade processor system configuration, SMI# can either be delivered through the ESC SMI# signal or I/O APIC. For the P54C/CM Dual-processor configuration, SMI# should be delivered through I/O APIC only. Ideally, the OS will put the CM processor in Autohalt after the CM processor received a Fast-Off SMI#. The CM processor will wake up if any non-masked system events occur.

13.5.2 STPCLK# TIED TO BOTH SOCKETS

To support a glueless upgrade socket, it is necessary to tie STPCLK# to both sockets. For P54C/CT processor configuration, the P54CT processor will disable P54C and the toggling of STPCLK# has no effect to P54C. For a P54C/CM DP configuration, the toggling of STPCLK# effects both processors (unless the processor is in Autohalt state). Both processors respond with a STPGNT special bus cycle after recognizing STPCLK# low. Both of the STPGNT special bus cycles are passed onto PCI by the PCMC as PCI STPGNT special cycles. The PCI STPGNT PCI cycle causes the PCEB component to assert the STPGNT# signal, depending on how the SCE bit in PCEB is programmed. The ESC recognizes the first STPGNT# assertion, and negates STPCLK# upon the Stop Clock timer expiration or a stop break event.

13.5.3 SMI#/INTR (APIC MODE)

When the APIC is used for interrupt delivery, additional considerations exist regarding ordering. If local interrupts (LINT0/1) are used in APIC mode, then the system can not guarantee an ordering between the local interrupts and any related SMI# events.

In DP mode, interrupts can generally be directed to a specific processor, which may not be the same processor that the SMI# is directed. The IRQ blocking logic in the ESC still operates with APIC delivery mode. Thus, if an IRQ is enabled to cause an SMI# event, it will be blocked until the CSMIGATE is cleared, regardless of where the IRQ or SMI is to be directed by the APIC.

14.0 ELECTRICAL CHARACTERISTICS

14.1 Maximum Ratings

Case Temperature Under Bias ..... −65°C to 110°C
Storage Temperature ..... −65°C to 150°C
Supply Voltages with Respect to Ground ..... −0.5V to V_{CC} + 0.5V
Voltage On Any Pin ..... −0.5V to V_{CC} + 0.5V
Power Dissipation .....0.70W fully loaded
.....0.55W with four slots

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

14.2 NAND Tree

A NAND Tree is provided primarily for VIL/VIH testing. The NAND Tree is also useful for Automated Test Equipment (ATE) at board-level testing. The NAND Tree allows the tester to test the solder connections for each individual signal pin.

The TEST pin, along with IRQ5 and IRQ3, activates the NAND Tree. Asserting TEST causes the output pulse train to appear on the EISAHOLD pin. IRQ5 must be driven high in order to enable the NAND Tree. The assertion of IRQ3 causes the ESC to disable its buffers.

The sequence of the ATE test is as follows:

1. Drive TEST low, IRQ3 high, and IRQ5 high.
2. Drive each pin that is a part of the NAND Tree high. Please note that not every pin is included in the tree. See table below for details.
3. Starting at pin 165 (DLIGHT#) and continuing with pins 167, 168, etc., individually drive each pin low. Expect EISAHOLD to toggle after each corresponding input pin is toggled. The final pin in the tree is pin 100 (EISAHOLD). Not every pin is toggled in sequential order. Please refer to the table for tree ordering. When IRQ3 is driven low, the test mode is exited, and the ESC's buffers will be enabled.
4. Before enabling the ESC's buffers (via IRQ3), turn off tester drivers.
5. Reset the ESC prior to proceeding with further testing.

Table 39. NAND Tree Cell Order (82374EB)

Pin #	Name
165	DLIGHT#[1]
167	FDCCS#
168	RTCWR#
169	RTCRD#
102	AFLUSH#
106	REFRESH#
107	APICCLK
108	APICD1
109	APICD0
110	IOCHK#
111	RSTDRV
112	IRQ9
113	DREQ2
114	NOWS#
115	CHRDY#
116	SMWTC#
121	SMRDC#
122	IOWC#
123	IORC#
125	DREQ3
127	DREQ1
135	IRQ7
136	IRQ6
141	BALE
142	OSC
143	SA1
144	SA0
145	M16#
146	SBHE#
147	IO16#
148	IRQ10

Pin #	Name
149	IRQ11
150	IRQ12
151	IRQ15
152	IRQ14
164	CRAMRD#
166	DSKCHG
171	ABFULL
179	CMD#
180	START#
186	EXRDY
187	EX32#
188	EX16#
189	SLBURST#
190	EOP
191	SPKR
193	IRQ8#
194	IRQ13
195	IRQ1
197	DREQ0
198	MRDC#
200	MWTC#
201	DREQ5
203	DREQ6
205	DREQ7
206	MASTER16#
3	LA31#/CPG4
4	LA30#/CPG3
5	LA29#/CPG2
6	LA28#/CPG1
7	LA27#/CPG0
8	LA26#
10	LA25#

Table 39. NAND Tree Cell Order (82374EB) (Continued)

Pin #	Name
11	LA24#
12	LA16
13	LA15
15	LA14
16	LA13
17	LA12
19	LA11
20	LA10
21	LA9
22	LA8
23	LA7
24	LA6
28	LA5
29	LA4
30	LA23
31	LA3
32	LA22
33	LA2
34	LA21
36	LA20
40	LA19
42	MREQ7#/PIRQ0#
43	LA18
44	MREQ6#/PIRQ1#
45	LA17
46	MREQ5#/PIRQ2#
47	MREQ4#/PIRQ3#
48	MREQ3#

Pin #	Name
49	MREQ2#
50	MREQ1#
51	MREQ0#
55	BE0#
56	BE1#
57	BE2#
58	BE3#
59	SD0
60	SD1
61	SD2
63	SD3
64	SD4
65	SD5
66	SD6
67	SD7
70	W/R#
71	M/IO#
72	MSBURST#
91	FERR#
95	RESET#
96	PERR#
97	SERR#
98	NMFLUSH#
99	PEREQ#/INTA#
138	IRQ4
139	IRQ3(2)
100	EISAHOLD(3)

Table 40. NAND Tree Cell Order (82374SB)

Pin #	Name
165	DLIGHT#(1)
167	FDCCS#
168	RTCWR#
169	RTCRD#
170	RTCALE
171	ABFULL
172	KYBDCS#
173	LBIOSCS#
174	SALAOE#
175	LASAOE#
176	SALE#
102	AFLUSH#
106	REFRESH#
107	APICCLK
108	APICD1
109	APICD0
110	IOCHK#
111	RSTDRV
112	IRQ9
113	DREQ2
114	NOWS#
115	CHRDY#
116	SMWTC#
117	AEN4
118	AEN3
119	AEN2
120	AEN1
121	SMRDC#
122	IOWC#
123	IORC#
124	DACK3#

Pin #	Name
125	DREQ3
126	DACK1#
127	DREQ1
133	STPGNT#
134	AEN#
135	IRQ7
136	IRQ6
140	DACK2#
141	BALE
142	OSC
143	SA1
144	SA0
145	M16#
146	SBHE#
147	IO16#
148	IRQ10
149	IRQ11
150	IRQ12
151	IRQ15
152	IRQ14
155	GPCS0#
159	GPCS1#
160	GPCS2#
161	XBUSOE#
162	XBUST/R#
163	CRAMWR#
164	CRAMRD#
166	DSKCHG
179	CMD#
180	START#
185	EXTSMI#
186	EXRDY

Table 40. NAND Tree Cell Order (82374SB) (Continued)

Pin #	Name
187	EX32#
188	EX16#
189	SLBURST#
190	EOP
191	SPKR
192	SLOWH#
193	IRQ8#
194	IRQ13
195	IRQ1
196	DACK0#
197	DREQ0
198	MRDC#
199	DACK5#
200	MWTC#
201	DREQ5
202	DACK6#
203	DREQ6
204	DACK7#
205	DREQ7
206	MASTER16#
207	MACK3#
3	LA31#/CPG4
4	LA30#/CPG3
5	LA29#/CPG2
6	LA28#/CPG1
7	LA27#/CPG0
8	LA26#
10	LA25#
11	LA24#
12	LA16
13	LA15
15	LA14

Pin #	Name
16	LA13
17	LA12
19	LA11
20	LA10
21	LA9
22	LA8
23	LA7
24	LA6
28	LA5
29	LA4
30	LA23
31	LA3
32	LA22
33	LA2
34	LA21
36	LA20
37	MACK2#
38	MACK1#
40	LA19
41	MACK0#
42	MREQ7#/PIRQ0#
43	LA18
44	MREQ6#/PIRQ1#
45	LA17
46	MREQ5#/PIRQ2#
47	MREQ4#/PIRQ3#
48	MREQ3#
49	MREQ2#
50	MREQ1#
51	MREQ0#
55	BE0#
56	BE1#

Table 40. NAND Tree Cell Order (82374SB) (Continued)

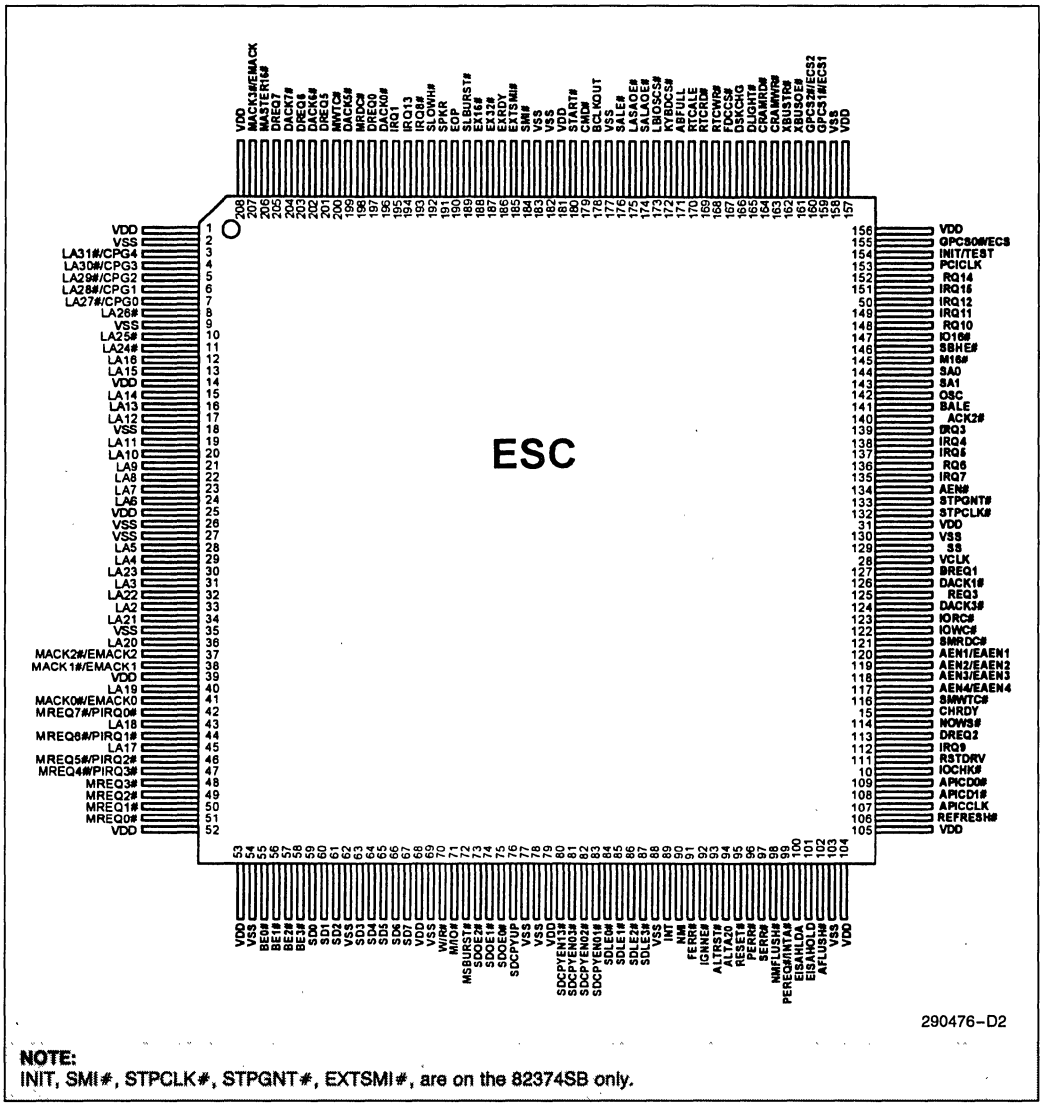
Pin #	Name
57	BE2#
58	BE3#
59	SD0
60	SD1
61	SD2
63	SD3
64	SD4
65	SD5
66	SD6
67	SD7
70	W/R#
71	M/IO#
72	MSBURST#
73	SDOE2#
74	SDOE1#
75	SDOE0#
76	SDCPYUP
80	SDCPYEN13#
81	SDCPYEN03#
82	SDCPYEN02#
83	SDCPYEN01#
84	SDLE0#
85	SDLE1#
86	SDLE2#
87	SDLE3#
91	FERR#
95	RESET#
96	PERR#
97	SERR#
98	NMFLUSH#
99	PEREQ#/INTA#
138	IRQ4
139	IRQ3(2)
100	EISAHOLD(3)

NOTES:
1. First Pin in NAND Tree.
2. Enables ESC's Buffers when 0.
3. Last Pin in NAND Tree.

15.0 PINOUT AND PACKAGE INFORMATION

The ESC package is a 208-pin Plastic Quad Flat Pack (PQFP). The package signals are shown in Figure 36 and listed in Table 41 and Table 42. Note that NC pins require individual pull-up resistors of 8 KΩ–10 KΩ.

15.1 Pinout And Pin Assignment



NOTE:
INIT, SMI#, STPCLK#, STPGNT#, EXTSMI#, are on the 82374SB only.

Figure 36. ESC Package Pinout

Table 41. ESC Alphabetical Pin Assignment

Name	Pin #	Type
ABFULL	171	in
AEN#	134	out
AEN1/EAEN1	120	out
AEN2/EAEN2	119	out
AEN3/EAEN3	118	out
AEN4/EAEN4	117	out
AFLUSH#	102	t/s
ALTA20	94	out
ALTRST#	93	out
APICCLK	107	in
APICD0#	109	od
APICD1#	108	od
BALE	141	out
BCLK	128	in
BCLKOUT	178	out
BE0#	55	t/s
BE1#	56	t/s
BE2#	57	t/s
BE3#	58	t/s
CHRDY	115	o/d
CMD#	179	out
CRAMRD#	164	out
CRAMWR#	163	out
DACK0#	196	out
DACK1#	126	out
DACK2#	140	out
DACK3#	124	out
DACK5#	199	out
DACK6#	202	out
DACK7#	204	out
DLIGHT#	165	out

Name	Pin #	Type
DREQ0	197	in
DREQ1	127	in
DREQ2	113	in
DREQ3	125	in
DREQ5	201	in
DREQ6	203	in
DREQ7	205	in
DSKCHG	166	in
EISAHLDA	100	in
EISAHOLD	101	out
EOP	190	t/s
EX16#	188	o/d
EX32#	187	o/d
EXRDY	186	o/d
EXTSMI# (82374SB)	185	in
FDCCS#	167	out
FERR#	91	in
GPCS0#/ECS0	155	out
GPCS1#/ECS1	159	out
GPCS2#/ECS2	160	out
IGNNE#	92	out
INIT/TEST (82374SB)	154	in
INTR	89	out
IO16#	147	o/d
IOCHK#	110	in
IORC#	123	t/s
IOWC#	122	t/s
IRQ1	195	in
IRQ3	139	in
IRQ4	138	in
IRQ5	137	in

Table 41. ESC Alphabetical Pin Assignment (Continued)

Name	Pin #	Type
IRQ6	136	in
IRQ7	135	in
IRQ8#	193	in
IRQ9	112	in
IRQ10	148	in
IRQ11	149	in
IRQ12	150	in
IRQ13	194	in
IRQ14	152	in
IRQ15	151	in
KYBDCS#	172	out
LA2	33	t/s
LA3	31	t/s
LA4	29	t/s
LA5	28	t/s
LA6	24	t/s
LA7	23	t/s
LA8	22	t/s
LA9	21	t/s
LA10	20	t/s
LA11	19	t/s
LA12	17	t/s
LA13	16	t/s
LA14	15	t/s
LA15	13	t/s
LA16	12	t/s
LA17	45	t/s
LA18	43	t/s
LA19	40	t/s
LA20	36	t/s
LA21	34	t/s
LA22	32	t/s

Name	Pin #	Type
LA23	30	t/s
LA24#	11	t/s
LA25#	10	t/s
LA26#	8	t/s
LA27#/CPG0	7	t/s
LA28#/CPG1	6	t/s
LA29#/CPG2	5	t/s
LA30#/CPG3	4	t/s
LA31#/CPG4	3	t/s
LASAOE#	175	out
LBIOSCS#	173	out
M/IO#	71	t/s
M16#	145	o/d
MACK0#/ EMACK0	41	out
MACK1#/ EMACK1	38	out
MACK2#/ EMACK2	37	out
MACK3#/ EMACK3	207	out
MASTER16#	206	in
MRDC#	198	t/s
MREQ0#	51	in
MREQ1#	50	in
MREQ2#	49	in
MREQ3#	48	in
MREQ4#PIRQ3#	47	in
MREQ5#/PIRQ2#	46	in
MREQ6#/PIRQ1#	44	in
MREQ7#/PIRQ0#	42	in
MSBURST#	72	t/s

1

Table 41. ESC Alphabetical Pin Assignment (Continued)

Name	Pin #	Type
MWTC#	200	t/s
NC (82374EB)	132	—
NC (82374EB)	133	—
NC (82374EB)	184	—
NC (82374EB)	185	—
NMFLUSH#	98	t/s
NMI	90	out
NOWS#	114	o/d
OSC	142	in
PCICLK	153	in
PEREQ#/INTA#	99	in
PERR#	96	in
REFRESH#	106	t/s
RESET#	95	in
RSTDRV	111	out
RTCALE	170	out
RTCRD#	169	out
RTCWR#	168	out
SA0	144	t/s
SA1	143	t/s
SALAOE#	174	out
SALE#	176	out
SBHE#	146	t/s
SD0	59	t/s
SD1	60	t/s
SD2	61	t/s
SD3	63	t/s
SD4	64	t/s
SD5	65	t/s
SD6	66	t/s
SD7	67	t/s
SDCPYEN01#	83	out

Name	Pin #	Type
SDCPYEN02#	82	out
SDCPYEN03#	81	out
SDCPYEN13#	80	out
SDCPYUP	76	out
SDLE0#	84	out
SDLE1#	85	out
SDLE2#	86	out
SDLE3#	87	out
SDOE0#	75	out
SDOE1#	74	out
SDOE2#	73	out
SERR#	97	in
SLBURST#	189	in
SLOWH#	192	out
SMI# (82374SB)	184	out
SMRDC#	121	out
SMWTC#	116	out
SPKR	191	out
START#	180	t/s
STPCLK# (82374SB)	132	out
STPGNT# (82374SB)	133	in
TEST (82374EB)	154	in
V_{DD}	1	V
V_{DD}	14	V
V_{DD}	25	V
V_{DD}	39	V
V_{DD}	52	V
V_{DD}	53	V
V_{DD}	68	V
V_{DD}	79	V
V_{DD}	104	V

Table 41. ESC Alphabetical Pin Assignment (Continued)

Name	Pin #	Type
V_{DD}	105	V
V_{DD}	131	V
V_{DD}	156	V
V_{DD}	157	V
V_{DD}	181	V
V_{DD}	208	V
V_{SS}	2	V
V_{SS}	9	V
V_{SS}	18	V
V_{SS}	26	V
V_{SS}	27	V
V_{SS}	35	V
V_{SS}	54	V
V_{SS}	62	V

Name	Pin #	Type
V_{SS}	69	V
V_{SS}	77	V
V_{SS}	78	V
V_{SS}	88	V
V_{SS}	103	V
V_{SS}	129	V
V_{SS}	130	V
V_{SS}	158	V
V_{SS}	177	V
V_{SS}	182	V
V_{SS}	183	V
W/R#	70	t/s
XBUSOE#	161	out
XBUST/R#	162	out

Table 42. ESC Numerical Pin Assignment

Name	Pin #	Type
1	V_{DD}	V
2	V_{SS}	V
3	LA31#/CPG4	t/s
4	LA30#/CPG3	t/s
5	LA29#/CPG2	t/s
6	LA28#/CPG1	t/s
7	LA27#/CPG0	t/s
8	LA26#	t/s
9	V_{SS}	V
10	LA25#	t/s
11	LA24#	t/s
12	LA16	t/s
13	LA15	t/s
14	V_{DD}	V
15	LA14	t/s
16	LA13	t/s
17	LA12	t/s
18	V_{SS}	V
19	LA11	t/s
20	LA10	t/s
21	LA9	t/s
22	LA8	t/s
23	LA7	t/s
24	LA6	t/s
25	V_{DD}	V
26	V_{SS}	V
27	V_{SS}	V
28	LA5	t/s

Name	Pin #	Type
29	LA4	t/s
30	LA23	t/s
31	LA3	t/s
32	LA22	t/s
33	LA2	t/s
34	LA21	t/s
35	V_{SS}	V
36	LA20	t/s
37	MACK2#/EMACK2	out
38	MACK1#/EMACK1	out
39	V_{DD}	V
40	LA19	t/s
41	MACK0#/EMACK0	out
42	MREQ7#/PIRQ0#	in
43	LA18	t/s
44	MREQ6#/PIRQ1#	in
45	LA17	t/s
46	MREQ5#/PIRQ2#	in
47	MREQ4#/PIRQ3#	in
48	MREQ3#	in
49	MREQ2#	in
50	MREQ1#	in
51	MREQ0#	in
52	V_{DD}	V
53	V_{DD}	V
54	V_{SS}	V
55	BE0#	t/s

Table 42. ESC Numerical Pin Assignment (Continued)

Name	Pin #	Type
56	BE1#	t/s
57	BE2#	t/s
58	BE3#	t/s
59	SD0	t/s
60	SD1	t/s
61	SD2	t/s
62	V_{SS}	V
63	SD3	t/s
64	SD2	t/s
65	SD5	t/s
66	SD6	t/s
67	SD7	t/s
68	V_{DD}	V
69	V_{SS}	V
70	W/R#	t/s
71	M/IO#	t/s
72	MSBURST#	t/s
73	SDOE2#	out
74	SDOE1#	out
75	SDOE0#	out
76	SDCPYUP	out
77	V_{SS}	V
78	V_{SS}	V
79	V_{DD}	V
80	SDCPYEN13#	out
81	SDCPYEN03#	out
82	SDCPYEN02#	out
83	SDCPYEN01#	out
84	SDLE0#	out
85	SDLE1#	out
86	SDLE2#	out
87	SDLE3#	out

Name	Pin #	Type
88	V_{SS}	V
89	INTR	out
90	NMI	out
91	FERR#	in
92	IGNNE#	out
93	ALTRST#	out
94	ALTA20	out
95	RESET#	in
96	PERR#	in
97	SERR#	in
98	NMFLUSH#	t/s
99	PEREQ# / INTA#	in
100	EISAHLDA	in
101	EISAHOLD	out
102	AFLUSH#	t/s
103	V_{SS}	V
104	V_{DD}	V
105	V_{DD}	V
106	REFRESH#	t/s
107	APICCLK	in
108	APICD1#	od
109	APICD0#	od
110	IOCHK#	in
111	RSTDRV	out
112	IRQ9	in
113	DREQ2	in
114	NOWS#	o/d
115	CHRDY	o/d
116	SMWTC#	out
117	AEN4/EAEN4	out
118	AEN3/EAEN3	out
119	AEN2/EAEN2	out

Table 42. ESC Numerical Pin Assignment (Continued)

Name	Pin #	Type
120	AEN1/EAEN1	out
121	SMRDC#	out
122	IOWC#	t/s
123	IORC#	t/s
124	DACK3#	out
125	DREQ3	in
126	DACK1#	out
127	DREQ1	in
128	BCLK	in
129	V_{SS}	V
130	V_{SS}	V
131	V_{DD}	V
132	NC (82374EB) STPCLK# (82374SB)	— out
133	NC (82374EB) STPGNT# (82374SB)	— in
134	AEN#	out
135	IRQ7	in
136	IRQ6	in
137	IRQ5	in
138	IRQ4	in
139	IRQ3	in
140	DACK2#	out
141	BALE	out
142	OSC	in
143	SA1	t/s
144	SA0	t/s
145	M16#	o/d
146	SBHE#	t/s
147	IO16#	o/d
148	IRQ10	in

Name	Pin #	Type
149	IRQ11	in
150	IRQ12	in
151	IRQ15	in
152	IRQ14	in
153	PCICLK	in
154	TEST (82374EB) INIT/TEST (82374SB)	in in
155	GPCS0#/ECS0	out
156	V_{DD}	V
157	V_{DD}	V
158	V_{SS}	V
159	GPCS1#/ECS1	out
160	GPCS2#/ECS2	out
161	XBUSOE#	out
162	XBUST/R#	out
163	CRAMWR#	out
164	CRAMRD#	out
165	DLIGHT#	out
166	DSKCHG	in
167	FDCCS#	out
168	RTCWR#	out
169	RTCRD#	out
170	RTCALE	out
171	ABFULL	in
172	KYBDCS#	out
173	LBIOSCS#	out
174	SALAOE#	out
175	LASAOE#	out
176	SALE#	out
177	V_{SS}	V
178	BCLKOUT	out
179	CMD#	out

Table 42. ESC Numerical Pin Assignment (Continued)

Name	Pin #	Type
180	START#	t/s
181	V_{DD}	V
182	V_{SS}	V
183	V_{SS}	V
184	NC (82374EB) SMI# (82374SB)	— out
185	NC (82374EB) EXTSMI# (82374SB)	— in
186	EXRDY	o/d
187	EX32#	o/d
188	EX16#	o/d
189	SLBURST#	in
190	EOP	t/s
191	SPKR	out
192	SLOWH#	out

Name	Pin #	Type
193	IRQ8#	in
194	IRQ13	in
195	IRQ1	in
196	DACK0#	out
197	DREQ0	in
198	MRDC#	t/s
199	DACK5#	out
200	MWTC#	t/s
201	DREQ5	in
202	DACK6#	out
203	DREQ6	in
204	DACK7#	out
205	DREQ7	in
206	MASTER16#	in
207	MACK3#/EMACK3	out
208	V_{DD}	V

15.2 Package Characteristics

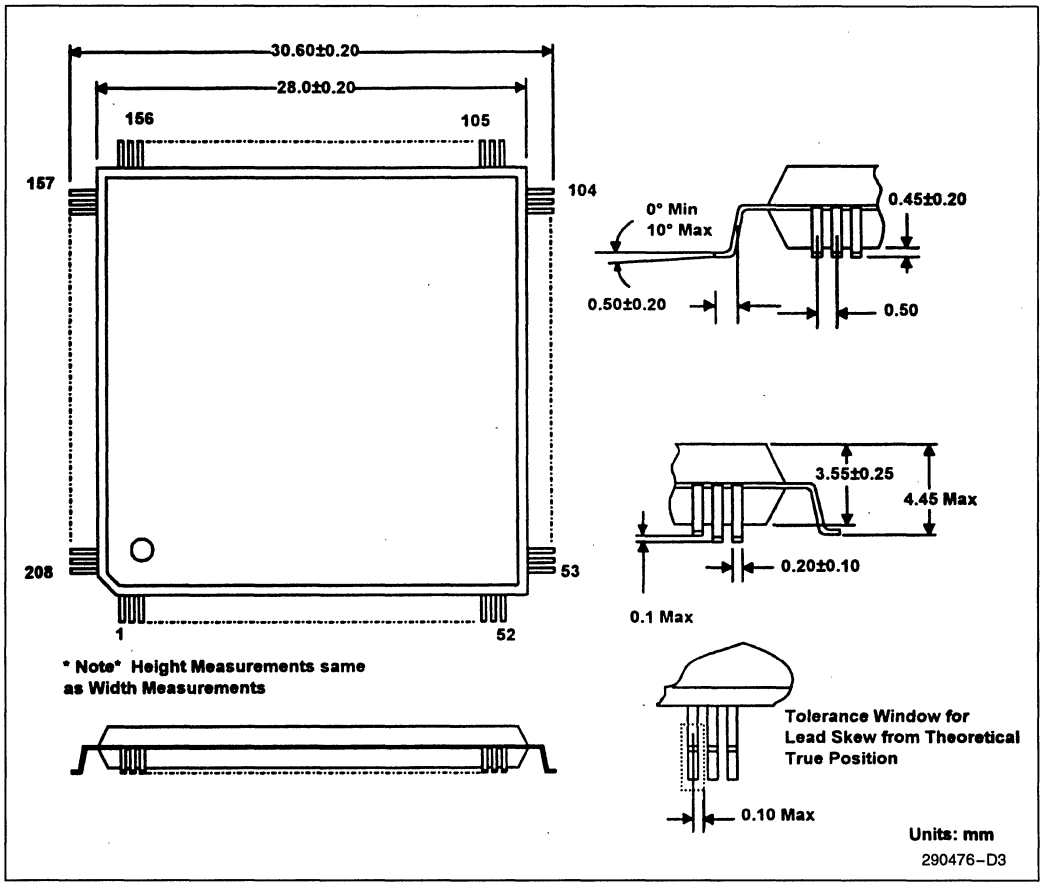


Figure 37. Packaging Dimension Information

82375EB/82375SB PCI-EISA BRIDGE (PCEB)

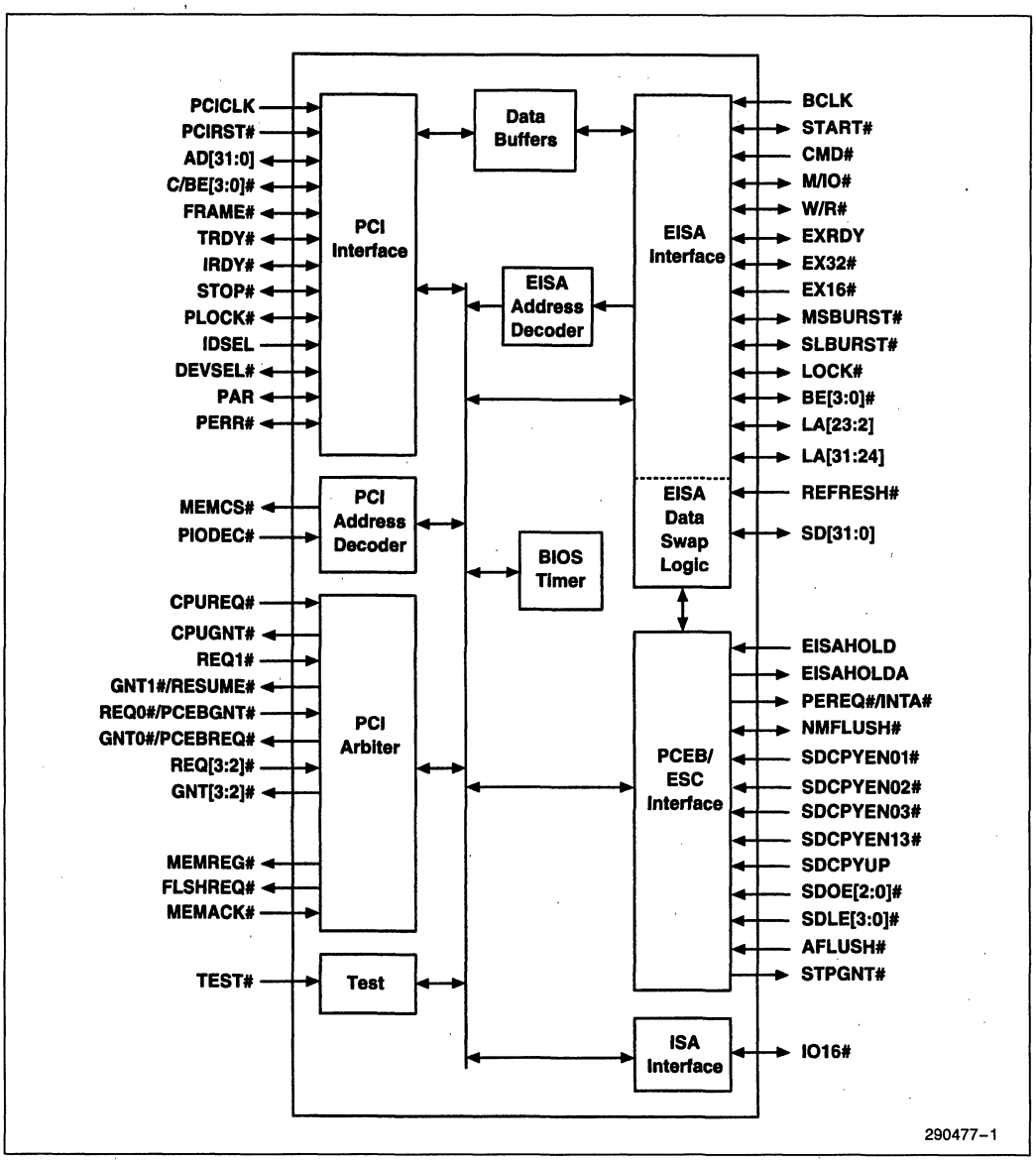
- Provides the Bridge Between the PCI Local Bus and EISA Bus
- 100% PCI and EISA Compatible
  - PCI and EISA Master/Slave Interface
  - Directly Drives 10 PCI Loads and 8 EISA Slots
  - Supports PCI from 25 to 33 MHz
- Data Buffers Improve Performance
  - Four 32-bit PCI-to-EISA Posted Write Buffers
  - Four 16-byte EISA-to-PCI Read/Write Line Buffers
  - EISA-to-PCI Read Prefetch
  - EISA-to-PCI and PCI-to-EISA Write Posting
- Data Buffer Management Ensures Data Coherency
  - Flush Posted Write Buffers
  - Flush or Invalidate Line Buffers
  - System-Wide Data Buffer Coherency Control
- Burst Transfers on both the PCI and EISA Buses
- 32-Bit Data Paths
- Integrated EISA Data Swap Buffers
- Arbitration for PCI Devices
  - Supports Six PCI Masters
  - Fixed, Rotating, or a Combination of the Two
  - Supports External PCI Arbiter and Arbiter Cascading
- PCI and EISA Address Decoding and Mapping
  - Positive Decode of Main Memory Areas (MEMCS# Generation)
  - Four Programmable PCI Memory Space Regions
  - Four Programmable PCI I/O Space Regions
- Programmable Main Memory Address Decoding
  - Main Memory Sizes up to 512 MBytes
  - Access Attributes for 15 Memory Segments in First 1 MByte of Main Memory
  - Programmable Main Memory Hole
- Integrated 16-bit BIOS Timer
- Only Available as Part of a Supported Kit

1

The 82375EB/SB PCI-EISA Bridge (PCEB) provides the master/slave functions on both the PCI Local Bus and the EISA Bus. Functioning as a bridge between the PCI and EISA buses, the PCEB provides the address and data paths, bus controls, and bus protocol translation for PCI-to-EISA and EISA-to-PCI transfers. Extensive data buffering in both directions increases system performance by maximizing PCI and EISA Bus efficiency and allowing concurrency on the two buses. The PCEB's buffer management mechanism ensures data coherency. The PCEB integrates central bus control functions including a programmable bus arbiter for the PCI Bus and EISA data swap buffers for the EISA Bus. Integrated system functions include PCI parity generation, system error reporting, and programmable PCI and EISA memory and I/O address space mapping and decoding. The PCEB also contains a BIOS Timer that can be used to implement timing loops. The PCEB is intended to be used with the EISA System Component (ESC) to provide an EISA I/O subsystem interface.

This document describes both the 82375EB and 82375SB components. Unshaded areas describe the 82375EB. Shaded areas, like this one, describe the 82375SB operations that differ from the 82375EB.

November 1995
Order Number: 290477-004



PCEB Simplified Block Diagram

82375EB/82375SB PCI-EISA BRIDGE (PCEB)

CONTENTS

CONTENTS

CONTENTS

CONTENTS

CONTENTS

1.0 ARCHITECTURAL OVERVIEW

The PCI-EISA bridge chip set provides an I/O subsystem core for the next generation of high-performance personal computers (e.g., those based on the Intel486™ or Pentium® processors). System designers can take advantage of the power of the PCI local bus while maintaining access to the large base of EISA and ISA expansion cards, and corresponding software applications. Extensive buffering and buffer management within the PCI-EISA bridge ensures maximum efficiency in both bus environments.

The chip set consists of two components—the 82375EB PCI-EISA Bridge (PCEB) and the 82374EB EISA System Component (ESC). These components work in tandem to provide an EISA I/O subsystem interface for personal computer platforms based on the PCI standard. This section provides an overview of the PCI and EISA Bus hierarchy followed by an overview of the PCEB and ESC components.

Bus Hierarchy—Concurrent Operations

Figure 1 shows a block diagram of a typical system using the PCI-EISA Bridge chip set. The system contains three levels of buses structured in the following hierarchy:

- Host Bus as the execution bus
- PCI Bus as a primary I/O bus
- EISA Bus as a secondary I/O bus

PCI Bus

The PCI Bus has been defined to address the growing industry needs for a standardized *local bus* that is not directly dependent on the speed and the size of the processor bus. New generations of personal computer system software such as Windows™ and Win-NT™ with sophisticated graphical interfaces, multi-tasking and multi-threading bring new requirements that traditional PC I/O architectures can not satisfy. In addition to the higher bandwidth, reliability and robustness of the I/O subsystem are becoming increasingly important. The PCI environment addresses these needs and provides an upgrade path for the future. PCI features include:

- Processor independent
- Multiplexed, burst mode operation
- Synchronous up to 33 MHz
- 120 MByte/sec usable throughput (132 MByte/sec peak) for 32-bit data path
- 240 MByte/sec usable throughput (264 MByte/sec peak) for 64-bit data path
- Optional 64-bit data path with operations that are transparent with the 32-bit data path
- Low latency random access (60 ns write access latency to slaves from a master parked on the bus)
- Capable of full concurrency with processor/memory subsystem
- Full multi-master capability allowing any PCI master peer-to-peer access to any PCI slave
- Hidden (overlapped) central arbitration
- Low pin count for cost effective component packaging (multiplexed address/data)
- Address and data parity
- Three physical address spaces: memory, I/O, and configuration
- Comprehensive support for autoconfiguration through a defined set of standard configuration functions

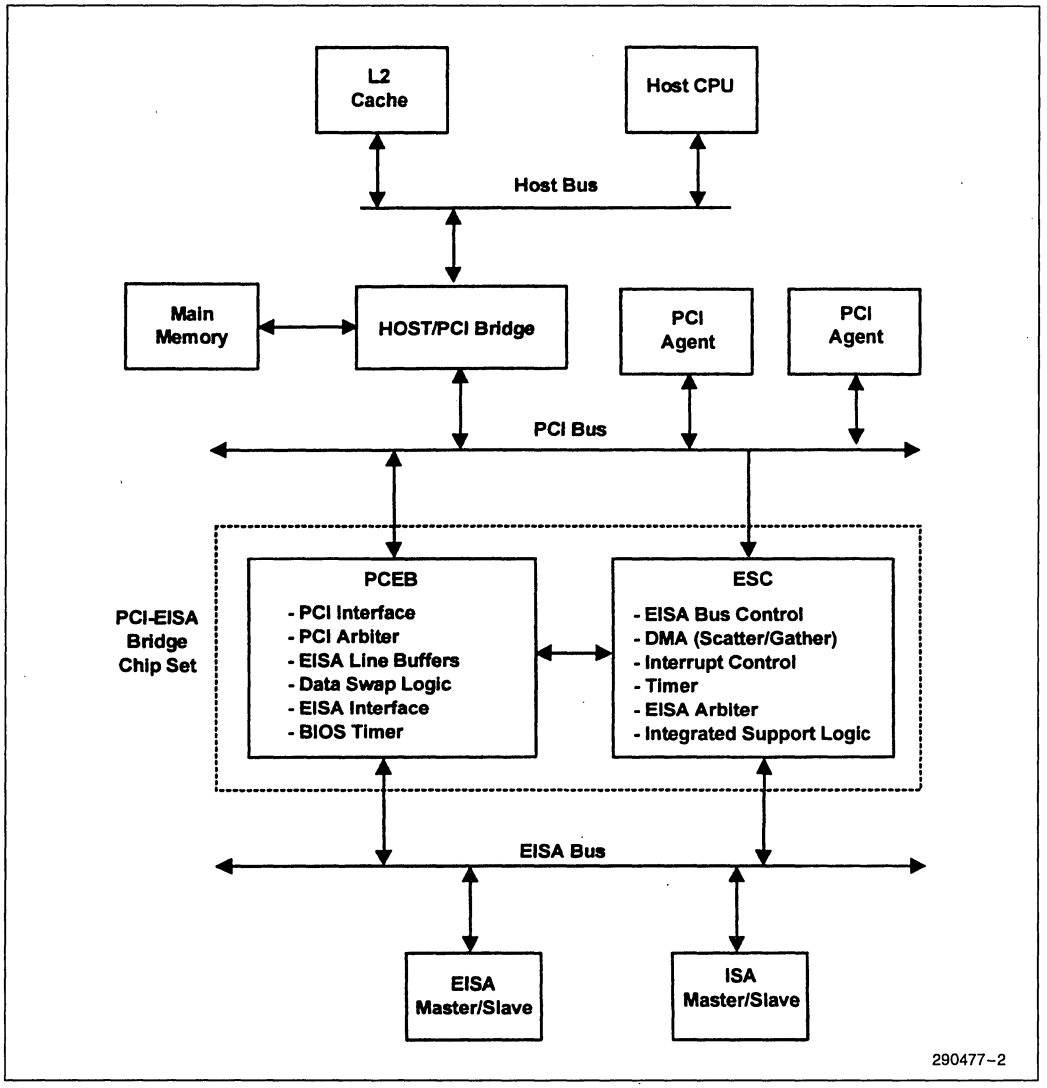


Figure 1. PCI-EISA System Diagram

System partitioning shown in Figure 1 illustrates how the PCI can be used as a common interface between different portions of a system platform that are typically supplied by the chip set vendor. These portions are the Host/PCI Bridge (including a main memory DRAM controller and an optional secondary cache controller) and the PCI-EISA Bridge. Thus, the PCI allows a system I/O core design to be decoupled from the processor/

memory treadmill, enabling the I/O core to provide maximum benefit over multiple generations of processor/memory technology. For this reason, the PCI-EISA Bridge can be used with different processors (i.e. derivatives of the Intel486 CPU or the new generation processors, such as the Pentium processor.) Regardless of the new requirements imposed on the processor side of the Host/PCI Bridge (e.g. 64-bit data path, 3.3V interface, etc.) the PCI side remains unchanged. This standard PCI environment allows reusability, not only of the rest of the platform chip set (i.e. PCI-EISA Bridge), but also of all other I/O functions interfaced at the PCI level. These functions typically include graphics, SCSI, and LAN.

EISA Bus

The EISA bus in the system shown in the Figure 1 represents a second level I/O bus. It allows personal computer platforms built around the PCI as a primary I/O bus to leverage the large EISA/ISA product base. Combinations of PCI and EISA buses, both of which can be used to provide expansion functions, will satisfy even the most demanding applications.

Along with compatibility for 16-bit and 8-bit ISA hardware and software, the EISA bus provides the following key features:

- 32-bit addressing and 32-bit data path
- 33 MByte/sec bus bandwidth
- Multiple bus master support through efficient arbitration
- Support for autoconfiguration

Integrated Bus Central Control Functions

The PCI-EISA Bridge chip set integrates central bus functions on both the PCI and EISA Buses. For PCI, the functions include PCI bus arbitration and the default bus driver. For the EISA Bus, central functions include the EISA Bus controller and EISA arbiter that are integrated in the ESC and EISA data swap buffers that are integrated in the PCEB.

Integrated System Functions

The PCI-EISA Bridge chip set integrates system functions including PCI parity and system error reporting, buffer management, PCI and EISA memory and I/O address space mapping and decoding. For maximum flexibility, all of these functions are programmable allowing for variety of optional features.

1.1 PCEB Overview

The PCEB and ESC form a PCI-EISA Bridge chip set. The PCEB/ESC interface provides the inter-chip communications between these two devices. The major functions provided by the PCEB are described in this section.

PCI Bus Interface

The PCEB can be either a master or slave on the PCI Bus and supports bus frequencies from 25-to-33 MHz. The PCEB becomes a slave when it positively decodes a PCI cycle. The PCEB also becomes a slave for unclaimed cycles on the PCI Bus. These unclaimed cycles are subtractively decoded by the PCEB and forwarded to the EISA Bus. As a slave, the PCEB supports single cycle transfers for memory, I/O, and configuration operations.

For EISA-initiated transfers to the PCI Bus, the PCEB is a PCI master. The PCEB permits EISA devices to access either PCI memory or I/O. While all PCI I/O transfers are single cycle, PCI memory cycles can be either single cycle or burst, depending on the status of the PCEB's Line Buffers. During EISA reads of PCI memory, the PCEB uses a burst read cycle of four Dwords to prefetch data into a Line Buffer. During EISA-to-PCI memory writes, the PCEB uses PCI burst cycles to flush the Line Buffers. The PCEB contains a programmable Master Latency Timer that provides the PCEB with a guaranteed time slice on the PCI Bus, after which it surrenders the bus.

As a master on the PCI Bus, the PCEB generates address and command signals (C/BE[3:0]#), address parity for read and write cycles, and data parity for write cycles. As a slave, the PCEB generates data parity for read cycles. Parity checking is not supported.

The PCEB, as a resource, can be locked by any PCI master. In the context of locked cycles, the entire PCEB subsystem (including the EISA Bus) is considered a single resource.

PCI Bus Arbitration

The PCI arbiter supports six PCI masters—the Host/PCI bridge, PCEB, and four other PCI masters. The arbiter can be programmed for twelve fixed priority schemes, a rotating scheme, or a combination of the fixed and rotating schemes. The arbiter can be programmed for bus parking that permits the Host/PCI Bridge default access to the PCI Bus when no other device is requesting service. The arbiter also contains an efficient PCI retry mechanism to minimize PCI Bus thrashing when the PCEB generates a retry.

EISA Bus Interface

The PCEB contains a fully EISA-compatible master and slave interface. The PCEB directly drives eight EISA slots without external data or address buffering. The PCEB is only a master or slave on the EISA Bus for transfers between the EISA Bus and PCI Bus. For transfers contained to the EISA Bus, the PCEB is never a master or slave. However, the data swap buffers contained in the PCEB are involved in these transfers, if data size translation is needed. The PCEB also provides support for I/O recovery.

EISA/ISA masters and DMA can access PCI memory or I/O. The PCEB only forwards EISA cycles to the PCI Bus if the address of the transfer matches one of the address ranges programmed into the PCEB for EISA-to-PCI positive decode. This includes the main memory segments used for generating MEMCS# from the EISA Bus, one of the four programmable memory regions, or one of the four programmable I/O regions. For EISA-initiated accesses to the PCI Bus, the PCEB is a slave on the EISA Bus. I/O accesses are always non-buffered and memory accesses can be either non-buffered or buffered via the Line Buffers. For buffered accesses, burst cycles are supported.

During PCI-initiated cycles to the EISA Bus, the PCEB is an EISA master. Single cycle transfers are used for I/O and memory read/write cycles from PCI to EISA.

PCI/EISA Address Decoding

The PCEB contains two address decoders—one to decode PCI-initiated cycles and the other to decode EISA-initiated cycles. The two decoders permit the PCI and EISA Buses to operate concurrently.

The PCEB can also be programmed to provide main memory address decoding on behalf of the Host/PCI bridge. When programmed, the PCEB monitors the PCI and EISA bus cycle addresses, and generates a memory chip select signal (MEMCS#) indicating that the current cycle is targeted to main memory residing behind the Host/PCI bridge. Programmable features include, read/write attributes for specific memory segments and the enabling/disabling of a memory hole. If not used, the MEMCS# feature can be disabled.

In addition to the main memory address decoding, there are four programmable memory regions and four programmable I/O regions for EISA-initiated cycles. EISA/ISA master or DMA accesses to one of these regions are forwarded to the PCI Bus.

Data Buffering

The PCEB contains four 16-byte wide Line Buffers for EISA-initiated cycles to the PCI Bus. The Line Buffers permit prefetching of read data from PCI memory and posting of data being written to PCI memory.

By using burst transactions to fill or flush these buffers, when appropriate, the PCEB maximizes bus efficiency. For example, an EISA device could fill a Line Buffer with byte, word, or Dword transfers and the PCEB would use a PCI burst cycle to flush the filled line to PCI memory.

BIOS Timer

The PCEB has a 16-bit BIOS Timer. The timer can be used by BIOS software to implement timing loops. The timer count rate is derived from the EISA clock (BCLK) and has an accuracy of ± 1 μs.

1.2 ESC Overview

The PCEB and ESC form a PCI-EISA bridge. The PCEB/ESC interface provides the inter-chip communications between these two devices. The major functions provided by the ESC are described in this section.

EISA Controller

The ESC incorporates a 32-bit master and an 8-bit slave. The ESC directly drives eight EISA slots without external data or address buffering. EISA system clock (BCLK) generation is integrated by dividing the PCI clock (divide by 3 or divide by 4) and wait state generation is provided. The AENx and MACKx signals provide a direct interface to four EISA slots and supports eight EISA slots with encoded AENx and MACKx signals.

The ESC contains an 8-bit data bus (lower 8 bits of the EISA data bus) that is used to program the ESC's internal registers. Note that for transfers between the PCI and EISA Buses, the PCEB provides the data path. Thus, the ESC does not require a full 32-bit data bus. A full 32-bit address bus is provided and is used during refresh cycles and for DMA operations.

The ESC performs cycle translation between the EISA Bus and ISA Bus. For mis-matched master/slave combinations, the ESC controls the data swap buffers that are located in the PCEB. This control is provided through the PCEB/ESC interface.

DMA Controller

The ESC incorporates the functionality of two 82C37 DMA controllers with seven independently programmable channels. Each channel can be programmed for 8- or 16-bit DMA device size, and ISA-compatible, type "A", type "B", or type "C" timings. Full 32-bit addressing is provided. The DMA controller also generates refresh cycles.

The DMA controller supports an enhanced feature called scatter/gather. This feature provides the capability of transferring multiple buffers between memory and I/O without CPU intervention. In scatter/gather mode, the DMA can read the memory address and word count from an array of buffer descriptors, located in main memory, called the scatter/gather descriptor (SGD) table. This allows the DMA controller to sustain DMA transfers until all of the buffers in the SGD table are handled.

Interrupt Controller

The ESC contains an EISA compatible interrupt controller that incorporates the functionality of two 82C59 Interrupt Controllers. The two interrupt controllers are cascaded providing 14 external and two internal interrupts.

Advanced Programmable Interrupt Controller (APIC)

In addition to the standard EISA compatible interrupt controller described above, the ESC incorporates the Advanced Programmable Interrupt Controller (APIC). While the standard interrupt controller is intended for use in a uni-processor system, APIC can be used in either a uni-processor or multi-processor system. APIC provides multi-processor interrupt management and incorporates both static and dynamic symmetric interrupt distribution across all processors. In systems with multiple I/O subsystems, each subsystem can have its own set of interrupts.

Timer/Counter

The ESC provides two 82C54 compatible timers (Timer 1 and Timer 2). The counters in Timer 1 support the system timer interrupt (IRQ0#), refresh request, and a speaker tone output (SPKR). The counters in Timer 2 support fail-safe time-out functions and the CPU speed control.

Integrated Support Logic

To minimize the chip count for board designs, the ESC incorporates a number of extended features. The ESC provides support for ALTA20 (Fast A20GATE) and ALTRST with I/O Port 92h. The ESC generates the control signals for SA address buffers and X-Bus buffer. The ESC also provides chip selects for BIOS, the keyboard controller, the floppy disk controller, and three general purpose devices. Support for generating chip selects with an external decoder is provided for IDE, a parallel port, and a serial port. The ESC provides support for a PC/AT compatible coprocessor interface and IRQ13 generation.

Power Management

Extensive power management capability permits a system to operate in a low power state without being powered down. Once in the low power state (called "Fast Off" state), the computer appears to be off. For example, the SMM code could turn off the CRT, line printer, hard disk drive's spindle motor, and fans. In addition, the CPU's clock can be governed. To the user, the machine appears to be in the off state. However, the system is actually in an extremely low power state that still permits the CPU to function and maintain communication connections normally associated with today's desktops (e.g., LAN, Modem, or FAX). Programmable options provide power management flexibility. For example, various system events can be programmed to place the system in the low power state or break events can be programmed to wake the system up.

2.0 SIGNAL DESCRIPTION

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

The "#" symbol at the end of a signal name indicates that the active, or asserted state occurs when the signal is at a low voltage level. When "#" is not present after the signal name, the signal is asserted when at the high voltage level.

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

The following notations are used to describe the signal type.

in Input is a standard input-only signal

out Totem Pole output is a standard active driver

o/d Open Drain input/output

t/s Tri-State is a bi-directional, tri-state input/output pin

s/t/s Sustained Tri-State is an active low tri-state signal owned and driven by one and only one agent at a time. The agent that drives a s/t/s pin low must drive it high for at least one clock before letting it float. A new agent can not start driving a s/t/s signal any sooner than one clock after the previous owner tri-states it. An external pull-up is required to sustain the inactive state until another agent drives it and must be provided by the central resource.

2.1 PCI Bus Interface Signals

Pin Name	Type	Description
PCICLK	in	**PCI CLOCK:** PCICLK provides timing for all transactions on the PCI Bus. All other PCI signals are sampled on the rising edge of PCICLK and all timing parameters are defined with respect to this edge. Frequencies supported by the PCEB range from 25 to 33 MHz.
PCIRST#	in	**PCI RESET:** PCIRST# forces the PCEB into a known state. All t/s and s/t/s signals are forced to a high impedance state, and the s/o/d signals are allowed to float high. The PCEB negates all GNT# lines to the PCI Bus and the PCEB negates its internal request. The PCEB drives AD[31:0], C/BE[3:0]#, and PAR during reset to keep these signals from floating (depending on the state of CPUREQ# and REQ1#—as described in the following paragraph). As long as PCIRST# is asserted, the PCEB drives the AD[31:0] signals to keep them from floating. Note that CPUREQ# must be sampled high when PCIRST# is asserted. All PCEB registers are set to their default values. PCIRST# may be asynchronous to PCICLK when asserted or negated. Although asynchronous, the negation of PCIRST# must be a clean, bounce-free edge. PCIRST# must be asserted for a minimum 1 μs, and PCICLK must be active during the last 100 μs of the PCIRST# pulse.
AD[31:0]	t/s	**ADDRESS AND DATA:** AD[31:0] is a multiplexed address and data bus. During the first clock of a transaction, AD[31:0] contain a physical address. During subsequent clocks, AD[31:0] contain data. A PCEB bus transaction consists of an address phase followed by one or more data phases. Little-endian byte ordering is used. AD[7:0] define the least significant byte (LSB) and AD[31:24] the most significant byte (MSB). The information contained in the two low order address bits varies by address space. In the I/O address space, AD[1:0] are used to provide full byte address. In the memory and configuration address space, AD[1:0] are driven "00" during the address phase. The other three encodings are reserved. See Section 5.0, PCI Interface for more details. When the PCEB is a target, AD[31:0] are inputs during the address phase of a transaction. During the following data phase(s), the PCEB may be asked to supply data on AD[31:0] as for a PCI read, or accept data as for a PCI write. As an Initiator, the PCEB drives a valid address on AD[31:0] (with exceptions related to AD[1:0]) during the address phase, and drives write or latches read data on AD[31:0] during the data phase. When PCIRST# is asserted, the PCEB drives the AD[31:0] signals to keep them from floating. In addition, the PCEB acts as the central resource responsible for driving the AD[31:0] signals when no device owns the PCI Bus and the bus is idle.

Pin Name	Type	Description
C/BE[3:0]#	t/s	**BUS COMMAND AND BYTE ENABLES:** The command and byte enable signals are multiplexed on the same PCI pins. During the address phase of a transaction, C/BE[3:0]# define the bus command for bus command definitions. During the data phase, C/BE[3:0]# are used as Byte Enables. The Byte Enables determine which byte lanes carry meaningful data. C/BE[0]# applies to byte 0 and C/BE[3]# to byte 3. C/BE[3:0]# are not used for address decoding. The PCEB drives C/BE[3:0]# as an initiator of a PCI Bus cycle and monitors C/BE[3:0]# as a target. When PCIRST# is asserted, the PCEB drives C/BE[3:0]# to keep them from floating. In addition, the PCEB acts as the central resource responsible for driving the C/BE[3:0]# signals when no device owns the PCI Bus and the bus is idle
FRAME#	s/t/s	**FRAME:** FRAME# is driven by the current initiator to indicate the beginning and duration of an access. FRAME# is asserted to indicate that a bus transaction is beginning. During a transaction, data transfers continue while FRAME# is asserted. When FRAME# is negated, the transaction is in the final data phase. FRAME# is an input when the PCEB is the target. FRAME# is an output when the PCEB is the initiator. During reset, this signal is tri-stated.
TRDY#	s/t/s	**TARGET READY:** TRDY#, as an output, indicates the target's ability to complete the current data phase of the transaction. TRDY# is used in conjunction with IRDY#. A data phase is completed on any clock that both TRDY# and IRDY# are sampled asserted. When PCEB is the target during a read cycle, TRDY# indicates that the PCEB has valid data present on AD[31:0]. During a write, it indicates that the PCEB, as a target, is prepared to latch data. TRDY # is an input to the PCEB when the PCEB is the initiator. During reset, this signal is tri-stated.
IRDY#	s/t/s	**INITIATOR READY:** IRDY#, as an output, indicates the initiator's ability to complete the current data phase of the transaction. IRDY# is used in conjunction with TRDY#. A data phase is completed on any clock that both IRDY# and TRDY# are sampled asserted. When PCEB is the initiator of a write cycle, IRDY# indicates that the PCEB has valid data present on AD[31:0]. During a read, it indicates the PCEB is prepared to latch data. IRDY# is an input to the PCEB when the PCEB is the target. During reset, this signal is tri-stated.
STOP#	s/t/s	**STOP:** As a target, the PCEB asserts STOP# to request that the master stop the current transaction. When the PCEB is an initiator, STOP# is an input. As an initiator, the PCEB stops the current transaction when STOP# is asserted. Different semantics of the STOP# signal are defined in the context of other handshake signals (TRDY# and DEVSEL#). During reset, this signal is tri-stated.
PLOCK#	s/t/s	**PCI LOCK:** PLOCK# indicates an atomic operation that may require multiple transactions to complete. PLOCK# is an input when PCEB is the target and output when PCEB is the initiator. When PLOCK# is sampled negated during the address phase of a transaction, a PCI agent acting as a target will consider itself a locked resource until it samples PLOCK# and FRAME# negated. When other masters attempt accesses to the PCEB (practically to the EISA subsystem) while the PCEB is locked, the PCEB responds with a retry termination. During reset, this signal is tri-stated.

Pin Name	Type	Description
IDSEL	in	**INITIALIZATION DEVICE SELECT:** IDSEL is used as a chip select during configuration read and write transactions. The PCEB samples IDSEL during the address phase of a transaction. If the PCEB samples IDSEL asserted during a configuration read or write, the PCEB responds by asserting DEVSEL# on the next cycle.
DEVSEL#	s/t/s	**DEVICE SELECT:** The PCEB asserts DEVSEL# to claim a PCI transaction as a result of positive or subtractive decode. As an output, the PCEB asserts DEVSEL# when it samples IDSEL asserted during configuration cycles to PCEB configuration registers. As an input, DEVSEL# indicates the response to a PCEB-initiated transaction. The PCEB, when not a master, samples this signal for all PCI transactions to decide whether to subtractively decode the cycle (except for configuration and special cycles). During reset, this signal is tri-stated.
PAR	t/s	**PARITY:** PAR is even parity across AD[31:0] and C/BE[3:0]#. When acting as a master, the PCEB drives PAR during the address and write data phases. As a target, the PCEB drives PAR during read data phases. When PCIRST# is asserted, the PCEB drives the PAR signal to keep it from floating. The PCEB acts as the central resource responsible for driving the PAR signal when no other device is granted the PCI Bus and the bus is idle. Note that the driving and tri-stating of the PAR signal is always one clock delayed from the corresponding driving and tri-stating of the AD[31:0] and C/BE[3:0]# signals.
PERR#	s/t/s	**PARITY ERROR:** PERR# reports data parity errors on all transactions, except special cycles. This signal can only be asserted (by the agent receiving data) two clocks following the data (which is one clock following the PAR signal that covered the data). The duration of PERR# is one clock for each data phase that a data parity error is detected. (If multiple data errors occur during a single transaction the PERR# signal is asserted for more than a single clock.) PERR# must be driven high for one clock before being tri-stated. During reset, this signal is tri-stated.

2.2 PCI Arbiter Signals

Pin Name	Type	Description
CPUREQ#	in	**CPU REQUEST:** CPUREQ# asserted indicates that the Host CPU requests use of the PCI Bus. During PCIRST#, this signal must be sampled high by the PCEB. When PCIRST# is asserted (and CPUREQ# is sampled high), the PCEB drives the AD, C/BE#, and PAR signals to keep them from floating.
REQ[3:0]#	in	**REQUEST:** A bus master asserts the corresponding request signal to request the PCI Bus.
CPUGNT#	out	**CPU GRANT:** The PCEB asserts CPUGNT# to indicate that the CPU master (Host Bridge) has been granted the PCI Bus. During PCI reset, CPUGNT# is tri-stated.
GNT[3:0]#	out	**GRANT:** The PCEB asserts one of the GNT[3:0] signals to indicate that the corresponding PCI master has been granted the PCI Bus. During PCI reset, these signals are tri-stated.
MEMREQ#	out	**MEMORY REQUEST:** If the PCEB is configured in Guaranteed Access Time (GAT) Mode, MEMREQ# is asserted when an EISA device or DMA requests the EISA Bus. The PCEB asserts this signal (along with FLSHREQ#) to indicate that the PCEB requires ownership of main memory. The PCEB asserts FLSHREQ# concurrently with asserting MEMREQ#. This signal is synchronous to the PCI clock. During reset, this signal is driven high. **FLSHREQ# / MEMREQ# / Meaning** 1 / 1 / Idle 0 / 1 / Flush buffers pointing towards PCI to avoid ISA deadlock 1 / 0 / GAT enabled or disabled: For buffer coherency in APIC systems, the buffers pointing to main memory must be flushed and disabled for the duration of assertion. 0 / 0 / GAT mode: Guarantee PCI Bus immediate access to main memory (this may or may not require the PCI-to-main memory buffers to be flushed first, depending on the number of buffers).
FLSHREQ#	out	**FLUSH REQUEST:** FLSHREQ# is asserted by the PCEB to command all of the system's posted write buffers pointing towards PCI to be flushed. This is required before granting the EISA Bus to an EISA master or the DMA. Note that, for APIC related buffer flush requests, this signal is negated. This signal is synchronous to the PCI clock. During reset, this signal is driven high.

Pin Name	Type	Description
MEMACK#	in	**MEMORY ACKNOWLEDGE:** MEMACK# is the response handshake that indicates to the PCEB that the function requested over the MEMREQ# and/or FLSHREQ# signals has been completed. If the PCEB is configured for Guaranteed Access Time Mode through the Arbiter Control Register, and both MEMREQ# and FLSHREQ# are asserted, the assertion of MEMACK# indicates to the PCEB that ownership of main memory has been granted and that all system buffers have been flushed and temporarily disabled. If MEMACK# is asserted in response to assertion of MEMREQ# (GAT either enabled or disabled), it indicates that the system's buffers pointing towards the main memory are flushed and temporarily disabled so that APIC can proceed with the interrupt message sequence. If FLSHREQ# is asserted and MEMREQ# is not asserted (with GAT mode being either enabled or disabled), the assertion of MEMACK# indicates that the system's posted write buffers pointing towards PCI are flushed and temporarily disabled, and the EISA Bus can be granted to an EISA master or DMA. This signal is synchronous to the PCI clock.

2.3 Address Decoder Signals

Pin Name	Type	Description
MEMCS#	out	**MEMORY CHIP SELECT:** MEMCS# is a programmable address decode signal provided to a Host CPU bridge. A Host bridge can use MEMCS# to forward a PCI cycle to the main memory behind the bridge. MEMCS# is asserted one PCI clock after FRAME# is sampled asserted (address phase) and is valid for one clock cycle before being negated. MEMCS# is driven high during reset.
PIODEC#	in	**PCI I/O SPACE DECODER:** PIODEC# can be used to provide arbitrarily complex EISA-to-PCI I/O address space mapping. This signal can be connected to the decode select output of an external I/O address decoder. When PIODEC# is asserted during an EISA I/O cycle, that cycle is forwarded to the PCI Bus. Note that an external pull-up resistor is required if this input signal is not used (i.e., not driven by the external logic).

2.4 EISA Interface Signals

Pin Name	Type	Description
BCLK	in	**BUS CLOCK:** BCLK is the system clock used to synchronize events on the EISA Bus. The ESC device generates BCLK (BCLKOUT), which is a divided down clock from a PCICLK. BCLK runs at a frequency that is dependent on PCICLK and a selected division factor (within the ESC). For example, a 25 MHz PCICLK and a division factor of 3 results in an 8.33 MHz BCLK.
START#	t/s	**START:** START# provides timing control at the start of the cycle and remains asserted for one BCLK period. When the PCEB is an EISA master, START# is an output signal. START# is asserted after LA[31:24]#, LA[23:2] and M/IO# become valid. START# is negated on the rising edge of the BCLK, one BCLK after it was asserted. The trailing edge of START# is always delayed from the rising edge of BCLK. When the PCEB is an EISA master, for cycles to a mismatched slave (see note at the end of this section), START# becomes an input signal at the end of the first START# phase and remains an input until the negation of the last CMD#. The ESC gains the control of the transfer and generates START#. When the PCEB is an EISA slave, START# is an input signal. It is sampled on the rising edge of BCLK. Upon PCIRST#, this signal is tri-stated and placed in output mode.
CMD#	in	**COMMAND:** CMD# provides timing control within the cycle. In all cases, CMD# is an input to the PCEB from the ESC. CMD# is asserted from the rising edge of BCLK, simultaneously with the negation of START#, and remains asserted until the end of the cycle.
M/IO#	t/s	**MEMORY OR I/O:** M/IO# identifies the current cycle as a memory or an I/O cycle. M/IO# is pipelined from one cycle to the next and must be latched by the slave. M/IO# = 1 indicates a memory cycle and M/IO# = 0 indicates an I/O cycle. When the PCEB is an EISA master, the M/IO# is an output signal. When the PCEB is an EISA slave, M/IO# is an input signal. The PCEB responds as an EISA slave for both memory and I/O cycles. Upon PCIRST#, this signal is tri-stated and is placed in output mode.
W/R#	t/s	**WRITE OR READ:** W/R# identifies the cycle as a write or a read cycle. The W/R# signal is pipelined from one cycle to the next and must be latched by the slave. W/R# = 1 indicates a write cycle and W/R# = 0 indicates a read cycle. When the PCEB is an EISA master, W/R# is an output signal. When the PCEB is an EISA slave, W/R# is an input signal. Upon PCIRST#, this signal is tri-stated and placed in output mode.

Pin Name	Type	Description
EXRDY	od	**EISA READY:** EXRDY is used by EISA I/O and memory slaves to request wait states during a cycle. Each wait state is a BCLK period. The PCEB, as an EISA master or slave, samples EXRDY. As an input, the EXRDY is sampled on the falling edge of BCLK after the CMD# has been asserted, and if inactive, each falling edge thereafter. When PCEB is an EISA slave, it may drive EXRDY low to introduce wait states. During reset, this signal is not driven.
EX32#	od	**EISA 32 BIT:** EX32# is used by the EISA slaves to indicate support of 32 bit transfers. When the PCEB is an EISA master, it samples EX32# on the same rising edge of BCLK that START# is negated. During mismatched cycles (see note at the end of this section), EX32# (and EX16#) is used to transfer the control back to the PCEB. EX32# (along with EX16#) is asserted by the ESC on the falling edge of BCLK before the rising edge of the BCLK when the last CMD# is negated. This indicates that the cycle control is transferred back to the PCEB. As an EISA slave, the PCEB always drives EX32# to indicate 32 bit support for EISA cycles. During reset, this signal is not driven.
EX16#	in	**EISA 16 BIT:** EX16# is used by the EISA slaves to indicate their support of 16 bit transfers. As an EISA master, the PCEB samples EX16# on the same rising edge of BCLK that START# is negated. During mismatched cycles (see note at the end of this section), EX16# (and EX32#) is used to transfer the control back to the PCEB. EX16# (along with EX32#) is asserted by the ESC on the falling edge of the BCLK before the rising edge of the BCLK when the last CMD# is negated. This indicates that the cycle control is transferred back to the PCEB. As an EISA slave, the PCEB never asserts EX16#.
MSBURST#	t/s	**MASTER BURST:** MSBURST# is an output when the PCEB is an EISA master and an input when the PCEB is a slave. As a master, the PCEB asserts MSBURST# to indicate to the slave that the next cycle is a burst cycle. If the PCEB samples SLBURST# asserted on the rising edge of BCLK after START# is asserted, the PCEB asserts MSBURST# on the next BCLK edge and proceeds with the burst cycle. As a slave, the PCEB monitors this signal in response to the PCEB asserting SLBURST#. The EISA master asserts MSBURST# to the PCEB to indicate that the next cycle is a burst cycle. As a slave, the PCEB samples MSBURST# on the rising edge of BCLK after the rising edge of BCLK that CMD# is asserted by the ESC. MSBURST# is sampled on all subsequent rising edges of BCLK until the signal is sampled negated. The burst cycle is terminated on the rising edge of BCLK when MSBURST# is sampled negated, unless EXRDY is sampled negated on the previous falling edge of BCLK. During reset, this signal is tri-stated.

Pin Name	Type	Description
SLBURST#	t/s	**SLAVE BURST:** SLBURST# is an input when the PCEB is an EISA master and an output when the PCEB is a slave. When the PCEB is a master, the slave indicates that it supports burst cycles by asserting SLBURST# to the PCEB. The PCEB samples SLBURST# on the rising edge of BCLK at the end of START# for EISA master cycles. When the PCEB is an EISA slave, this signal is an output. As a slave, the PCEB asserts this signal to the master indicating that the PCEB supports EISA burst cycles. During reset, this signal is tri-stated.
LOCK#	t/s	**LOCK:** When asserted, LOCK# guarantees exclusive memory access. This signal is asserted by the PCEB when the PCI master is running locked cycles to EISA slaves. When asserted, this signal locks the EISA subsystem. LOCK# can also be activated by a device on the EISA Bus. This condition is propagated to the PCI Bus via the PLOCK# signal. During reset, this signal is tri-stated.
BE[3:0]#	t/s	**BYTE ENABLES:** BE[3:0]# identify the specific bytes that are valid during the current EISA Bus cycles. When the PCEB is an EISA master and the cycles are directed to a matched slave (slave supports 32-bit transfers), the BE[3:0]# are outputs from the PCEB. When the cycles are directed to a mis-matched slave (slave does not support 32-bit transfers - see note), the BE[3:0]# are floated one and half BCLKs after START# is asserted. These signals become inputs (driven by the ESC) for the rest of the cycle. BE[3:0]# are pipelined signals and must be latched by the addressed slave. When the PCEB is an EISA/ISA/DMA slave, BE[3:0]# are inputs to the PCEB. Upon PCIRST#, these signals are tri-stated and placed in output mode.
LA[31:24]#, LA[23:2]	t/s	**LATCHABLE ADDRESS:** LA[31:24]# and LA[23:2] are the EISA address signals. When the PCEB is an EISA master, these signals are outputs from the PCEB. These addresses are pipelined and must be latched by the EISA slave. LA[31:24]# and LA[23:2] are valid on the falling edge of START#. Note that the upper address bits are inverted before being driven on LA[31:24]#. The timing for LA[31:24] and LA[23:2] are the same. When the PCEB is an EISA slave, these signals are inputs and are latched by the PCEB. For I/O cycles, the PCEB, as an EISA master, floats LA[31:24]# to allow for ESC's address multiplexing (during I/O cycle to configuration RAM). LA[23:2] are actively driven by the PCEB. For memory cycles, the PCEB as an EISA master, drives the LA address lines. During reset, these signals are tri-stated.

Pin Name	Type	Description
SD[31:0]	t/s	**SYSTEM DATA:** SD[31:0] are bi-directional data lines that transfer data between the PCEB and other EISA devices. Data transfer between EISA and PCI devices use these signals. The data swapping logic in the PCEB ensures that the data is available on the correct byte lanes for any given transfer. During reset, these signals are tri-stated.
REFRESH#	in	**REFRESH:** When asserted, REFRESH# indicates to the PCEB that the current cycle on the EISA Bus is a refresh cycle. It is used by the PCEB decoder to distinguish between EISA memory read cycles and refresh cycles.

NOTE:
Mis-matched Cycles. When the PCEB is an EISA master, cycles to the slaves, other than 32 bits transfers, are considered a mis-matched cycle. For mis-matched cycles, the PCEB backs off the EISA Bus one and half BCLKs after it asserted START# by releasing (floating) START#, BE[3:0]# and the SD[31:0] lines. The ESC device then takes control of the transfer. The ESC controls the transfer until the last transfer. At the end of the last transfer, the control is transferred back to the PCEB. The ESC transfers control back to the PCEB by asserting EX32# and EX16# on the falling edge of BCLK before the rising edge of BCLK when the last CMD# is negated.

2.5 ISA Interface Signals

An ISA interface signal is included to improve the PCEB's handling of I/O cycles on the EISA side of the bridge. This signal permits ISA masters to address PCI I/O slaves using the full 16-bit bus size. The signal also allows the PCEB to identify 8-bit I/O slaves for purposes of generating the correct amount of I/O recovery.

Pin Name	Type	Description
IO16#	o/d	**16-BIT I/O CHIP SELECT:** As an EISA slave, the PCEB asserts IO16# when PIODEC# is asserted or an I/O cycle to PCI is detected. As an EISA master, the PCEB uses IO16# as an input to determine the correct amount of I/O recovery time from the I/O Recovery Time (IORT) Register. This register contains bit-fields that are used to program recovery times for 8-bit and 16-bit I/O. When IO16# is asserted, the recovery time programmed into the 16-bit I/O field (bits [1:0]) , if enabled, is used. When IO16# is negated, the recovery time programmed into the 8-bit I/O field (bits [5:3]), if enabled, is used. This signal must have an external pull-up resistor. During reset, this signal is not driven.

2.6 PCEB/ESC Interface Signals

Pin Name	Type	Description
ARBITRATION AND INTERRUPT ACKNOWLEDGE CONTROL		
EISAHOLD	in	**EISA HOLD:** EISAHOLD is used by the ESC to request control of the EISA Bus from the PCEB. This signal is synchronous to PCICLK and is driven inactive when PCIRST# is asserted.
EISAHLDA	out	**EISA HOLD ACKNOWLEDGE:** The PCEB asserts EISAHLDA to inform the ESC that it has been granted ownership of the EISA Bus. This signal is synchronous to the PCICLK.
PEREQ#/ INTA#	out	**PCI-TO-EISA REQUEST OR INTERRUPT ACKNOWLEDGE:** PEREQ#/INTA# is a dual-function signal. The signal function is determined by the state of EISAHLDA signal. When EISAHLDA is negated, this signal is an interrupt acknowledge (i.e., PEREQ#/ INTA# asserted indicates to the ESC that the current cycle on the EISA is an interrupt acknowledge). When EISAHLDA is asserted, this signal is a PCI-to-EISA request (i.e. PEREQ#/ INTA# asserted indicates to the ESC that the PCEB needs to obtain the ownership of the EISA Bus on behalf of a PCI agent). This signal is synchronous to the PCICLK and it is driven inactive when PCIRST# is asserted.
STPGNT#	out	**STOP GRANT ACKNOWLEDGE:** STPGNT# is asserted when the PCEB receives a STOP GRANT PCI special cycle for one PCICLK period. This signal is only asserted when the PCI AD[31:0] signals equal 00120002h during the first data phase of the PCI special cycle. Data of 00120002h on AD[31:0] in subsequent data phases during a PCI special cycle does not result in the assertion of STPGNT#.

PIN NAME	Type	Description
PCEB BUFFER COHERENCY CONTROL		
NMFLUSH#	t/s	**NEW MASTER FLUSH:** The bi-directional NMFLUSH# signal provides handshake between the PCEB and ESC to control flushing of PCI system buffers on behalf of EISA masters. During an EISA Bus ownership change, before the ESC can grant the bus to the EISA master (or DMA), the ESC must ensure that system buffers are flushed and the buffers pointing towards the EISA subsystem are disabled. The ESC asserts NMFLUSH# for one PCI clock to request system buffer flushing. (After asserting NMFLUSH# for 1 PCI clock, the ESC tri-states NMFLUSH#.) When the PCEB samples NMFLUSH# asserted, it starts immediately to assert NMFLUSH# and begins flushing its internal buffers, if necessary. The PCEB also requests PCI system buffer flushing via the MEMREQ#, FLSHREQ#, and MEMACK# signals. When the PCEB completes its internal buffer flushing and MEMACK# is asserted (indicating that the PCI system buffer flushing is complete), the PCEB negates NMFLUSH# for 1 PCI clock and stops driving it. When the ESC samples NMFLUSH# negated, it grants the EISA Bus to an EISA master (or DMA). The ESC resumes responsibility of the default NMFLUSH# driver and starts driving NMFLUSH# negated until the next time a new EISA master (or DMA) wins arbitration. This signal is synchronous with PCICLK and is negated by the ESC at reset.
AFLUSH#	t/s	**APIC FLUSH:** AFLUSH# is bi-directional signal between the PCEB and ESC that controls system buffer flushing on behalf of the APIC. After a reset the ESC negates AFLUSH# until the APIC is initialized and the first interrupt request is recognized.

Pin Name	Type	Description
DATA SWAP BUFFER CONTROL		
SDCPYEN01# SDCPYEN02# SDCPYEN03# SDCPYEN13#	in	**COPY ENABLE:** These active Low signals perform byte copy operation on the EISA data bus (SD[31:0]). The Copy Enable signals are asserted during mis-matched cycles and are used by the PCEB to enable byte copy operations between the SD data byte lanes 0, 1, 2, and 3 as follows: SDCPYEN01#: Copy between Byte Lane 0 (SD[7:0]) and Byte Lane 1 (SD[15:8]) SDCPYEN02#: Copy between Byte Lane 0 (SD[7:0]) and Byte Lane 2 (SD[23:16]) SDCPYEN03#: Copy between Byte Lane 0 (SD[7:0]) and Byte Lane 3 (SD[31:24]) SDCPYEN13#: Copy between Byte Lane 1 (SD[15:8]) and Byte Lane 3 (SD[31:24]) Note that the direction of the copy is controlled by SDCPYUP.
SDCPYUP	in	**SYSTEM DATA COPY UP:** SDCPYUP controls the direction of the byte copy operation. A high on SDCPYUP indicates a COPY UP operation where the lower byte(s) of the SD data bus are copied onto the higher byte(s) of the bus. A low on the signal indicates a COPY DOWN operation where the higher byte(s) of the data bus are copied on to the lower byte(s) of the bus. The PCEB uses this signal to perform the actual data byte copy operation during mis-matched cycles.
SDOE[2:0]#	in	**SYSTEM DATA OUTPUT ENABLE:** These active Low signals enable the SD data output onto the EISA Bus. The ESC only activates these signals during mis-matched cycles. The PCEB uses these signal to enable the SD data buffers as follows: SDOE0# Enables byte lane 0 SD[7:0] SDOE1# Enables byte lane 1 SD[15:8] SDOE2# Enables byte lane 3 SD[31:24] and byte lane 2 SD[23:16]
SDLE[3:0]#	in	**SYSTEM DATA LATCH ENABLE:** SDLE[3:0]# enable the latching of data on the EISA Bus. These signals are activated only during mis-matched cycles, except PCEB-initiated write cycles. The PCEB uses these signals to latch the SD data bus as follows: SDLE0# Latch byte lane 0 SD[7:0] SDLE1# Latch byte lane 1 SD[15:8] SDLE2# Latch byte lane 2 SD[23:16] SDLE3# Latch byte lane 3 SD[31:24]

2.7 Test Signal

Pin Name	Type	Description
TEST#	in	**TEST:** This pin is used to tri-state all PCEB outputs. During normal operations, this pin must be tied high.

3.0 REGISTER DESCRIPTION

The PCEB contains both PCI configuration registers and I/O registers. The configuration registers (Table 1) are located in PCI configuration space and are only accessible from the PCI Bus. The addresses shown in the table for each register are offset values that appear on AD[7:2] and C/BE[3:0]#. The configuration registers can be accessed as Byte, Word (16-bit), or Dword (32-bit) quantities. All multi-byte numeric fields use "little-endian" ordering (i.e., lower addresses contain the least significant parts of the fields).

The BIOS Timer is the only non-configuration register (Section 3.2, I/O Registers). This register, like the configuration registers, is only accessible from the PCI Bus. The BIOS Timer Register can be accessed as byte, word, or Dword quantities.

Some of the PCEB registers contain reserved bits. These bits are labeled "Reserved". Software must take care to deal correctly with bit-encoded fields that are reserved. On reads, software must use appropriate masks to extract the defined bits and not rely on reserved bits being any particular value. On writes, software must ensure that the values of reserved bits are preserved. That is, the values of reserved bit positions must first be read, merged with the new values for other bit positions and the data then written back.

In addition to reserved bits within a register, the PCEB contains address locations in the PCI configuration space that are marked "Reserved" (Table 1). The PCEB responds to accesses to these address locations by completing the PCI cycle. When a reserved register location is read, 0000h is returned. Writes have no affect on the PCEB.

During a hard reset (PCIRST# asserted), the PCEB registers are set to pre-determined **default** states. The default values are indicated in the individual register descriptions.

During the address phase of a configuration cycle, Bits [10:8] encode one of eight possible functions on a device. The PCEB only supports one function; that of a bridge between the PCI and EISA/ISA Busses. This function has the code of 000. Thus, for accessing PCEB configuration registers, Bits[10:8] = 000 of the address. If the PCEB IDSEL is asserted and any of the above three bits is 1, the PCEB returns all zeros for a read and does not respond to a write.

3.1 Configuration Registers

Table 1 summarizes the PCEB configuration space registers. Following the table, is a detailed description of each register and register bit. The register descriptions are arranged in the order that they appear in Table 1. The following nomenclature is used for access attributes.

RO **Read Only**. If a register is read only, writes to this register have no effect.

R/W **Read/Write**. A register with this attribute can be read and written.

R/WC **Read/Write Clear**. A register bit with this attribute can be read and written. However, a write of a 1 clears (sets to 0) the corresponding bit and a write of a 0 has no effect.

NOTE:

Some register fields are used to program address ranges for various PCEB functions. The register contents represent the address bit value and not the signal level on the bus. For example, the upper address lines on the EISA Bus have inverted signals (LA[31:24]#). However, this inversion is automatically handled by the PCEB hardware and is transparent to the programmer.

Table 1. Configuration Registers

Address Offset	Abbreviation	Register Name	Access
00–01h	VID	Vendor Identification	RO
02–03h	DID	Device Identification	RO
04–05h	PCICMD	Command Register	R/W
06–07h	PCISTS	Status Register	RO, R/WC
08h	RID	Revision Identification	RO
09–0Ch	—	Reserved	—
0Dh	MLTIM	Master Latency Timer	R/W
0E–3Fh	—	Reserved	—
40h	PCICON	PCI Control	R/W
41h	ARBCON	PCI Arbiter Control	R/W
42h	ARBPRI	PCI Arbiter Priority Control	R/W
43h	ARBPRIX	PCI Arbiter Priority Control Extension	R/W
44h	MCSCON	MEMCS# Control	R/W
45h	MCSBOH	MEMCS# Bottom of Hole	R/W
46h	MCSTOH	MEMCS# Top of Hole	R/W
47h	MCSTOM	MEMCS# Top of Memory	R/W
48–49h	EADC1	EISA Address Decode Control 1	R/W
4A–4Bh	—	Reserved	—
4Ch	IORTC	ISA I/O Recovery Time Control	R/W
4Dh–53h	—	Reserved	—
54h	MAR1	MEMCS# Attribute Register #1	R/W
55h	MAR2	MEMCS# Attribute Register #2	R/W
56h	MAR3	MEMCS# Attribute Register #3	R/W
57h	—	Reserved	—
58h	PDCON	PCI Decode Control	R/W
59h	—	Reserved	—
5Ah	EADC2	EISA Address Decode Control 2	R/W
5Bh	—	Reserved	—

Table 1. Configuration Registers (Continued)

Address Offset	Abbreviation	Register Name	Access
5Ch	EPMRA	EISA-to-PCI Memory Region Attributes	R/W
5D–5Fh	—	Reserved	—
60–6Fh	MEMREGN[4:1]	EISA-to-PCI Memory Region Address (4 Registers)	R/W
70–7Fh	IOREGN[4:1]	EISA-to-PCI I/O Region Address (4 Registers)	R/W
80–81h	BTMR	BIOS Timer Base Address	R/W
84h	ELTCR	EISA Latency Timer Control Register	R/W
85–87h	—	Reserved	—
88–8Bh	PTCR	PCEB Test Control Register—**DO NOT WRITE**	—
8C–FFh	—	Reserved	—

3.1.1 VID—VENDOR IDENTIFICATION REGISTER

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

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

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

3.1.2 DID—DEVICE IDENTIFICATION REGISTER

Address Offset: 02–03h
Default Value: 0482h
Attribute: Read Only
Size: 16 bits

The DID Register contains the device identification number. This register, along with the VID Register, define the PCEB. Writes to this register have no effect.

Bit	Description
15:0	**Device Identification Number:** This is a 16-bit value assigned to the PCEB.

3.1.3 PCICMD—PCI COMMAND REGISTER

Address Offset: 04–05h
Default Value: 0007h
Attribute: Read/Write, Read Only
Size: 16 bits

This 16-bit register contains PCI interface control information. This register enables/disables PCI parity error checking, enables/disables PCEB bus master capability, and enables/disables the PCEB to respond to PCI-originated memory and I/O cycles. Note that, for certain PCI functions that are not implemented within the PCEB, the control bits are still shown (labeled "not supported").

Bit	Description
15:9	**Reserved**
8	**SERR# Enable (SERRE)—Not Supported—RO:** Function of this bit is to control the SERR# signal. Since the PCEB does not implement the SERR# signal, this bit always reads as 0 (disabled).
7	**Wait State Control (WSC)—Not Supported—RO:** This bit controls insertion of wait-states for devices that do not meet the 33-10 PCI specification. Since PCEB meets the 33-10 specification, this control function is not implemented. WSC is always read as 0.
6	**Parity Error Enable (PERRE)—R/W:** PERRE controls the PCEB's response to PCI parity errors. When PERRE = 1, the PCEB asserts the PERR# signal when a parity error is detected. When PERRE = 0, the PCEB ignores any parity errors that it detects. After PCIRST#, PERRE = 0 (parity checking disabled).
5	**VGA Palette Snoop (VGPS)—Not Supported—RO:** This bit is intended only for specific control of PCI-based VGA devices and it is not applicable to the PCEB. This bit is not implemented and always reads as 0.
4	**Memory Write and Invalidate Enable (MWIE)—Not Supported—RO:** This is an enable bit for using the Memory Write and Invalidate command. The PCEB doesn't support this command as a master. As a slave the PCEB aliases this command to a memory write. This bit always reads as 0 (disabled).
3	**Special Cycle Enable (SCE)—Not Supported—RO:** Since this capability is not implemented, the PCEB does not respond to any type of special cycle. This bit always reads as 0.
2	**Bus Master Enable (BME)—R/W:** ME enables/disables the PCEB's PCI Bus master capability. When BME = 0, the PCEB bus master capability is disabled. This prevents the PCEB from requesting the PCI Bus on behalf of EISA/ISA masters, the DMA, or the Line Buffers. When BME = 1, the bus master capability is enabled. This bit is set to 1 after PCIRST#.
1	**Memory Space Enable (MSE)—R/W:** This bit enables the PCEB to accept PCI-originated memory cycles. When MSE = 1, the PCEB responds to PCI-originated memory cycles to the EISA Bus. When MSE = 0, the PCEB does not respond to PCI-originated memory cycles to the EISA Bus (DEVSEL# is inhibited). This bit is set to 1 (enabled for BIOS access) after PCIRST#.
0	**I/O Space Enable (IOSE)—R/W:** This bit enables the PCEB to accept PCI-originated I/O cycles. When IOSE = 1, the PCEB responds to PCI-originated I/O cycles. When IOSE = 0, the PCEB does not respond to a PCI I/O cycle (DEVSEL# is inhibited), including I/O cycles bound for the EISA Bus. This bit is set to 1 (I/O space enabled) after PCIRST#.

3.1.4 PCISTS—PCI STATUS REGISTER

Address Offset: 06–07h
Default Value: 0200h
Attribute: Read Only, Read/Write Clear
Size: 16 bits

This 16-bit register provides status information for PCI Bus-related events. Some bits are read/write clear. These bits are set to 0 whenever the register is written, and the data in the corresponding bit location is 1 (R/WC). For example, to clear bit 12 and not affect any other bits, write the value 0001_0000_0000_0000b to this register. Note that for certain PCI functions that are not implemented in the PCEB, the control bits are still shown (labeled "not supported").

Bit	Description
15	**Parity Error Status (PERRS)—R/WC:** This bit is set to 1 whenever the PCEB detects a parity error, even if parity error handling is disabled (as controlled by bit 6 in the PCI Command Register). Software sets PERRS to 0 by writing a 1 to this bit location.
14	**SERR# Status (SERRS)—Not Supported:** This bit is used to indicate that a PCI device asserted the SERR# signal. The PCEB does not implement this signal. SERRS is always read as 0.
13	**Master Abort Status (MA)—R/WC:** When the PCEB, as a master, generates a master abort, this bit is set to 1. Software sets MA to 0 by writing a 1 to this bit location.
12	**Received Target Abort Status (RTAS)—R/WC:** When the PCEB, as a master, receives a target abort condition, this bit is set to 1. Software sets RTAS to 0 by writing a 1 to this bit location.
11	**Signaled Target Abort Status (STAS)—Not Supported:** This bit is set to 1 by a PCI target device when they generate a Target Abort. Since the PCEB never generates a target abort, this bit is not implemented and will always be read as a 0.
10:9	**DEVSEL Timing Status (DEVT)—RO:** This read only field indicates the timing of the DEVSEL# signal when PCEB responds as a target. The PCI Specification defines three allowable timings for assertion of DEVSEL#: 00b = fast, 01b = medium, and 10b = slow (11b is reserved). DEVT indicates the slowest time that a device asserts DEVSEL# for any bus command, except configuration read and configuration write cycles. The PCEB implements medium speed DEVSEL# timing and, therefore, DEVT[10:9] = 01 when read.
8:0	**Reserved**

3.1.5 RID—REVISION IDENTIFICATION REGISTER

Address Offset: 08h
Default Value: 03h (82375EB, A-2 stepping)
04h (82375SB, B-0 stepping)
Attribute: Read Only
Size: 8 bits

This 8-bit register contains the device revision number of the PCEB. Writes to this register have no effect.

Bit	Description
7:0	**Revision Identification Number:** This 8-bit value is the revision number of the PCEB.

3.1.6 MLT—MASTER LATENCY TIMER REGISTER

Address Offset: 0Dh
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

This 8-bit register contains the programmable value of the Master Latency Timer for use when the PCEB is a master on the PCI Bus. The granularity of the timer is 8 PCI clocks. Thus, bits[2:0] are not used and always read as 0s.

Bit	Description
7:3	**Count Value:** This 5-bit field contains the count value of the Master Latency Timer, with a granularity of 8 PCI clocks. For example, value 00101b provides a time-out period of 5x8 = 40 PCI clocks. Maximum count value is 11111b, which corresponds to 248 PCI clocks.
2:0	**Reserved**

3.1.7 PCICON—PCI CONTROL REGISTER

Address Offset: 40h
Default Value: 20h
Attribute: Read/Write
Size: 8 bits

This 8-bit register enables/disables the PCEB's data buffers, defines the subtractive decoding sample point, and enables/disables response to the PCI interrupt acknowledge cycle.

NOTE:
The Line Buffers are typically enabled or disabled during system initialization. These buffers should not be dynamically enabled/disabled during runtime. Otherwise, data coherency can be affected, if a buffer containing valid write data is disabled and then, later, re-enabled.

Bit	Description
7	**Reserved**
6	**EISA-To-PCI Line Buffer Enable (ELBE):** When ELBE = 0, the EISA-to-PCI Line Buffers are disabled and when ELBE = 1, the EISA-to-PCI Line Buffers are enabled. After PCIRST#, the Line Buffers are disabled (ELBE = 0). Note that when ELBE is set to 1, the line buffers are utilized for transfers to or from the regions defined by the REG[4:1] bits in the EPMRA register (offset 5Ch).
5	**Interrupt Acknowledge Enable (IAE):** When IAE = 0, the PCEB decodes PCI interrupt acknowledge cycles in a semi-subtractive manner. When there is data posted in the Line Buffers, the PCEB intervenes in the PCI interrupt acknowledge cycle by generating a retry. The PCEB also initiates a buffer flush operation and will keep generating retries until the buffers are flushed. The PCEB then subtractively decodes the PCI interrupt acknowledge cycle in order to allow an external PCI-based interrupt controller to respond with the vector. If no external PCI-based interrupt controller has responded to the PCI Interrupt Acknowledge cycle at the DEVSEL# sampling point, the cycle is handled by the PCEB in a subtractive decode manner. When IAE = 1, the PCEB positively decodes the interrupt acknowledge cycles and responds to the cycles in the normal fashion (i.e., uses the PEREQ#/INTA# signal to fetch the vector from the ESC, after the internal buffers are flushed).

Bit	Description
4:3	**Subtractive Decoding Sample Point (SDSP):** The SDSP field determines the DEVSEL# sample point, after which an inactive DEVSEL# results in the PCEB forwarding the unclaimed PCI cycle to the EISA Bus (subtractive decoding). This setting should match the slowest device in the system. When the MEMCS# function is enabled, MEMCS# is sampled as well as an early indication of an eventual DEVSEL#. **Bits[4:3] Operation** 00 Slow sample point (default value) 01 Typical sample point 10 Reserved 11 Reserved
2	**Reserved.** This bit must be 0 when programming this register.
1:0	**Reserved**

3.1.8 ARBCON—PCI ARBITER CONTROL REGISTER

Address Offset: 41h
Default Value: 80h
Attribute: Read/Write
Size: 8 bits

This register controls the operation of the PCEB's internal PCI arbiter. The register enables/disables auto-PEREQ#, controls the master retry timer, enables/disables CPU bus parking, controls bus lock, and enables/disables the guaranteed access time (GAT) mode for EISA/ISA accesses.

NOTE:

1. For proper system operation, the master retry timer (bits[4:3]) must not be disabled. This field defaults to 00 (disabled) and must be program to either 01, 10, or 11.
2. The PCMC Host bridge device requires that bit 7 be set to 1 (default). However, other chip sets might need to have this function disabled to provide more optimum performance for EISA subsystems. This functionality is built-in to prevent starvation of PCI agents (in particular, the host bridge, i.e., CPU) when EISA masters are performing transactions in the GAT mode. If this function is disabled, the host bridge must be capable of generating the PCI Bus request, even when the Host Bus is not controlled by the CPU (CPU tri-stated all Host Bus signals, or even only address bus, in response to HOLD/AHOLD). The CPU pin that provides an indication of a request for the external bus (e.g. after cache miss) can be used by the host bridge to generate the request for the PCI Bus during GAT mode operations, even when no address lines are driven by the CPU.

Bit	Description
7	**Auto-PEREQ# Control (APC):** APC Enables/Disables control of the auto-PEREQ# function when GAT mode is enabled via bit 0 (GAT = 1). When APC = 1 (and GAT = 1), the PEREQ# signal is asserted whenever the EISAHLDA signal is asserted. When APC = 0, the PEREQ# signal is not automatically asserted but it will be activated upon PCI Bus request from any PCI agent. After PCIRST#, APC = 1 (enabled). See note.
6:5	**Reserved**
4:3	**Master Retry Timer (MRT):** This 2-bit field determines the number of PCICLKs after the first retry that a PCI initiator's bus request will be masked. Note that for proper system operation, this register must be programmed with either 01, 10, 11. **Bits[4:3]** — **Operation** 00 — Timer disabled, Retries never masked. (Default) 01 — Retries unmasked after 16 PCICLK's. 10 — Retries unmasked after 32 PCICLK's. 11 — Retries unmasked after 64 PCICLK's.
2	**Bus Park (BP):** When BP = 1, the PCEB will park CPUREQ# on the PCI Bus when it detects the PCI Bus idle. If BP = 0, the PCEB takes responsibility for driving AD, C/BE# and PAR signals upon detection of bus idle state. After PCIRST#, BP = 0 (disabled).
1	**Bus Lock (BL):** When BL = 1, Bus Lock is enabled. The arbiter considers the entire PCI Bus locked upon initiation of any LOCKed transaction. When BL = 0, Resource Lock is enabled. A LOCKed agent is considered a LOCKed resource and other agents may continue normal PCI transactions. After PCIRST#, BL = 0 (disabled).
0	**Guaranteed Access Time (GAT):** When GAT = 1, the PCEB is configured for Guaranteed Access Time mode. This mode guarantees the 2.1 μs CHRDY time-out specification for the EISA/ISA Bus. When the PCEB is a PCI initiator on behalf of an EISA/ISA master, the PCI and main memory bus (host) are arbitrated for in serial and must be owned before the EISA/ISA master is given ownership of the EISA Bus. If the PCEB is not programmed for Guaranteed Access Time (GAT = 0), the EISA/ISA master is first granted the EISA Bus, before the PCI Bus is arbitrated. After a PCIRST#, GAT = 0 (disabled).

3.1.9 ARBPRI—PCI ARBITER PRIORITY CONTROL REGISTER

Address Offset: 42h
Default Value: 04h
Attribute: Read/Write
Size: 8 bits

This register controls the operating modes of the PCEB's internal PCI arbiter. The arbiter consists of four arbitration banks that support up to six masters and three arbitration priority modesfixed priority, rotating priority and mixed priority modes. See Section 5.4, PCI Bus Arbitration for details on programming and using different arbitration modes.

Bit	Description
7	**Bank 3 Rotate Control:** 1 = Enable; 0 = Disable
6	**Bank 2 Rotate Control:** 1 = Enable; 0 = Disable
5	**Bank 1 Rotate Control:** 1 = Enable; 0 = Disable
4 3:2	**Bank 0 Rotate Control:** 1 = Enable; 0 = Disable **Bank 2 Fixed Priority Mode Select—b,a:** ba 00 = Bank0 > Bank3 > Bank1 10 = Bank3 > Bank1 > Bank0 01 = Bank1 > Bank0 > Bank3 11 = Reserved
1	**Bank 1 Fixed Priority Mode Select:** 1 = REQ3# > CPUREQ#; 0 = CPUREQ# > REQ3
0	**Bank 0 Fixed Priority Mode Select:** 1 = REQ0# > PCEBREQ#; 0 = PCEBREQ# > REQ0#. Note that PCEBREQ# is a PCEB internal signal.

3.1.10 ARBPRIX—PCI ARBITER PRIORITY CONTROL EXTENSION REGISTER

Address Offset: 43h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

This register controls the fixed priority mode for bank 3 of the PCEB's internal arbiter. The ARBPRIX Register is used in conjunction with the PCI Arbiter Priority Control (ARBPRI) Register.

Bit	Description
7:1	**Reserved**
0	**Bank 3 Fixed Priority Mode Select:** 1 = REQ2# > REQ1#; 0 = REQ1# > REQ2#.

3.1.11 MCSCON—MEMCS# CONTROL REGISTER

Address Offset: 44h
Default value: 00h
Attribute: Read/Write
Size: 8 bits

The MCSCON Register provides the master enable for generating MEMCS#. This register also provides read enable (RE) and write enable (WE) attributes for two main memory regions (the 512 KByte - 640 KByte region and an upper BIOS region). PCI accesses within the enabled regions result in the generation of MEMCS#. Note that the 0-512 KByte region does not have RE and WE attribute bits. The 0-512 KByte region can only be disabled with the MEMCS# Master Enable bit (bit 4). Note also, that when the RE and WE bits are both 0 for a particular region, the PCI master can not access the corresponding region in main memory (MEMCS# is not generated for either reads or writes).

Bit	Description
7:5	**Reserved**
4	**MEMCS# Master Enable:** When bit 4 = 1, the PCEB asserts MEMCS# for all accesses to the defined MEMCS# region (as defined by the MCSTOM Register and excluding the memory hole defined by the MCSBOH and MCSTOH Registers), if the accessed location is in a region enabled by bits [3:0] of this register or in the regions defined by the MAR1, MAR2, and MAR3 registers. When bit 4 = 0, the entire MEMCS# function is disabled and MEMCS# is never asserted.
3	**Write Enable For 0F0000–0FFFFFh (Upper 64 KByte BIOS):** When bit 3 = 1, the PCEB generates MEMCS# for PCI master memory write accesses to the address range 0F0000–0FFFFFh. When bit 3 = 0, the PCEB does not generate MEMCS# for PCI master memory write accesses to the address range 0F0000–0FFFFFh.
2	**Read Enable For 0F0000–0FFFFFh (Upper 64 KByte BIOS):** When bit 2 = 1, the PCEB generates MEMCS# for PCI master memory read accesses to the address range 0F0000–0FFFFFh. When bit 2 = 0, the PCEB does not generate MEMCS# for PCI master memory read accesses to the address range 0F0000–0FFFFFh.
1	**Write Enable For 080000–09FFFFh (512–640 KByte):** When bit 1 = 1, the PCEB generates MEMCS# for PCI master memory write accesses to the address range 080000–09FFFFh. When bit 1 = 0, the PCEB does not generate MEMCS# for PCI master memory write accesses to the address range 080000–09FFFFh.
0	**Read Enable For 080000–09FFFFh (512–640 KByte):** When bit 0 = 1, the PCEB generates MEMCS# for PCI master memory read accesses to the address range 080000–09FFFFh. When bit 0 = 0, the PCEB does not generate MEMCS# for PCI master memory read accesses to the address range 080000–09FFFFh.

3.1.12 MCSBOH—MEMCS# BOTTOM OF HOLE REGISTER

Address Offset: 45h
Default value: 10h
Attribute: Read/Write
Size: 8 bits

This register defines the bottom of the MEMCS# hole. MEMCS# is not generated for accesses to addresses within the hole defined by this register and the MCSTOH Register. The hole is defined by the following equation:

TOH ≥ address ≥ BOH. TOH is the top of the MEMCS# hole defined by the MCSTOH Register and BOH is the bottom of the MEMCS# hole defined by this register.

For example, to program the BOH at 1 MByte, the value of 10h should be written to this register. To program the BOH at 2 MByte + 64 KByte this register should be programmed to 21h. To program the BOH at 8 MByte this register should be programmed to 80h.

When the TOH < BOH the hole is disabled. If TOH = BOH, the hole size is 64 KBytes. It is the responsibility of the programmer to guarantee that the BOH is at or above 1 MB. AD[31:24] must be 0's for the hole, meaning the hole is restricted to be under the 16 MByte boundary. The default value for the BOH and TOH disables the hole.

Bit	Description
7:0	**Bottom of MEMCS# Hole:** Bits[7:0] correspond to address lines AD[23:16], respectively.

3.1.13 MCSTOH—MEMCS# TOP OF HOLE REGISTER

Address Offset: 46h
Default value: 0Fh
Attribute: Read/Write
Size: 8 bits

This register defines the top of the MEMCS# hole. MEMCS# is not generated for accesses to addresses within the hole defined by this register and the MCSBOH Register. The hole is defined by the following equation:

TOH ≥ address ≥ BOH. TOH is the top of the MEMCS# hole defined by this register and BOH is the bottom of the MEMCS# hole defined by the MCSBOH Register.

For example, to program the TOH at 1 MByte + 64 KByte, this register should be programmed to 10h. To program the TOH at 2 MByte + 128 KByte this register should be programmed to 21h. To program the TOH at 12 MByte this register should be programmed to BFh.

When the TOH < BOH the hole is disabled. If TOH = BOH, the hole size is 64 KBytes. It is the responsibility of the programmer to guarantee that the TOH is above 1 MByte. AD[31:24] must be 0's for the hole, meaning the hole is restricted to be under the 16 MByte boundary. The default value for the BOH and TOH disables the hole.

Bit	Description
7:0	**Top of MEMCS# Hole:** Bits[7:0] correspond to address lines AD[23:16], respectively.

3.1.14 MCSTOM—MEMCS# TOP OF MEMORY REGISTER

Address Offset: 47h
Default value: 00h
Attribute: Read/Write
Size: 8 bits

This register determines MEMCS# top of memory boundary. The top of memory boundary ranges from 2 MBytes-1 to 512 MBytes-1, in 2 MByte increments. This register is typically set to the top of main memory. Accesses ≥ 1 MByte and ≤ top of memory boundary results in the assertion of the MEMCS# signal (unless the address resides in the hole programmed via the MCSBOH and MCSTOH Registers). A value of 00h sets top of memory at 2 MBytes-1 (including the 2 MByte-1 address). A value of FFh sets the top of memory at 512 MByte-1 (including the 512 MByte-1 address).

Bit	Description
7:0	**Top of MEMCS# Memory Boundary:** Bits[7:0] correspond to address lines AD[28:21], respectively.

3.1.15 EADC1—EISA ADDRESS DECODE CONTROL 1 REGISTER

Address Offset 48–49h
Default value: 0001h
Attribute: Read/Write
Size: 16 bits

This 16-bit register specifies EISA-to-PCI mapping of the 0-1 MByte memory address range. For each bit position, the memory block is enabled if the corresponding bit = 1 and is disabled if the bit = 0. EISA or DMA memory cycles to the enabled blocks result in the EISA cycle being forwarded to the PCI Bus. For disabled memory blocks, the EISA memory cycle is not forwarded to the PCI Bus.

Bit	Description
15	**880–896 KBytes Memory Enable:** EISA-to-PCI mapping for this memory space is enabled when this bit is 1 and disabled when this bit is 0.
14	**864–880 KBytes Memory Enable:** EISA-to-PCI mapping for this memory space is enabled when this bit is 1 and disabled when this bit is 0.
13	**848–864 KBytes Memory Enable:** EISA-to-PCI mapping for this memory space is enabled when this bit is 1 and disabled when this bit is 0.
12	**832–848 KBytes Memory Enable:** EISA-to-PCI mapping for this memory space is enabled when this bit is 1 and disabled when this bit is 0.
11	**816–832 KBytes Memory Enable:** EISA-to-PCI mapping for this memory space is enabled when this bit is 1 and disabled when this bit is 0.
10	**800–816 KBytes Memory Enable:** EISA-to-PCI mapping for this memory space is enabled when this bit is 1 and disabled when this bit is 0.
9	**784–800 KBytes Memory Enable:** EISA-to-PCI mapping for this memory space is enabled when this bit is 1 and disabled when this bit is 0.
8	**768–784 KBytes Memory Enable:** EISA-to-PCI mapping for this memory space is enabled when this bit is 1 and disabled when this bit is 0.
7:3	**Reserved**
2	**640–768 KBytes VGA Memory Enable:** EISA-to-PCI mapping for this memory space is enabled when this bit is 1 and disabled when this bit is 0.
1	**512–640 KBytes Memory Enable:** EISA-to-PCI mapping for this memory space is enabled when this bit is 1 and disabled when this bit is 0.
0	**0–512 KBytes Memory Enable:** EISA-to-PCI mapping for this memory space is enabled when this bit is 1 and disabled when this bit is 0.

3.1.16 IORT—ISA I/O RECOVERY TIMER REGISTER

Address Offset: 4Ch
Default Value: 56
Attribute: Read/Write
Size: 8 bits

The I/O recovery logic is used to guarantee a minimum amount of time between back-to-back 8-bit and 16-bit PCI-to-ISA I/O slave accesses. These minimum times are programmable.

The I/O recovery mechanism in the PCEB is used to add recovery delay between PCI-originated 8-bit and 16-bit I/O cycles to ISA devices. The delay is measured from the rising edge of the EISA command signal (CMD#) to the falling edge of the next EISA command. The delay is equal to the number of EISA Bus clocks (BCLKs) that correspond to the value contained in bits [1:0] for 16-bit I/O devices and in bits[5:3] for 8-bit I/O devices. Note that no additional delay is inserted for back-to-back I/O "sub-cycles" generated as a result of byte assembly or disassembly. This register defaults to 8- and 16-bit recovery enabled with two clocks of I/O recovery.

Bit	Description
7	**Reserved**
6	**Bit I/O Recovery Enable:** This bit enables the recovery times programmed into bits 0 and 1 of this register. When this bit is set to 1, the recovery times shown for bits 5-3 are enabled. When this bit is set to 0, recovery times are disabled.
5:3	**8-Bit I/O Recovery times:** This 3-bit field defines the recovery times for 8-bit I/O. Programmable delays between back-to-back 8-bit PCI cycles to ISA I/O slaves is shown in terms of EISA clock cycles (BCLK). The selected delay programmed into this field is enabled/disabled via bit 6 of this register. **Bits [5:3] BCLK** 001 1 010 2 011 3 100 4 101 5 110 6 111 7 000 8
2	**16-Bit I/O Recovery Enable:** This bit enables the recovery times programmed into bits 0 and 1 of this register. When this bit is set to 1, the recovery times shown for bits 0 and 1 are enabled. When this bit is set to 0, recovery times are disabled.
1:0	**16-Bit I/O Recovery Times:** This 2-bit field defines the Recovery time for 16-bit I/O. Programmable delays between back-to-back 16-bit PCI cycles to ISA I/O slaves is shown in terms of EISA clock cycles (BCLK). The selected delay programmed into this field is enabled/disabled via bit 2 of this register. **Bits [1:0]B CLK** 01 1 10 2 11 3 00 4

3.1.17 MAR1—MEMCS# ATTRIBUTE REGISTER #1

Address Offset: 54h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

RE—Read Enable. When the RE bit (bit 6, 4, 2, 0) is set to a 1, the PCEB generates MEMCS# for PCI master, DMA, or EISA master memory read accesses to the corresponding segment in main memory. When the RE bit is set to a 0, the PCEB does not generate MEMCS# for PCI master, DMA, or EISA master memory read accesses to the corresponding segment. When the RE and WE bits are both 0 (or bit 4 in the MEMCS# Control Register is set to a 0-disabled), the PCI master, DMA, or EISA master can not access the corresponding segment in main memory.

WE—Write Enable. When the WE bit (bit 7, 5, 3, 1) is set to a 1, the PCEB generates MEMCS# for PCI master, DMA, or EISA master memory write accesses to the corresponding segment in main memory. When this bit is set to a 0, the PCEB does not generate MEMCS# for PCI master, DMA, or EISA master memory write accesses to the corresponding segment. When the RE and WE bits are both 0 (or bit 4 in the MEMCS# Control Register is set to a 0-disabled), the PCI master, DMA, or EISA master can not access the corresponding segment in main memory.

Bit	Description
7	**0CC000–0CFFFFh Add-on BIOS: WE:** 1 = Enable; 0 = Disable
6	**0CC000–0CFFFFh Add-on BIOS: RE:** 1 = Enable; 0 = Disable
5	**0C8000–0CBFFFh Add-on BIOS: WE:** 1 = Enable; 0 = Disable
4	**0C8000–0CBFFFh Add-on BIOS: RE:** 1 = Enable; 0 = Disable
3	**0C4000–0C7FFFh Add-on BIOS: WE:** 1 = Enable; 0 = Disable
2	**0C4000–0C7FFFh Add-on BIOS: RE:** 1 = Enable; 0 = Disable
1	**0C0000–0C3FFFh Add-on BIOS: WE:** 1 = Enable; 0 = Disable
0	**0C0000–0C3FFFh Add-on BIOS: RE:** 1 = Enable; 0 = Disable

3.1.18 MAR2—MEMCS# ATTRIBUTE REGISTER #2

Address Offset: 55h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

RE—Read Enable. When the RE bit (bit 6, 4, 2, 0) is set to a 1, the PCEB generates MEMCS# for PCI master, DMA, or EISA master memory read accesses to the corresponding segment in main memory. When this bit is set to a 0, the PCEB does not generate MEMCS# for PCI master, DMA, or EISA master memory read accesses to the corresponding segment. When the RE and WE bits are both 0 (or bit 4 in the MEMCS# Control Register is set to a 0-disabled), the PCI master, DMA, or EISA master can not access the corresponding segment in main memory.

WE—Write Enable. When the WE bit (bit 7, 5, 3, 1) is set to a 1, the PCEB generates MEMCS# for PCI master, DMA, or EISA master memory write accesses to the corresponding segment in main memory. When this bit is set to a 0, the PCEB does not generate MEMCS# for PCI master, DMA, or EISA master memory write accesses to the corresponding segment. When the RE and WE bits are both 0 (or bit 4 in the MEMCS# Control Register is set to a 0-disabled), the PCI master, DMA, or EISA master can not access the corresponding segment in main memory.

Bit	Description
7	**0DC000–0DFFFFh Add-on BIOS: WE:** 1 = Enable; 0 = Disable
6	**0DC000–0DFFFFh Add-on BIOS: RE:** 1 = Enable; 0 = Disable
5	**0D8000–0DBFFFh Add-on BIOS: WE:** 1 = Enable; 0 = Disable
4	**0D8000–0DBFFFh Add-on BIOS: RE:** 1 = Enable; 0 = Disable
3	**0D4000–0D7FFFh Add-on BIOS: WE:** 1 = Enable; 0 = Disable
2	**0D4000–0D7FFFh Add-on BIOS: RE:** 1 = Enable; 0 = Disable
1	**0D0000–0D3FFFh Add-on BIOS: WE:** 1 = Enable; 0 = Disable
0	**0D0000–0D3FFFh Add-on BIOS: RE:** 1 = Enable; 0 = Disable

3.1.19 MAR3—MEMCS# ATTRIBUTE REGISTER #3

Address Offset: 56h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

RE—Read Enable. When the RE bit (bit 6, 4, 2, 0) is set to a 1, the PCEB generates MEMCS# for PCI master, DMA, EISA master memory read accesses to the corresponding segment in main memory. When this bit is set to a 0, the PCEB does not generate MEMCS# for PCI master, DMA, or EISA master memory read accesses to the corresponding segment. When the RE and WE bits are both 0 (or bit 4 in the MEMCS# Control Register is set to a 0-disabled), the PCI master can not access the corresponding segment in main memory.

WE—Write Enable. When the WE bit (bit 7, 5, 3, 1) is set to a 1, the PCEB generates MEMCS# for PCI master , DMA, EISA master memory write accesses to the corresponding segment in main memory. When this bit is set to a 0, the PCEB does not generate MEMCS# for PCI master, DMA, or EISA master memory write accesses to the corresponding segment. When the RE and WE bits are both 0 (or bit 4 in the MEMCS# Control Register is set to a 0-disabled), the PCI master can not access the corresponding segment in main memory.

Bit	Description
7	**0EC000–0EFFFFh BIOS Extension: WE:** 1 = Enable; 0 = Disable
6	**0EC000–0EFFFFh BIOS Extension: RE:** 1 = Enable; 0 = Disable
5	**0E8000–0EBFFFh BIOS Extension: WE:** 1 = Enable; 0 = Disable
4	**0E8000–0EBFFFh BIOS Extension: RE:** 1 = Enable; 0 = Disable
3	**0E4000–0E7FFFh BIOS Extension: WE:** 1 = Enable; 0 = Disable
2	**0E4000–0E7FFFh BIOS Extension: RE:** 1 = Enable; 0 = Disable
1	**0E0000–0E3FFFh BIOS Extension: WE:** 1 = Enable; 0 = Disable
0	**0E0000–0E3FFFh BIOS Extension: RE:** 1 = Enable; 0 = Disable

3.1.20 PDCON—PCI DECODE CONTROL REGISTER

Address Offset: 58h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

This register enables/disables positive decode of PCI accesses to the IDE and 8259 locations residing in the expansion bus subsystem. For the 82374SB, this register controls the mode of address decode (subtractive or negative) for memory cycles on the PCI Bus.

Subtractive decoding:

PCI memory cycles that are not claimed on the PCI Bus (i.e., DEVSEL# inactive) are forwarded to the EISA Bus. This is the default on power up.

Negative decoding (82374SB Only):

PCI memory cycles that are not mapped to one of the regions defined by A, B, or C below, are immediately forwarded to the EISA Bus (i.e. without waiting for DEVSEL# time-out). PCI memory cycles that are decoded to one of the four programmable PCI memory regions, but are not claimed (DEVSEL# negated), are forwarded to the EISA Bus by subtractive decode.

A. Main memory locations defined by the MEMCS# mapping (MCSCON, MCSBOH, MCSTOH, MCSTOM, MAR1, MAR2, and MAR3 Registers).

B. The enabled Video Frame Buffer region, 0A0000–0BFFFFh (as indicated by bit 2 of the EADC1 Register).

C. The four programmable PCI memory regions (defined by the MEMREGN[4:1] registers).

NOTE:
If there are devices on the PCI that are not mapped into any of the regions defined by A, B, or C, then negative decoding can not be used.

Bit	Description
7:6	**Reserved**
5	**8259 Decode Control (8259DC):** This bit enables/disables positive decode of 8259 locations 0020h, 0021h, 00A0h and 00A1h. When this bit is 1, positive decode for these locations are enabled. When this bit is 0, positive decode for these locations is disabled. After reset, this bit is 0. Note that if positive decode is disabled, these 8259 locations can still be accessed via subtractive decode.
4	**IDE Decode Control (IDEDC):** This bit enables/disables positive decode of IDE locations 1F0–1F7h (primary) or 170–177h (secondary) and 3F6h,3F7h (primary) or 376h,377h (secondary). When IDEDC = 0, positive decode is disabled. When IDECD = 1, positive decode is enabled. After reset, this bit is 0. Note that if positive decode is disabled, these IDE locations can still be accessed via subtractive decode.
3:1	**Reserved**
0	**82375EB: Reserved.** Must be 0 when programming this register. **82375SB: PCI Memory Address Decoding Mode (PMAD):** This bit selects between subtractive and negative decoding. When PMAD = 1, negative decoding is selected. When PMAD = 0, subtractive decoding is selected. After reset, this bit is 0.

3.1.21 EADC2—EISA ADDRESS DECODE CONTROL EXTENSION REGISTER

Address Offset: 5Ah
Default value: 00h
Attribute: Read/Write
Size: 8 bits

This register specifies EISA-to-PCI mapping for the 896 KByte to 1 MByte memory address range (BIOS). If this memory block is enabled, EISA memory accesses in this range will result in the EISA cycles being forwarded to the PCI Bus. (Note that enabling this block is necessary if BIOS resides within the PCI and not within the EISA subsystem.)

This register also defines mapping for the 16 MByte - 64 KByte to 16 MByte memory address range. This mapping is important if the BIOS is aliased at the top 64 KBytes of 16 MBytes. If the region is enabled and this address range is within the hole defined by the MCSBOH and MCSTOH Registers or above the top of main memory defined by the MCSTOM Register, the EISA cycle is forwarded to the PCI.

Bit	Description
7:6	**Reserved**
5	**Top 64 KByte of 16 MByte Memory Space Enable (FF0000–FFFFFFh):** This memory block is enabled when this bit is 1 and disabled when this bit is 0.
4	**960 KBytes—1 MByte Memory Space Enable (0F0000–0FFFFFh):** This memory block is enabled when this bit is 1 and disabled when this bit is 0.
3	**944–960 KByte Memory Space Enable (0EC000–0EFFFFh):** This memory block is enabled when this bit is 1 and disabled when this bit is 0.
2	**928–944 KByte Memory Space Enable (0E8000–0EBFFFh):** This memory block is enabled when this bit is 1 and disabled when this bit is 0.
1	**912–928 KByte Memory Space Enable (0E4000–0E7FFFh):** This memory block is enabled when this bit is 1 and disabled when this bit is 0.
0	**896–912 KByte Memory Space Enable (0E0000–0E3FFFh):** This memory block is enabled when this bit is 1 and disabled when this bit is 0.

3.1.22 EPMRA—EISA-TO-PCI MEMORY REGION ATTRIBUTES REGISTER

Address Offset: 5Ch
Default value: 00h
Attribute: Read/Write
Size: 8 bits

This register defines buffering attributes for EISA accesses to PCI memory regions specified by MEMREGN[4:1] Registers. When an EPMRA bit is 1 (and the Line Buffers are enabled via the PCICON Register), EISA accesses to the corresponding PCI memory region are performed in buffered mode. In buffered mode, read prefetching and write posting/assembly are enabled. When an EPMRA bit is 0, EISA accesses to the corresponding PCI memory region are performed in non-buffered mode. In non-buffered mode, a buffer bypass path is used to complete the transaction.

NOTE:

1. Using buffered mode for EISA accesses to PCI memory regions that contain memory-mapped I/O devices can cause unintended side effects. In buffered mode, strong ordering is not preserved within a Dword. If the order of the writes to an I/O device is important, non-buffered mode should be used. Also, read-prefetch can cause unintended changes of status registers in the memory-mapped I/O device.
2. The Line Buffers are typically enabled or disabled during system initialization. These buffers should not be dynamically enabled/disabled during runtime. Otherwise, data coherency can be affected, if a buffer containing valid write data is disabled and then, later, re-enabled.

Bit	Description
7:4	**Reserved**
3	**Region 4 Attribute (REG-4):** EISA accesses to this PCI memory region are buffered when this bit is 1 and non-buffered when this bit is 0. If the Line Buffers are disabled via the PCICON Register (bit 6), buffering is disabled, regardless of the value of this bit.
2	**Region 3 Attribute (REG-3):** EISA accesses to this PCI memory region are buffered when this bit is 1 and non-buffered when this bit is 0. If the Line Buffers are disabled via the PCICON Register (bit 6), buffering is disabled, regardless of the value of this bit.
1	**Region 2 Attribute (REG-2):** EISA accesses to this PCI memory region are buffered when this bit is 1 and non-buffered when this bit is 0. If the Line Buffers are disabled via the PCICON Register (bit 6), buffering is disabled, regardless of the value of this bit.
0	**Region 1 Attribute (REG-1):** EISA accesses to this PCI memory region are buffered when this bit is 1 and non-buffered when this bit is 0. If the Line Buffers are disabled via the PCICON Register (bit 6), buffering is disabled, regardless of the value of this bit.

3.1.23 MEMREGN[4:1]—EISA-TO-PCI MEMORY REGION ADDRESS REGISTERS

Address Offset: 60-63h (Memory Region 1)
64-67h (Memory Region 2)
68-6Bh (Memory Region 3)
6C-6Fh (Memory Region 4)
Default Value: 0000FFFFh
Attribute: Read/Write
Size: 32 bits

These 32-bit registers provide four windows for EISA-to-PCI memory accesses. Each window defines a positively decoded programmable address region for mapping EISA memory space to the corresponding PCI memory space. This base and limit address fields define the size and location of the region within the 4 GByte PCI memory space. The base and limit addresses can be aligned on any 64 KByte boundary and each region can be sized in 64 KByte increments, up to the theoretical maximum size of 4 GByte. The default values of this register ensure that the regions are initially disabled.

A region is selected based on the following formula: Base Address ≤ address ≤ Limit Address.

Bit	Description
31:16	**Memory Region Limit Address:** For EISA-to-PCI accesses, bits[31:16] correspond to address lines LA[31:16] on the EISA Bus and AD[31:16] on the PCI Bus. This field determines the limit address of the memory region within the 4 GByte PCI memory space.
15:0	**Memory Region Base Address:** For EISA-to-PCI accesses, bits[15:0] correspond to address lines LA[31:16] on the EISA Bus and AD[31:16] on the PCI Bus. This field determines the starting address of the memory region within the 4 GByte PCI memory space.

3.1.24 IOREGN[4:1]—EISA-TO-PCI I/O REGION ADDRESS REGISTERS

Address Offset: 70-73h (I/O Region 1)
74-77h (I/O Region 2)
78-7Bh (I/O Region 3)
7C-7Fh (I/O Region 4)
Default value: 0000FFFCh
Attribute: Read/Write
Size: 32 bits

These 32-bit registers provide four windows for EISA-to-PCI I/O accesses. The windows define positively decoded programmable address regions for mapping EISA I/O space to the corresponding PCI I/O space. Each register determines the starting and limit addresses of the particular region within the 64 KByte PCI I/O space. The base and limit addresses can be aligned on any Dword boundary and each region can be sized in Dword increments (32-bits) up to the theoretical maximum size of 64 KByte. Default values for the base and limit fields ensure that the regions are initially disabled.

The I/O regions are selected based on the following formula: Base Address ≤ address ≤ Limit Address.

Bit	Description
31:18	**I/O Region Limit Address:** For EISA-to-PCI I/O accesses, bits[31:18] correspond to address lines LA[15:2] on the EISA Bus and AD[15:2] on the PCI Bus. This field determines the limit address of the region within the 64 KByte PCI I/O space.
17:16	**Reserved**
15:2	**I/O Region Base Address:** For EISA-to-PCI I/O accesses, bits[15:2] correspond to address lines LA[15:2] on the EISA Bus and AD[15:2] on the PCI Bus. This field determines the starting address of the region within the 65 KByte PCI I/O space.
1:0	**Reserved**

3.1.25 BTMR—BIOS TIMER BASE ADDRESS REGISTER

Address Offset: 80–81h
Default value: 0078h
Attribute: Read/Write
Size: 16 bits

This 16-bit register determines the base address for the BIOS Timer Register located in PCI I/O space. The BIOS Timer resides in the PCEB and is the only internal resource mapped to PCI I/O space. The base address can be set at Dword boundaries anywhere in the 64 KByte PCI I/O space. This register also provides the BIOS Timer access enable/disable control bit.

Bit	Description
15:2	**BIOS Timer Base Address:** Bits[15:2] correspond to PCI address lines AD[15:2].
1	**Reserved**
0	**BIOS Timer Enable (BTE):** When BTE = 1, the BIOS Timer is enabled. When BTE = 0, the BIOS Timer is disabled. The default is 0 (disabled).

3.1.26 ELTCR—EISA LATENCY TIMER CONTROL REGISTER

Address Offset: 84h
Default value: 7Fh
Attribute: Read/Write
Size: 8 bits

This register provides the control for the EISA Latency Timer (ELT). The register holds the initial count value used by the ELT. The ELT uses the PCI clock for counting . The ELT time-out period is equal to:

$$ELT_{timeout} = Value\{ELTCR(7:0)\} \times T_{pciclck}\ [ns]$$

where:

Tpciclk = 30 ns at 33 MHz (40 ns at 25 MHz).

Therefore, a maximum ELT time-out period at 33 MHz is 256 x 30 ns = 7.68 μs. The value written into this register is system dependent. It should be based on PCI latency characteristics controlled by the PCI Master Latency Timer mechanism and on EISA Bus arbitration/latency parametrics. A typical value corresponds to the ELT time-out period of 1–3 μs. When the value in the ELTCR Register is 0, the ELT mechanism is disabled. The ELTCR Register must be initialized before EISA masters or DMA are enabled.

Bit	Description
7:0	**EISA Latency Timer Count Value:** Bits[7:0] contain the initial count value for the EISA Latency Timer. When this field contains 00h, the EISA Latency Timer is disabled.

3.2 I/O Registers

The only PCEB internal resource mapped to the PCI I/O space is the BIOS Timer Register.

3.2.1 BIOSTM—BIOS TIMER REGISTER

Register Location: Programmable I/O address location (Dword aligned)
Default Value: 00 00 xx xxh
Attribute: Read/Write
Size: 32 bits

This 32-bit register is mapped to the PCI I/O space location determined by the value in the BTMR Register. Bit 0 of BTMR must be 1 to enable access to the BIOS Timer. The BIOS timer clock is derived from the EISA Bus clock (BCLK); either 8.25 or 8.33 MHz depending on the PCI clock. BCLK is divided by 8 to obtain the timer clock of 1.03 or 1.04 MHz. If a frequency other than 33 MHz or 25 MHz is used for PCI clock, the BIOS Timer clock will be affected. (It will always keep the same relation to the BCLK, i.e. 1:4 or 1:3, depending on the clock divisor.) The BIOS Timer is only accessible from the PCI Bus and is not accessible from the EISA Bus.

After data is written into BIOS Timer Register (BE1# and/or BE0# must be asserted), the BIOS timer starts decrementing until it reaches zero. It "freezes" at zero until the new count value is written.

Bit	Description
31:16	**Reserved**
15:0	**BIOS Timer Count Value:** The initial count value is written to bits[15:0] to start the timer. The value read is the current value of the BIOS Timer.

4.0 ADDRESS DECODING

Conceptually, the PCEB contains two programmable address decoders: one to decode PCI Bus cycles that need to be forwarded to the EISA Bus or serviced internally and the other to decode EISA Bus cycles that need to be forwarded to the PCI Bus. Two decoders permit the PCI and EISA Buses to operate concurrently (Figure 2). The PCEB can be programmed to respond to certain PCI memory or I/O region accesses as well as configuration space accesses to the PCEB's internal configuration registers. PCEB address decoding is discussed in Section 4.1.

The EISA address decoder decodes EISA Bus cycles generated by the bus master (DMA controller, ISA compatible master, or EISA compatible master) that need to be forwarded to the PCI Bus. The EISA decode logic can be programmed to respond to certain memory or I/O region accesses.

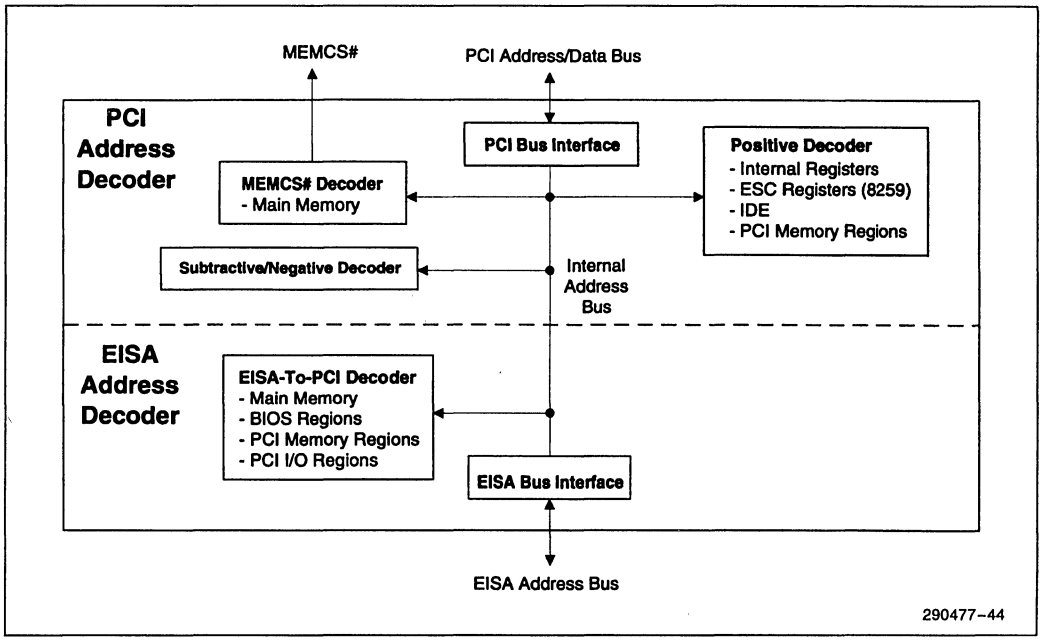


Figure 2. Block Diagram of Address Decoder

The PCEB provides three methods for decoding the current PCI Bus cycle. The PCEB can use positive, subtractive, or negative (82374SB only) decoding for these cycles, depending on the type of cycle, actions on the PCI Bus, and programming of the PCEB registers. For EISA Bus cycles, only positive decoding is used.

1. **Positive decoding.** With positive decoding, the PCI/EISA Bus cycle address is compared to the corresponding address ranges set up in the PCEB for positive decode. A match causes the PCEB decode logic to immediately service the cycle. The PCEB can be programmed (via the configuration registers) to positively decode selected memory or I/O accesses on both the PCI Bus and EISA Bus. Depending on the programming of the internal registers, the PCEB provides positive decoding for PCI accesses to selected address ranges in memory and I/O spaces and for EISA accesses to selected address ranges in memory and I/O spaces. Note that the decoding method for PCI accesses to the PCEB internal registers (configuration and I/O space registers) is not programmable and these accesses are always positively decoded.

2. **Subtractive decoding**. For PCI memory or I/O cycles, the PCEB uses subtractive decoding (or negative decoding, described in #3 of this list for the 82375SB) to respond to addresses that are not positively decoded. With subtractive decoding, if a memory or I/O cycle is not claimed on the PCI Bus (via DEVSEL#), the PCEB forwards the cycle to the EISA Bus. The PCEB waits a programmable number of PCICLKs (1 to 3 PCICLKs, as selected via the PCICON Register) for a PCI agent to claim the cycle. If the cycle is not claimed within the programmed number of PCICLKs (DEVSEL# time-out), the PCEB claims the cycle (asserts DEVSEL#) and forwards it to the EISA Bus. Note that the number of PCICLKs for a DEVSEL# time-out should be programmed to accommodate the slowest PCI Bus device.
3. **Negative decoding**. For the 82375SB, negative decoding is a programmable option (via the PDCON Register) that is only used for PCI memory cycles. With negative decoding, a PCI memory cycle that is not positively decoded by the PCEB as a main memory area (one of the MEMCS# generation areas) and is not in one of the four programmable EISA-to-PCI memory regions (defined by MEMREGN[4:1]) is immediately forwarded to the EISA Bus. This occurs without waiting for a DEVSEL# time-out to see if the cycle is going to be claimed on the PCI Bus. Thus, negative decoding can reduce the latency incurred by waiting for a DEVSEL# time-out that is associated with subtractive decoding. This increases throughput to the EISA Bus for unclaimed PCI memory cycles. If the DEVSEL# time-out is set to a 2 PCICLKtime-out, the latency is reduced by 1 PCICLK and for a 3 PCICLK time-out, the latency is reduced by 2 PCICLKs. For more information on negative (and subtractive) decoding, see Section 4.1.1.3, Subtractively and Negatively Decoded Cycles to EISA.

   Note that negative decoding imposes a restriction on the PCI system memory address map. PCI memory-mapped devices are restricted to one of the four programmable EISA-to-PCI regions (MEMREGN[4:1]). These regions always use subtractive decoding to forward an unclaimed cycle to the EISA Bus, even if negative decoding is enabled. Locating devices in these regions ensures that the PCI device has the allotted number of programmed PCICLKs (DEVSEL# time-out) to respond with DEVSEL#. Further, since the PCEB does not negatively decode I/O space addresses, enabling this feature does not impose restrictions on devices that are mapped to PCI I/O space.

4.1 PCI Cycle Address Decoding

The PCEB decodes addresses presented on the multiplexed PCI address/data bus during the address bus phase. AD[31:0] and the byte enables (C/BE[3:0]# during the data phase) are used for address decoding. C/BE[3:0]# are used during the data phase to indicate which byte lanes contain valid data. For memory cycles, the PCI address decoding is always a function of AD[31:2]. In the case of I/O cycles, all 32 address bits (AD[31:0]) are used to provide addressing with byte granularity. For configuration cycles, only a subset of the address lines carry address information.

The PCEB decodes the following PCI cycle addresses based on the contents of the relevant programmable registers:

1. Positively decodes PCEB configuration registers.
2. Positively decodes I/O addresses contained within the PCEB (BIOS Timer).
3. Positively decodes the following compatibility I/O registers to improve performance: a) Interrupt controller (8259) I/O registers contained within the ESC to optimize interrupt processing, if enabled through the PDCON Register, b) IDE registers, if enabled through the PDCON Register.
4. Positively decodes four programmable memory address regions contained within the PCI memory space.

5. Positively decodes memory addresses for selected regions of main memory (located behind the Host/PCI Bridge). When a main memory address is positively decoded, the PCEB asserts the MEMCS# signal to the Host/PCI Bridge. The PCEB does not assert DEVSEL#.
6. Subtractively or negatively (82375SB only) decodes cycles to the EISA Bus (see Section 4.1.1, Memory Space Address Decoding) .

NOTE:
A PCI requirement is that, upon power-up, PCI agents do not respond to any address. Typically, the only access to a PCI agent is through the IDSEL configuration mechanism until the agent is enabled during initialization. The PCEB/ESC subsystem is an exception to this since it controls access to the BIOS boot code. The PCEB subtractively decodes BIOS accesses and passes the accesses to the EISA Bus where the ESC generates BIOS chip select. This allows BIOS memory to be located in the PCI memory space.

4.1.1 MEMORY SPACE ADDRESS DECODING

The MCSCON, MCSTOP, MCSBOH, MCSTOM, and PDCON Registers are used to program the decoding for PCI Bus memory cycles.

4.1.1.1 Main Memory Decoding (MEMCS#)

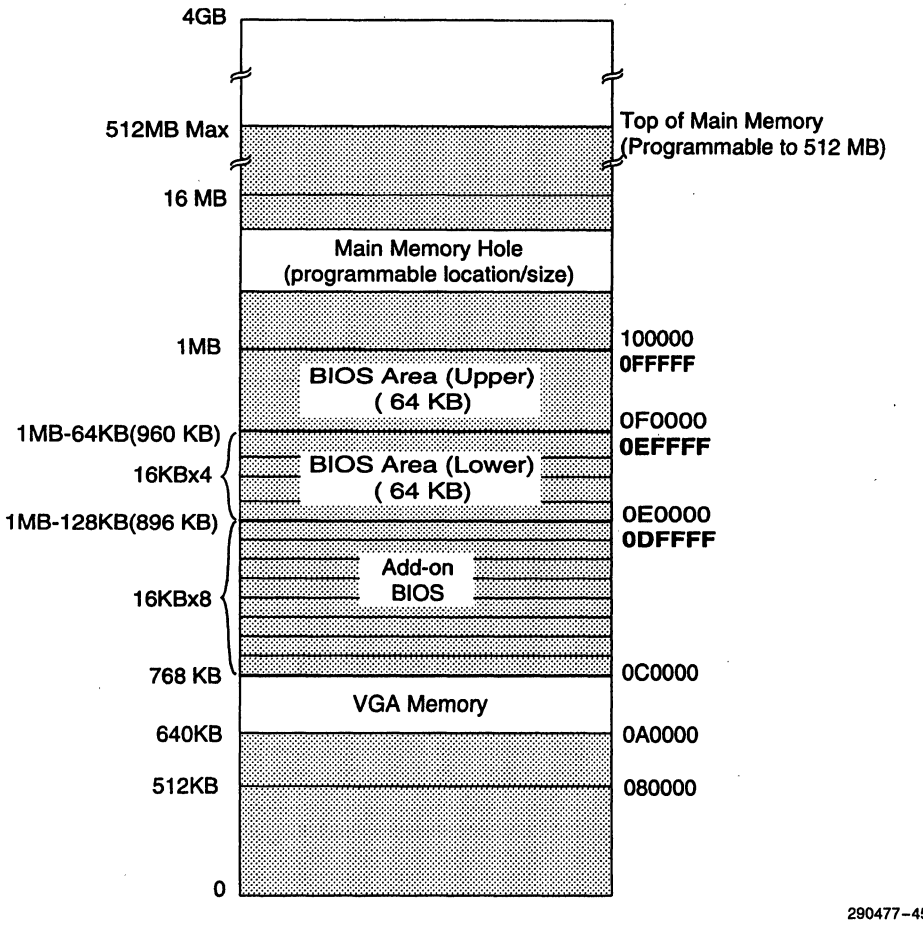
The PCEB supports positive decode of main memory areas by generating a memory chip select signal (MEMCS#) to the Host/PCI Bridge that contains the main memory interface control. The PCEB supports memory sizes up to 512 MBytes (i.e., the PCEB can be programmed to generate MEMCS# for this memory range). For PCI memory accesses above 512 MByte (512 MBytes to 4 GBytes), the PCEB does not generate MEMCS# and unclaimed cycles are forwarded to the EISA Bus using either subtractive or negative (82374SB only) decoding.

If a memory region is enabled, accesses to that region are positively decoded and result in the PCEB asserting MEMCS#. If a memory region is disabled, accesses do not generate MEMCS# and the cycle is either subtractively or negatively (82374SB only) decoded and forwarded to the EISA Bus.

Within the 512 MByte main memory range, the PCEB supports the enabling/disabling of sixteen individual memory ranges (Figure 3). Fourteen of the ranges are within the 640 KByte - 1 MByte area and have Read Enable (RE) and Write Enable (WE) attributes. These attributes permit positive address decoding for reads and writes to be independently enabled/disabled. This permits, for example, an address range to be positively decoded for a memory read and subtractively or negatively (82374SB only) decoded to the EISA Bus for a memory write.

The fifteenth range (0–512 KByte) and sixteenth range (programmable limit address from 2 MByte up to 512 MByte on 2 MByte increments) can be enabled or disabled but do not have RE/WE attributes. A seventeenth range is available that identifies a memory hole. Addresses within this hole will not generate a MEMCS#. These memory address ranges are:

- 0–512 KByte
- 512–640 KByte
- 640–768 KBytes (VGA memory page)
- 960 KByte to 1 MByte (BIOS Area)
- 768–896 KByte in 16 KByte segments (total of 8 segments)
- 896–960 KByte in 16 KByte segments (total of 4 segments)
- 960 KByte to 1 MByte (Upper BIOS area)
- 1–512 MByte in 2 MByte increments.
- Programmable memory hole in 64 KByte increments between 1 MByte and 16 MByte.



NOTE:
Shaded areas represent main memory decode regions for MEMCS#.

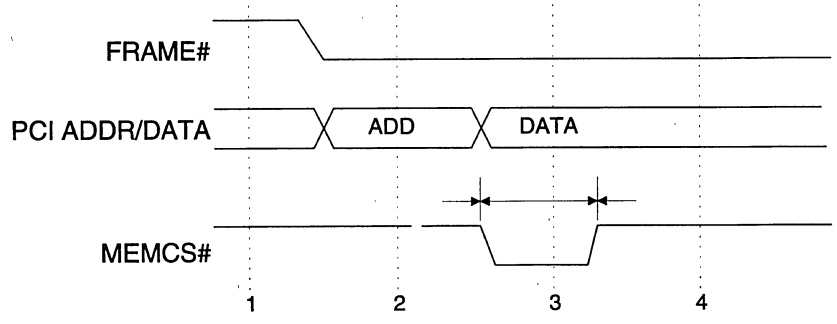
Figure 3. MEMCS# Decode Areas

Table 2 summarizes the attribute registers used in MEMCS# decoding. The MCSCON, MAR1, MAR2, and MAR3 Registers are used to assign RE/WE attributes to a particular memory range. The MEMCS# hole is programmed using the MCSTOH and MCSBOH Registers. The region above 1 MByte is programmed using the MCSTOM Register. The region from 0–512 KByte is enabled/disabled using bit 4 of the MCSCON Register. MCSCON bit 4 is also used to enable and disable the entire MEMCS# function.

Table 2. Read Enable/Write Enable Attributes For MEMCS# Decoding

Memory Attribute Registers (Register Bits are Shown in Brackets)	Attribute	Memory Segments	Comments
MCSCON[1:0]	WE RE	080000–09FFFFh	512K to 640K
MCSCON[3:2]	WE RE	0F0000–0FFFFFh	BIOS Area
MAR1[1:0]	WE RE	0C0000–0C3FFFh	Add-on BIOS
MAR1[3:2]	WE RE	0C4000–0C7FFFh	Add-on BIOS
MAR1[5:4]	WE RE	0C8000–0CBFFFh	Add-on BIOS
MAR1[7:6]	WE RE	0CC000–0CFFFFh	Add-on BIOS
MAR2[1:0]	WE RE	0D0000–0D3FFFh	Add-on BIOS
MAR2[3:2]	WE RE	0D4000–0D7FFFh	Add-on BIOS
MAR2[5:4]	WE RE	0D8000–0DBFFFh	Add-on BIOS
MAR2[7:6]	WE RE	0DC000–0DFFFFh	Add-on BIOS
MAR3[1:0]	WE RE	0E0000–0E3FFFh	BIOS Extension
MAR3[3:2]	WE RE	0E4000–0E7FFFh	BIOS Extension
MAR3[5:4]	WE RE	0E8000–0EBFFFh	BIOS Extension
MAR3[7:6]	WE RE	0EC000–0EFFFFh	BIOS Extension

The PCEB generates MEMCS# from the decode of the PCI address. MEMCS# is asserted during the first data phase as indicated in the Figure 4. MEMCS# is only asserted for one PCI clock period. The PCEB does not take any other action as a result of this decode, except to generate MEMCS#. It is the responsibility of the device using the MEMCS# signal to generate DEVSEL#, TRDY# and any other cycle response. The device using the MEMCS# will always generate DEVSEL# on the next clock. This fact can be used to avoid an extra clock delay in the subtractive decoder described in the next section.



NOTE:
Since MEMCS# is point-to-point (a sideband signal), the signal meets the guaranteed setup time to clock edge 3 (and clock edge 4). This fast generation of MEMCS# prevents the penalty caused by the decoding delay.

Figure 4. MEMCS# Generation

4.1.1.2 BIOS Memory Space

The BIOS memory space is subtractively decoded. BIOS is typically "shadowed" after configuration and initialization is complete. Thus, negative decoding is not implemented for accesses to the BIOS EPROM residing on the expansion bus.

The ESC decoder supports BIOS space up to 512 KBytes. The standard 128 KByte BIOS memory space is 000E 0000h to 000F FFFFh (top of 1 MByte), and aliased at FFFE 0000h to FFFF FFFFh (top of 4 GByte) and FFEE 0000h to FFEF FFFFh (top of 4 GByte -1 MByte). These aliased regions account for the CPU reset vector and the uncertainty of the state of the A20Gate when a software reset occurs.

Note that the ESC component contains the BIOS space decoder that provides address aliasing for BIOS at 4 GByte or 4 GByte - 1 MByte by ignoring the LA20 address line.

The additional 384 KByte BIOS memory space at FFF8 0000h to FFFD FFFFh is known as the enlarged BIOS memory space. Note that EISA memory (other than BIOS) must not reside within the address range from 4 GByte -1.5 MByte to 4 GByte - 1 MByte and from 4 GByte - 512 KByte to 4 GByte to avoid conflict with BIOS space.

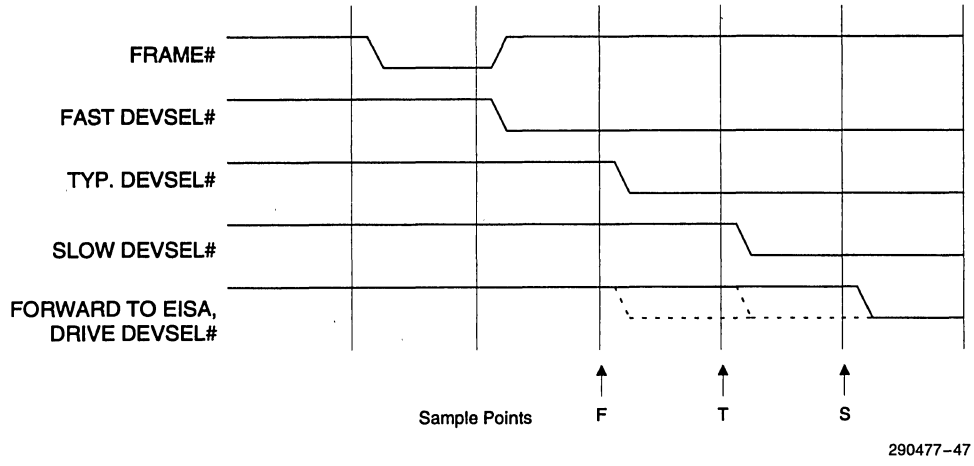
Since the BIOS device is 8 or 16 bits wide and typically has very long access times, PCI burst reads from BIOS space invoke a disconnect target termination (using the STOP# signal) after the first data transaction in order to meet the PCI incremental latency guidelines.

4.1.1.3 Subtractively And Negatively Decoded Cycles To EISA

The PCEB uses subtractive and negative (82375SB only) decoding to forward PCI Bus cycles to the EISA Bus. These modes are defined at the beginning of section 4.0. Bit 0 of the PDCON Register selects between negative and subtractive decoding.

For subtractive decoding on the 82375EB, the DEVSEL# sample point (Figure 5) can be configured to two different settings by programming the PCICON Register. If the "typical" point is selected, DEVSEL# is sampled at T, and, if inactive, the cycle is forwarded to EISA. If the "slow" point is selected, DEVSEL# is sampled at F, T, and S. The sample point should be configured to match the slowest PCI device in the system. This programmable capability (T or S) permits systems to optimize the DEVSEL# time-out latency to the response capabilities of the PCI devices in the system. The sample point selected must accommodate the slowest device on the PCI Bus. Note that when these unclaimed cycles are forwarded to the EISA Bus, the PCEB drives the DEVSEL# active. An active MEMCS# always results in an active DEVSEL# on the "Typical" sample point.

For subtractive decoding on the 82375SB, the DEVSEL# sample point can be configured to two different settings by programming the PCICON Register (slow and typical).



NOTE:
Fast DEVSEL# is not supported on the 82378EB or 82375SB.

Figure 5. DEVSEL# Sample Points

Only unclaimed PCI cycles within the memory address range from 0 to 4 GByte and I/O address range from 0 to 64 KByte are forwarded to EISA. Unclaimed PCI I/O cycles to address locations above 64 KBytes are not forwarded to the EISA Bus and the PCEB does not respond with DEVSEL#. In this case, these unclaimed cycles cause the master to terminate the PCI cycle with a master abort.

For the 82374SB, if negative decoding is used, the PCEB begins the PCI-to-EISA cycle forwarding process at the "fast" sample point. Compared to the system that uses subtractive decode at the "slow" sample point, negative decoding reduces the decoding overhead by 2 PCI clock cycles. In the case of subtractive decode at the "typical" sampling point, negative decoding reduces the overhead by 1 PCI clock.

The PCEB contains programmable configuration registers that define address ranges for PCI resident devices. There is a set of registers associated with MEMCS# decoding of main memory areas and set of registers for defining address mapping of up to four memory regions that are mapped to PCI for EISA Bus initiated cycles. Note that on the 82375EB, there is no equivalent mechanism for mapping the PCI memory regions to EISA and, therefore, all PCI memory cycles that need to be forwarded to the EISA Bus use subtractive decoding.

For the 82374SB, when negative decoding is selected, memory cycles with addresses other than those specified by the MEMCS# mapping for positive decode (via the MCSCON, MCSBOH, MCSTOH, MCSTOM, MAR1, MAR2, and MAR3 Registers) or the four programmable EISA-to-PCI memory regions (via MEMREGN[4:1]) are immediately forwarded to the EISA Bus without waiting for a DEVSEL# time-out.

Negative decoding has the following properties.

- All addresses above the top of main memory or within the MEMCS# hole (as defined by the MEMCS# map) are negatively decoded to EISA, except for the four programmable EISA-to-PCI memory regions. These regions (MEMREGN[4:1]) can overlap with active main memory ranges, the main memory hole, or with the memory space above the top of main memory. PCI accesses to MEMREGN[4:1] are always subtractively decoded to EISA.
- All addresses within MEMCS# defined ranges 640 KByte to 1 MByte can be either mapped to PCI or EISA using positive decoding. Some of these regions allow more detailed mapping based on programmable access attributes (read enable and write enable). This permits a region to be positively decoded for the enabled attribute and negatively decoded, if enabled, to the EISA Bus for the disabled attribute. For example, if a region is enabled for reads and disabled for writes, accesses to the region are positively decoded to the PCI for reads and negatively decoded, if enabled, to EISA for writes. If negative decoding is disabled (i.e., subtractive decoding enabled), the write is subtractively decoded to EISA.
- When negative decoding is enabled, MEMREGN[4:1] can still be set up for subtractive decoding. A PCI device that requires subtractive decoding must reside within Region [4:1]. As a result, the subtractive decoding penalty is only associated with some address ranges (i.e. some devices) and not with all non-PCI ranges. This feature can be used with PCI devices that dynamically change response on PCI cycles based on cycle type or an internal device state (e.g. intervention cycle).

  If a PCI device can not be located in one of the regions (Region [4:1]), then negative decoding can not be used. This could occur for systems with very specific address mapping requirements or systems where the device addresses that reside on the PCI Bus are highly fragmented and could not be accommodated with four regions.

  Note that the four regions do not limit mapping to only four devices. More than one device can be mapped into the same programmable region. These devices will reside within their own sub-regions, which are not necessarily contiguous.

4.1.2 PCEB CONFIGURATION REGISTERS

PCI accesses to the PCEB configuration registers are positively decoded. For a detailed address map of the PCEB configuration registers, see Section 3.1, Configuration Registers.

4.1.3 PCEB I/O REGISTERS

The only I/O-mapped register in the PCEB is the BIOS Timer Register. Section 3.2 provides details on the address mapping of this register. Note that the internal decode of the BIOS Timer Register is disabled after reset and all I/O accesses that are not contained within the PCI are subtractively decoded and passed to EISA Bus. To enable I/O access to the PCEB's BIOS Timer Register, The BTMR Register must be programmed.

4.1.4 POSITIVELY DECODED COMPATIBILITY I/O REGISTERS

The 8259 interrupt controller and IDE register locations are positively decoded. Access to the corresponding I/O address ranges must first be enabled through the PDCON Register.

PCI accesses to these registers are broadcast to the EISA Bus. These PCI accesses require the ownership of the EISA Bus, and will be retried if the EISA Bus is owned by an EISA/ISA master or the DMA.

4.1.4.1 ESC Resident PIC Registers

Access to the 8259 registers are positively decoded, if enabled through PDCON Register, to minimize access time to the system interrupt controller during interrupt processing (in particular during the EOI command sequence). Table 3 shows the 8259 I/O address map. After PCIRST#, positively decoded access to these address ranges is disabled.

Table 3. ESC Resident Programmable Interrupt Controller (PIC) Registers

Address (hex)	Address Bits FEDC	Address Bits BA98	Address Bits 7654	Address Bits 3210	Access Type	Register Name
0020h	0000	0000	001x	xx00	R/W	INT 1 Control
0021h	0000	0000	001x	xx01	R/W	INT 1 Mask
00A0h	0000	0000	101x	xx00	R/W	INT 2 Control
00A1h	0000	0000	101x	xx01	R/W	INT 2 Mask

4.1.4.2 EISA Resident IDE Registers

The PCI address decoder positively decodes IDE I/O addresses (Primary and Secondary IDE) that exist within the EISA subsystem (typically on the X-bus or as an ISA slave). This feature is implemented to minimize the decoding penalty for the systems that use IDE as a mass-storage controller. Table 4 shows IDE's I/O address map. Note that the PDCON Register controls the enable/disable function for IDE decoding. After PCIRST#, positive decode of the IDE address range is disabled.

Table 4. EISA Resident IDE Registers

Address (hex)	Addess (Bits)				Access Type	Register Name
	FEDC	BA98	7654	3210		
0170h	0000	0001	0111	0000	R/W	Secondary Data Register
0171h	0000	0001	0111	0001	R/W	Secondary Error Register
0172h	0000	0001	0111	0010	R/W	Secondary Sector Count Register
0173h	0000	0001	0111	0011	R/W	Secondary Sector Number Register
0174h	0000	0001	0111	0100	R/W	Secondary Cylinder Low Register
0175h	0000	0001	0111	0101	R/W	Secondary Cylinder High Register
0176h	0000	0001	0111	0110	R/W	Secondary Drive/head Register
0177h	0000	0001	0111	0111	R/W	Secondary Status Register
01F0h	0000	0001	1111	0000	R/W	Primary IDE Data Register
01F1h	0000	0001	1111	0001	R/W	Primary Error Register
01F2h	0000	0001	1111	0010	R/W	Primary Sector Count Register
01F3h	0000	0001	1111	0011	R/W	Primary Sector Number Register

Table 4. EISA Resident IDE Registers (Continued)

Address (hex)	Addess (Bits)				Access Type	Register Name
	FEDC	BA98	7654	3210		
01F4h	0000	0001	1111	0100	R/W	Primary Cylinder Low Register
01F5h	0000	0001	1111	0101	R/W	Primary Cylinder High Register
01F6h	0000	0001	1111	0110	R/W	Primary Drive/head Register
01F7h	0000	0001	1111	0111	R/W	Primary Status Register
0376h	0000	0011	0111	0110	R/W	Secondary Alternate Status Register
0377h	0000	0011	0111	0111	R	Secondary Drive Address Register
03F6h	0000	0011	1111	0110	R/W	Primary Alternate Status Register
03F7h	0000	0011	1111	0111	R	Primary Drive Address Register

4.2 EISA Cycle Address Decoding

For EISA Bus cycles, the PCEB address decoder determines the destination of EISA/ISA master and DMA cycles. This decoder provides the following functions:

- Positively decodes memory and I/O addresses that have been programmed into the PCEB for forwarding to the PCI Bus. This includes accesses to devices that reside directly on the PCI (memory Regions [4:1] and I/O Regions [4:1]) and segments of main memory that resides behind the Host/PCI Bridge.
- Provides access attributes for memory Regions [4:1]. These attributes are used to select the most optimum access mode (buffered or non-buffered).
- All cycles that are not positively decoded to be forwarded to PCI are contained within EISA.

NOTE:
The registers that reside in the PCEB (configuration registers and BIOS Timer) are not accessible from the EISA Bus.

4.2.1 POSITIVELY DECODED MEMORY CYCLES TO MAIN MEMORY

The EISA/ISA master or DMA addresses that are positively decoded by the PCEB are forwarded to the PCI Bus. If the address is not positively decoded by the PCEB, the cycle is not forwarded to the PCI Bus. Subtractive and negative (82374SB only) decoding are not used on the EISA Bus.

The PCEB permits several EISA memory address ranges (items a-i) to be positively decoded. EISA Bus cycles to these regions are forwarded to the PCI Bus. Regions described by a-f and h are fixed and can be enabled or disabled independently. These regions are controlled by the EADC1 and EADC2 Registers.

The region described by g defines a space starting at 1 MByte with a programmable upper boundary of 4 GByte - 2 MByte. Within this region a hole can be opened. Its size and location are programmable to allow a hole to be opened in memory space (for a frame buffer on the EISA Bus, for example). The size of this region and the hole are controlled by the MCSTOM, MCSBOH and MCSTOH Registers. If a hole in main memory is defined, then accesses to that address range are contained within EISA, unless defined by the EISA-to-PCI memory regions as a PCI destined access. (See next section.)

a. 0–512 KByte

b. 512–640 KByte

c. 640–768 KByte (VGA memory)

d. 768–896 KByte in eight 16 KByte sections (Expansion ROM)

e. 896–960 KByte in four 16 KByte sections (lower BIOS area)

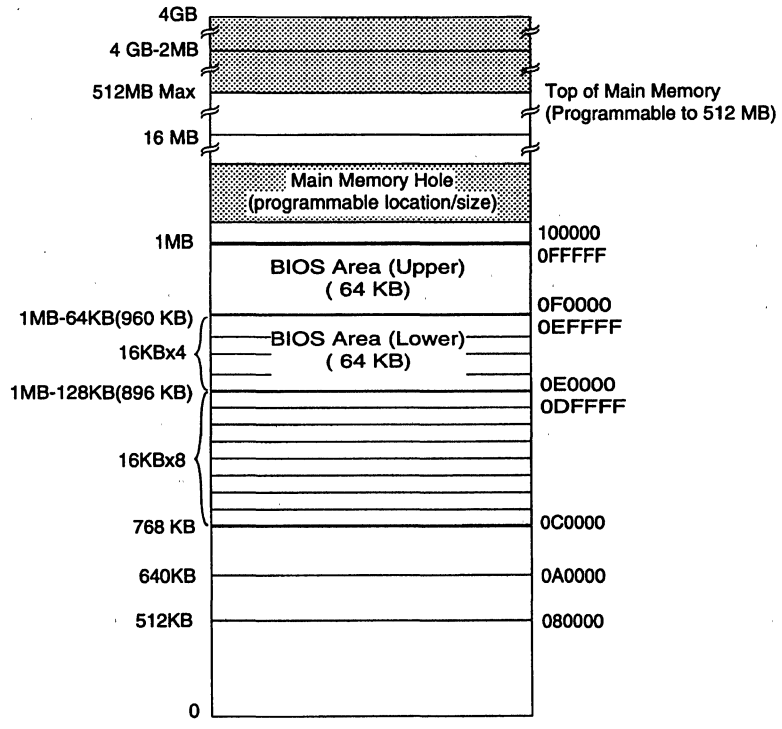
f. 960 KByte to 1 MByte (upper BIOS area)

g. 1 MByte to the top of memory (up to 4 GByte–2 MByte) within which a hole can be opened. Accesses to the hole are not forwarded to PCI. The top of the region can be programmed on 2 MByte boundaries up to 4 GByte–2 MByte. The hole can be between 64 KByte and 4 GByte–2 MByte in 64 KByte increments and located on any 64 KByte boundary.

h. 16 MByte–64 KByte to 16 MByte (FF0000–FFFFFFh). EISA memory cycles in this range are always forwarded to the PCI Bus, if this range exists in main memory as defined by the MEMCS# registers. In this case, the enable/disable control bit in EADC2 Register is a don't care. If this range is not defined in main memory (i.e., above the top of memory or defined as a hole in the main memory), EISA cycles to this address range are forwarded to the PCI Bus, based on the enable/disable bit in the EADC2 Register. (This capability is used to support access of BIOS at 16 MBytes.)

i. 4 GByte–2 MByte to 4 GByte. The address map must be programmed in a such way that this address range is always contained within EISA. This is to avoid conflict with local BIOS memory response in this address range. If this region must be mapped to PCI, then programming of the BIOS decoder Registers contained within the ESC must ensure that there is no conflict. To map this region to PCI, one of the four programmable EISA-to-PCI memory regions must be used. Mapping of this region to the PCI might be required in the case when BIOS resides on the PCI and the PCI/EISA system must have consistent address maps for both PCI and EISA.

For detailed information on the PCEB registers used to control these address regions, refer to Section 3.1, PCEB Configuration Registers.



NOTES:
1. The four programmable EISA-to-PCI regions (Regions[4:1]) are not shown. These regions can be located anywhere within the 4 GByte memory space. Accesses to these regions are forwarded to the PCI Bus.
2. EISA cycles that access shaded regions are contained to the EISA Bus, unless the access hits one of the four programmable EISA-to-PCI regions described in note 1.
3. Memory accesses to non-shaded areas are forwarded to the PCI Bus, if the regions is enabled. If the region is disabled, the access is contained to the EISA Bus, unless the access hits one of the four programmable EISA-to-PCI regions described in note 1.

Figure 6. EISA Address Decoder Map

EISA memory cycles positively decoded for forwarding to PCI are allowed to be handled by the PCEB's Line Buffer management logic, if the line buffering is enabled through the PCICON Register.

For EISA-to-PCI transactions there are 2 modes of operation of the PCEB's Line Buffers:

- Buffered: Read-prefetch, write posting with data assembly.
- Non-buffered: Bypass path used.

Accesses within the main memory address range are normally performed in buffered mode. If there are programmable memory regions defined within the main memory hole or above the top of the main memory MEMREGN[4:1], then the mode of access depends on configuration bits of the EPMRA Register. Access attribute bits associated with these regions override the default buffered mode for a particular address range in the case of programmable regions overlapping with active main memory regions.

Access to the 64 KByte area at the top of 16 MBytes (FF0000–FFFFFFh) on the PCI, if this region is within main memory or within the main memory hole and enabled via the EADC2 Register, are always forwarded in a non-buffered mode, unless overlapped with a programmable region that defines buffered access mode.

4.2.2 PROGRAMMABLE EISA-TO-PCI MEMORY ADDRESS REGIONS

The PCEB supports four programmable memory regions for EISA-to-PCI transfers. The PCEB positively decodes EISA memory accesses to these regions and forwards the cycle to the PCI Bus. This feature permits EISA master accesses to PCI devices that reside within these address ranges.

Regions can be enabled or disabled. After reset, all regions are disabled. Each region has an associated Base and Limit Address fields MEMREGN[4:1] that determine the size and location of each region. These registers are programmed with the starting address of the region (Base) and ending address of the region (Limit). The address range for a particular region is defined by the following equation:

$$Base\_Address \leq Address \leq Limit\_Address$$

These regions can be defined anywhere in the 4 GByte address space at 64 KByte boundaries and with 64 KByte granularity. In practical applications, the regions will be mapped within the main memory hole or above the top of the memory defined by the MEMCS# map.

Access to the memory locations within a region can be performed in one of two modes:

- **Non-Buffered Mode**: PCEB's EISA-to-PCI Line Buffers can be disabled for all EISA-to-PCI memory read/write accesses through the PCICON Register or for selected accesses through EPMRA Register.
- **Buffered Mode**: Line Buffers enabled. Read-prefetch and write-assembly/posting allowed (without strong ordering).

Since buffered mode provides maximum performance (and concurrency in non-GAT mode), it should be selected, unless the particular region is used for memory-mapped I/O devices. I/O devices can not be accessed in read-prefetch or write-assembly/posted fashion because of potential side-effects (see Section 6.0, Data Buffering).

4.2.3 PROGRAMMABLE EISA-TO-PCI I/O ADDRESS REGIONS

The PCEB provides four programmable I/O address regions. These regions are defined by Base and Limit addresses fields contained in the associated IOREGN[4:1] Registers. These regions can be defined anywhere within the 64 KByte I/O space on Dword boundaries (and with Dword granularity). See Section 4.1, PCEB Configuration Registers.

4.2.4 EXTERNAL EISA-TO-PCI I/O ADDRESS DECODER

Since the I/O address map may be highly fragmented, it is impractical to provide enough programmable regions to completely define mapping of registers for I/O devices on the PCI. The PCEB's input signal pin PIODEC# can be used, if a more complex I/O decode scheme is needed. PIODEC# complements the functions of the four PCEB programmable I/O regions with external decode logic. If PIODEC# is asserted during an EISA I/O cycle, the cycle is forwarded to the PCI Bus.

If the PIODEC# signal is not used, a pull-up resistor is required to provide an inactive signal level.

4.3 Palette DAC Snoop Mechanism

Some advanced graphics EISA/ISA expansion boards use the pre-DAC VGA pixel data from the VGA Special Feature Connector and merge it with advanced graphics data (multi-media for example). The merged data is then run through a replicated palette DAC on the advanced graphics expansion board to create the video monitor signal. The replicated palette DAC is kept coherent by snooping VGA palette DAC writes. Snooping becomes an issue in a system where the VGA controller is placed on the PCI Bus and the snooping graphics board is on the EISA expansion bus. Normally, the PCI VGA controller will respond to the palette DAC writes with DEVSEL#, so the PCEB will not propagate the cycle to the EISA Bus using subtractive decoding.

The burden for solving this problem is placed on the VGA subsystem residing on the PCI. The VGA subsystem on PCI must have an enable/disable bit associated with palette DAC accesses. When this bit is enabled the PCI VGA device responds in handshake fashion (generates DEVSEL#, TRDY#, etc.) to I/O reads and writes to the palette DAC space.

When this bit is disabled, the PCI VGA device responds in handshake fashion only to I/O reads to palette DAC space. I/O writes to the palette DAC space will be snooped (data latched) by the PCI VGA device, but the PCI VGA subsystem will not generate a DEVSEL#. In this case, the I/O write will be forwarded to the EISA Bus by the PCEB as a result of subtractive decode. The PCI VGA device must be able to snoop these cycles in the minimum EISA cycle time.

The state of palette-DAC snooping control bit does not affect I/O reads from the palette DAC space. Regardless of whether this bit is enabled or disabled, the PCI VGA device will service the I/O reads from the palette DAC space.

5.0 PCI INTERFACE

The PCEB provides the PCI Interface for the PCI-EISA Bridge. The PCEB can be an initiator (master) or target (slave) on the PCI Bus and supports the basic PCI Bus commands as described in Section 5.1.1, PCI Command Set. For EISA-to-PCI transfers, the PCEB is a master on the PCI Bus on behalf of the requesting EISA device. An EISA device can read and write either PCI memory or I/O space.

The PCEB forwards unclaimed PCI Bus cycles to EISA. For PCI Bus cycles that are not claimed, the PCEB becomes a slave on the PCI Bus (claiming the cycle via subtractive or negative decoding) and forwards the cycle to the EISA Bus. Note that negative decoding is only used on the 82374SB.

This section describes the PCI Bus transactions supported by the PCEB. The section also covers the PCI Bus latency mechanisms in the PCEB that limit a master's time on the bus and the PCEB support of parity. In addition, the PCEB contains PCI Bus arbitration circuitry that supports up to six masters. PCI Bus arbitration is described in Section 5.4.

NOTE:

1. All signals are sampled on the rising edge of the PCI clock. Each signal has a setup and hold window with respect to the rising clock edge, in which transitions are not allowed. Outside of this range, signal values or transitions have no significance.
2. The terms initiator and master are synonymous. Likewise, the terms target and slave are synonymous.
3. Readers should be familiar with the PCI Bus specification.

5.1 PCI Bus Transactions

This section presents the PCI Bus transactions supported by the PCEB.

5.1.1 PCI COMMAND SET

PCI Bus commands indicate to the target the type of transaction requested by the master. These commands are encoded on the C/BE[3:0]# lines during the address phase of a transfer. Table 5 summarizes the PCEB's support of the PCI Bus commands.

Table 5. PCEB-Supported PCI Bus Commands

C/BE[3:0]#	Command Type	Supported As Target	Supported As Initiator
0000	Interrupt Acknowledge	Yes	No
0001	Special Cycle	No	No
0010	I/O Read	Yes	Yes
0011	I/O Write	Yes	Yes
0100	Reserved	N/A(3)	N/A(3)
0101	Reserved	N/A(3)	N/A(3)
0110	Memory Read	Yes	Yes
0111	Memory Write	Yes	Yes
1000	Reserved	N/A(3)	N/A(3)
1001	Reserved	N/A(3)	N/A(3)
1010	Configuration Read	Yes	No
1011	Configuration Write	Yes	No
1100	Memory Read Multiple	No(2)	No
1101	Reserved	N/A(3)	N/A(3)
1110	Memory Read Line	No(2)	No
1111	Memory Write and Invalidate	No(1)	No

NOTES:

1. As a target, the PCEB treats this command as a memory write command.
2. As a target, the PCEB treats this command as a a memory read command.
3. The PCEB considers a reserved command invalid and, as a target, completely ignores the transaction. All internal address decoding is ignored and the PCEB never asserts DEVSEL#. As a PCI master, the PCEB never generates a bus cycle with a reserved command type.

5.1.2 PCI CYCLE DESCRIPTIONS

Each PCI Command is listed below with the following format of information:

Command Type
  PCEB target support
    -Decode method
    -Data path
    -PCEB response
    -Result of no response on EISA
  PCEB initiator support
    -Data path
    -Conditions for generating command
    -Result of no response on PCI

5.1.2.1 Interrupt Acknowledge

Target support:
Decode: Positive
Data Path: Flow through
Response:
The interrupt acknowledge cycle is subject to retry. If the PCEB is locked, or if the interrupt acknowledge cycle triggers buffer management activity, or if the EISA Bus is occupied by an EISA/ISA master or the DMA, the interrupt acknowledge cycle is retried.

The interrupt acknowledge command is a single byte read that is implicitly addressed to the interrupt controller in the ESC component. The address bits are logical "don't cares" during the address phase and the byte enables indicate to the PCEB that an 8-bit interrupt vector is to be returned on byte 0. After performing the necessary buffer management operations and obtaining ownership of the EISA Bus, the PCEB generates a single pulse on the PEREQ#/INTA# inter-chip signal and performs an I/O read cycle (on the EISA Bus) to the ESC internal registers residing at I/O address 04h. The ESC decode logic uses the PEREQ#/INTA# signal to distinguish between standard accesses to I/O address 04h (DMA controller) and special accesses that result in a vector being read by the PCEB. The PCEB holds the PCI Bus, in wait states, until the interrupt vector is returned. PEREQ#/INTA# remains asserted until the end of the read cycle.

Result of no response on EISA:
The PCEB runs a standard length EISA I/O read cycle and terminates normally. The value of the data returned as an interrupt vector is meaningless.

Initiator support: None.

NOTE:
The PCEB only responds to PCI interrupt acknowledge cycles if this operation is enabled via bit 5 of the PCICON Register).

5.1.2.2 Special Cycle

Target support: None.

Initiator support: None.

5.1.2.3 I/O Read

Target support:
Decode: Positive (PCEB and some ESC registers) & Subtractive
Data Path: Flow through
Response:
The PCEB claims I/O read cycles via positive or subtractive decoding and generates DEVSEL#. The internal PCEB registers (BIOS Timer) and the IDE and the 8259 registers are positively decoded. Any unclaimed cycle below 64 KByte is subtractively decoded and forwarded to the EISA Bus. The I/O read cycle is subject to retry. If the PCEB is locked, if the cycle triggers buffer management activity, or if the EISA Bus is occupied by an EISA/ISA master or the DMA, the I/O read cycle is retried. If the cycle gets retried due to an occupied EISA Bus, the EISA Bus is requested.

Once an I/O read cycle is accepted (not retried) by the PCEB, the PCI Bus is held in wait states using TRDY# until the cycle is completed internally or on the EISA Bus.

Burst I/O reads to the EISA Bus or to the PCEB are not supported. Therefore, any burst I/O read cycles decoded by the PCEB are target terminated after the first data transaction using the disconnect semantics of the STOP# signal (Disconnect A).

Result of no response on EISA: The PCEB runs a standard length EISA I/O cycle and terminates normally.

Initiator support:
The PCEB generates PCI Bus I/O read cycles on behalf of an EISA master. EISA cycles are forwarded to the PCI Bus if the I/O address is within one of four programmable I/O address regions as defined in Section 4.0 Address Decoding.

Result of no response on PCI: Master abort due to DEVSEL# time-out. PCEB returns data value FFFFFFFFh.

5.1.2.4 I/O Write

Target support:
Decode: Positive (PCEB/ESC registers) & Subtractive
Data Path: Flow through
Response:
I/O write cycles can be claimed by the PCEB via positive or subtractive decoding. In either case, the PCEB generates DEVSEL#. The internal PCEB registers (BIOS Timer), IDE registers and 8259 registers are positively decoded, if enabled. Any unclaimed cycle below 64 KByte is subtractively decoded and forwarded to the EISA Bus. The I/O write cycle is subject to retry. If the PCEB is locked, if the cycle triggers buffer management activity, or if the EISA Bus is occupied by an EISA/ISA master or the DMA, the I/O write cycle is retried. If the cycle is retried due to an occupied EISA Bus, the EISA Bus is requested.

Once an I/O write cycle is accepted (not retried) by the PCEB, the PCI Bus is held in wait states using TRDY# until the cycle is completed within the PCEB or on the EISA Bus.

Burst I/O writes to the EISA Bus or to the PCEB are not supported. Therefore, any burst I/O write cycles decoded by the PCEB are target terminated after the first data transaction using the disconnect semantics of the STOP# signal (Figure 5-12, Disconnect A).

Result of no response on EISA:
The PCEB runs a standard length EISA I/O cycle and terminates normally.

Initiator support:
The PCEB generates PCI I/O write cycles on behalf of an EISA master. EISA cycles are forwarded to the PCI Bus if the I/O address is within one of the four programmable I/O address regions defined in Section 4.0, Address Decoding.

Result of no response on PCI:
Master abort due to DEVSEL# time-out.

5.1.2.5 Memory Read

Target support:
Decode: Negative (82374SB only) and Subtractive
Data Path: Flow through
PCEB Response:
Memory read cycles may be claimed by the PCEB via negative or subtractive decoding. The PCEB claims the cycle by asserting DEVSEL#. Unclaimed PCI cycles (DEVSEL# time-out) are claimed by the PCEB via subtractively decoding and forwarded to the EISA Bus. The memory read cycle is subject to retry. If the PCEB is locked, if the cycle triggers buffer management activity, or if the EISA Bus is occupied by an EISA/ISA master or the DMA, the memory read cycle is retried. If the cycle is retried due to an occupied EISA Bus, the EISA Bus is requested.

Once a memory read cycle is accepted (not retried) by the PCEB, the PCI Bus is held in wait states, using TRDY#, until the cycle is completed to the EISA Bus.

Incremental burst memory reads destined for the EISA Bus take longer than the allowed 8 PCICLKs. Therefore, any burst memory read cycle decoded by the PCEB causes the PCEB to target terminate the cycle after the first data transaction using the disconnect semantics of the STOP# signal (Figure 5-8, Disconnect A).

Result of no response on EISA:
The PCEB runs a standard length EISA memory read cycle and terminates normally.

Initiator support:
Data Path: Line Buffer when enabled. Flow through when Line Buffer is disabled or it is a bypass cycle.

Cycle Generation Conditions:
As an initiator, the PCEB generates a PCI memory read cycle when it decodes an EISA memory read cycle destined to the PCI that can not be serviced by the Line Buffer. This condition occurs for EISA/ISA master and DMA cycles that can not be serviced by the Line Buffer because the Line Buffer is empty, there is a Line Buffer miss, or Line Buffering is disabled.

As an initiator, the PCEB only generates linear incrementing burst ordering that is signaled by AD[1:0] = 00 during the address phase. Other types of burst transfers (i.e. cache line toggle mode) are never initiated by the PCEB.

The PCEB generates a burst memory read when it is fetching 16 bytes into one of the four Line Buffers.

Result of no response on PCI:
Master abort due to DEVSEL# time-out. PCEB returns data value FFFFFFFFh.

5.1.2.6 Memory Write

Target support:
Decode: Negative (82374SB only) and Subtractive
Data Path: Flow through
PCEB Response:
Memory write cycles may be claimed by the PCEB via negative or subtractive decoding. The PCEB asserts DEVSEL# to claim the cycle. Unclaimed PCI cycles (DEVSEL# time-out) within the 4 GByte memory space are claimed by the PCEB via subtractively decoding and forwarded to the EISA Bus. The memory write cycle is subject to retry. If the PCEB is locked, the cycle triggers buffer management activity. If the EISA Bus is occupied by an EISA/ISA master or the DMA, the memory write cycle is retried. If the cycle is retried due to a disabled buffer because the EISA Bus is occupied, the EISA Bus is requested.

Once a memory write cycle is accepted (not retried) by the PCEB, the PCEB holds the PCI Bus in wait states (using TRDY#) until the cycle is completed on the EISA Bus.

Result of no response on EISA: The PCEB initiates a standard length EISA memory write cycle and terminates normally.

Initiator support:
Data Path: Line Buffer when enabled, flow through when Line Buffer is disabled.
Cycle Generation Conditions:
As an initiator, the PCEB generates a PCI memory write cycle when it decodes an EISA memory write cycle destined to PCI, that can not be serviced by the Line Buffer because it is disabled. This occurs for EISA/ISA masters and DMA cycles when the Line Buffer is disabled. The PCEB also generates a memory write cycle when the Line Buffer needs to be flushed. The Line Buffer is flushed under several conditions, including when the 16 byte line is full, when there is a "miss" to the current 16 byte line, or when it is required by the buffer management logic. (See Section 6.0, Data Buffering).

As an initiator, the PCEB generates only linear incrementing burst ordering that is signaled by AD[1:0] = "00" during address phase. Other types of burst transfers (i.e. cache line toggle mode) are never initiated by the PCEB.

Result of no response on PCI: Master abort due to DEVSEL# time-out.

5.1.2.7 Configuration Read, Configuration Write

Target support:
Decode: via IDSEL pin
Data Path: Flow through
Response:
The PCEB responds to configuration cycles by generating DEVSEL# when its IDSEL signal is asserted, regardless of the address. During configuration cycles, AD[7:2] are used to address the PCEB's configuration space. AD[31:8] are not used and are logical "don't cares". AD[1:0] must be zero.

Result of no response on EISA: N/A

Initiator support:
Configuration cycles are never generated by the PCEB.

5.1.2.8 Memory Read Multiple

Target support:
The PCEB aliases this command to a normal memory read cycle. See the Memory Read command description.

Initiator support:
Memory read multiple cycles are never generated by the PCEB.

5.1.2.9 Memory Read Line

Target support:
The PCEB aliases this command to a normal memory read. See the Memory Read command description.

Initiator support:
Memory read line cycles are never generated by the PCEB.

5.1.2.10 Memory Write And Invalidate

Target support:
Response:
The PCEB treats this command like a memory write. See the Memory Write command description.

Initiator support:
Cycle Generation Conditions:
The PCEB does not generate this command cycle.

5.1.3 PCI TRANSFER BASICS

The basic bus transfer mechanism on the PCI Bus is a burst. A burst is comprised of an address phase and one or more data phases. The PCI protocol specifies the following types of burst ordering (signaled via A[1:0] during the address phase):

A[1:0]	Burst Order
0 0	Linear Incrementing
0 1	Cache line toggle mode
1 X	Reserved

The PCEB only supports linear incrementing burst ordering as an initiator. Data transfers for ordering other than linear incrementing are disconnected by the PCEB (burst split into multiple single data transfers).

The fundamentals of all PCI data transfers are controlled with the following three signals:

- FRAME# is driven by the PCI master to indicate the beginning and end of a transaction.
- IRDY# is driven by the PCI master, allowing it to force wait states.
- TRDY# is driven by the PCI target, allowing it to force wait states.

The PCI Bus is idle when both FRAME# and IRDY# are negated. The first clock edge that FRAME# is sampled asserted is the address phase, and the address and bus command code are transferred on that clock edge. The next clock edge begins the first of one or more data phases. During the data phases, data is transferred between master and slave on each clock edge that both IRDY# and TRDY# are sampled asserted. Wait states may be inserted by either the master (by negating IRDY#) or the target (by negating TRDY#). When a PCI master has one more data transfer to complete the cycle (which could be immediately after the address phase), it negates FRAME#. IRDY# must be asserted at this time, indicating that the master is ready for the final data transfer. After the target indicates the final data transfer (TRDY# asserted), the master negates IRDY#, causing the target's PCI interface to return to the idle state (FRAME# and IRDY# negated), on the next clock edge.

For I/O cycles, PCI addressing is on byte boundaries and all 32 AD lines are decoded to provide the byte address. For memory cycles, AD[1:0] are used to define the type of burst ordering. For configuration cycles, DEVSEL# is strictly a function of IDSEL#. Configuration registers are selected as Dwords using AD[7:2]. The AD[1:0] must be 00 for the target to directly respond to the configuration cycle. The byte enables determine which byte lanes contain valid data.

Each PCI agent is responsible for its own positive address decode. Only one agent (the PCEB) on the PCI Bus may use subtractive decoding. The little endian addressing model is used.

The byte enables are used to determine which bytes carry meaningful data. These signals are permitted to change between data phases. The byte enables must be driven valid from the edge of the clock that starts each data phase and must stay valid for the entire data phase. Figure 7, the data phases begin on clocks 3 and 4. (Changing byte enables during a read burst transaction is generally not useful, but is supported on the bus.) The master is permitted to change the byte enables on each new data phase, although the read diagram does not show this. The timing for changing byte enables is the same for read and write transactions. If byte enables are important for the target on a read transaction, the target must wait for the byte enables to be driven on each data phase before completing the transfer.

5.1.3.1 Turn-Around-Cycle Definition

A turn-around-cycle is required on all signals that may be driven by more than one agent. The turn-around-cycle is required to avoid contention when one agent stops driving a signal and another agent begins, and must last at least one clock. The symbol that represents a turn-around-cycle in the timing relationship figures is a circular set of two lines, each with an arrow that points to the other's tail. This turn-around-cycle occurs at different times for different signals. For example, the turn-around-cycle for IRDY#, TRDY# and DEVSEL# occurs during the address phase and for FRAME#, C/BE# and AD, it occurs during the idle cycle.

5.1.3.2 Idle Cycle Definition

The cycle between clocks 7 and 8 in Figure 8 is called an idle cycle. Idle cycles appear on the PCI Bus between the end of one transaction and the beginning of the next. An idle cycle occurs when both FRAME# and IRDY# are negated.

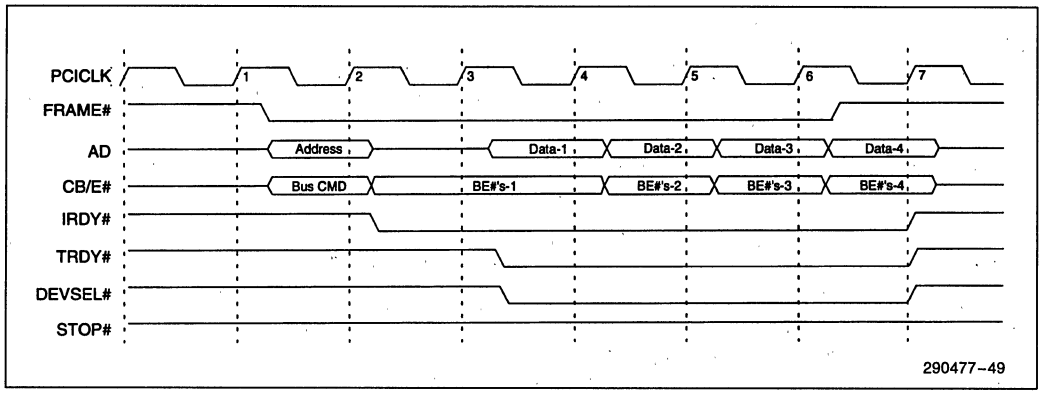


Figure 7. PCEB Burst Read from PCI Memory

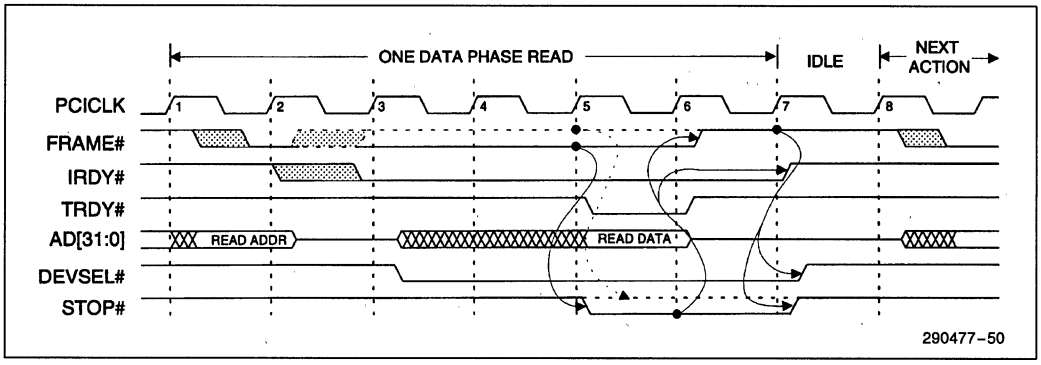


Figure 8. PCI Master Read from the PCEB (Burst with Target Termination)

5.1.4 BASIC READ

As a PCI master, the PCEB performs memory and I/O read transfers. Figure 7 shows a PCEB zero wait state burst read from PCI memory (PCEB is a master). If buffering of memory accesses is enabled, read transfers use prefetching. When reading data from PCI memory, the PCEB requests a minimum of 16 bytes (one data line of the Line Buffer), via a four data phase burst read cycle, to fill one of its internal Line Buffers. The PCEB does not buffer PCI I/O reads and only required data is transferred during these cycles. Read Cycles to PCI are generated on behalf of EISA/ISA masters and DMA devices.

The PCEB asserts FRAME# on clock 1 and places the address on AD[31:2]. CB/E[3:0]# contain a valid bus command. AD[1:0] contain the byte address for I/O cycles, burst order indication for memory cycles, and are 00 for configuration cycles.

The clock following the address phase is the beginning of the data phase. During the data phase, C/BE[3:0]# indicate which byte lanes are involved in the transaction. If the byte lanes involved in the transaction are different for data 1 and data 2, the PCEB drives new C/BE[3:0] values on clock 4. C/BE[3:0]# remain active until the end of the burst transfer.

The first data phase of a read transaction requires a turn-around-cycle, which is enforced by the target preventing the assertion of TRDY# until at least clock 3. The PCEB stops driving the address at clock 2. The target can not drive the AD bus until clock 3. This allows enough time for the PCEB to float its AD outputs. The target is required to drive the AD lines as soon as possible after clock 3, even though valid data may not be ready and the target may want to stretch the initial data phase by delaying TRDY#. This insures that the AD lines are not left floating for long intervals. The target must continue to drive these lines until the end of the burst transaction.

A single data phase is completed when the initiator of the cycle samples TRDY# asserted on the same clock that IRDY# is asserted. To add wait states, the target must negate TRDY# for one or more clock cycles. As a master, the PCEB does not add wait states. In Figure 7, data is transferred on clocks 4 and 5. The PCEB knows, at clock 6, that the next data phase is the last and negates FRAME#. As noted before, the PCEB can burst a maximum of four data cycles when reading from PCI memory.

As a PCI target, the PCEB responds to both I/O and memory read transfers. Figure 8 shows the PCEB, as a target, responding to a PCI master read cycle. For multiple read transactions, the PCEB always target terminates after the first data read transaction by asserting STOP# and TRDY#. These signals are asserted at the end of the first data phase. For single read transactions, the PCEB completes the cycle in a normal fashion (by asserting TRDY# without asserting STOP#). Figure 8 shows the fastest PCEB response to an access of an internal configuration register. During EISA Bus read accesses, the PCEB always adds wait states by negating TRDY# until the transfer on the EISA Bus is completed.

When the PCEB, as a target, samples FRAME# active during a read cycle and positively decodes the cycle, it asserts DEVSEL# on the following clock (clock 3 in Figure 8). Note that, if the PCEB subtractively or negatively (82374SB only) decodes the cycle, DEVSEL# is not asserted for two to three PCICLK's after FRAME# is sampled active. (see Section 5.1.9, Device Selection). When the PCEB asserts DEVSEL#, it also drives AD[31:0], even though valid data is not available. TRDY# is also driven from the same clock edge but it is not asserted until the PCEB is ready to drive valid data. TRDY# is asserted on the same clock edge that the PCEB drives valid data on AD[31:0]. If the PCEB presents valid read data during the first data phase and FRAME# remains active (multiple transaction indicated), the PCEB asserts TRDY# and STOP# to indicate target termination of the transfer.

5.1.5 BASIC WRITE

Figure 9 shows the PCEB, as a master, writing to PCI memory in zero wait states. Figure 10 shows the fastest response of the PCEB, as a target, to a memory or I/O write transaction generated by a PCI master.

As a PCI master, the PCEB performs memory write and I/O transfers. If buffering of memory accesses is enabled, write transfers are posted. When writing data to PCI memory, the PCEB writes a maximum of 16 bytes (one line of the Line Buffer) using a burst write cycle. I/O writes are always non-buffered transactions.

The PCEB generates PCI write cycles on behalf of EISA masters and DMA devices, and when the PCEB flushes its internal Line Buffer.

As a PCI target, the PCEB responds to both I/O and memory write transfers. If the EISA Bus is occupied, the PCI write is retried by the PCEB. When the PCEB owns the EISA Bus, the transaction proceeds. For burst I/O writes, the PCEB always target terminates after the first data transaction by asserting STOP# and TRDY# at the end of the first data phase. During a burst memory write, the PCEB always target terminates after the first data phase.

Figure 10 shows the fastest PCEB response to a write cycle targeted to an internal PCI configuration register. During I/O or memory write accesses to the EISA Bus, the PCEB always adds wait states. The PCEB adds wait states by holding TRDY# high until the transfer on the EISA Bus is completed.

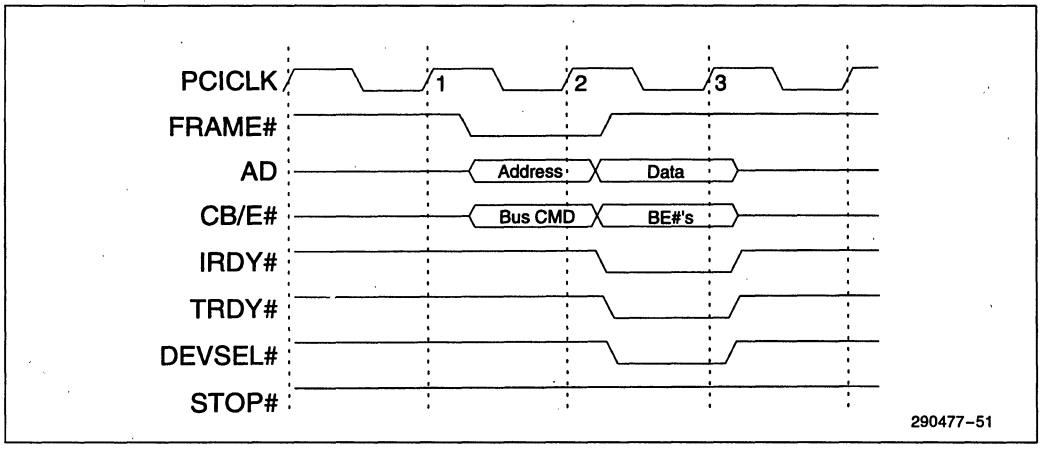


Figure 9. PCEB Write to PCI Memory

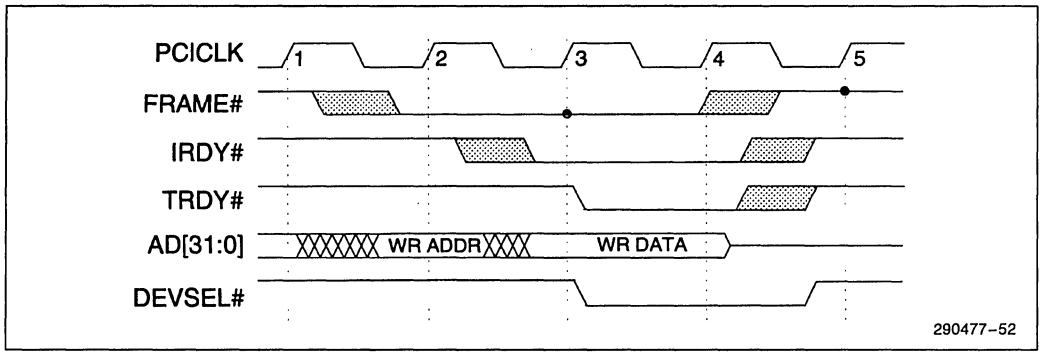


Figure 10. Fastest PCI Write to PCEB

5.1.6 CONFIGURATION CYCLES

One of the requirements of the PCI specification is that upon power up, PCI agents do not respond to any address. The only access allowed is through the IDSEL configuration mechanism. The PCEB is an exception to this since it controls access to the BIOS boot code. All PCEB/ESC subsystem addresses that are enabled after reset are accessible immediately after power up.

The configuration read or write command is used to configure the PCEB. During the address phase of the configuration read or write cycle, the PCEB samples its IDSEL (ID select) signal (not the address lines) to generate DEVSEL#. In this way, IDSEL acts as a chip select. During the address phase, AD[7:2] are used to select a particular configuration register and BE[3:0] to select a particular byte(s). The PCEB only responds to configuration cycles if AD[1:0] = 00. Reference Figure 11 for configuration reads and writes. Note that IDSEL is normally a "don't care", except during the address phase of a transaction. Upon decode of a configuration cycle and sampling IDSEL active, the PCEB responds by asserting DEVSEL# and TRDY#. An unclaimed configuration cycle is never forwarded to the EISA Bus.

Configuration cycles are not normally run in burst mode. If this happens, the PCEB splits the transfer into single cycles using the slave termination mechanism.

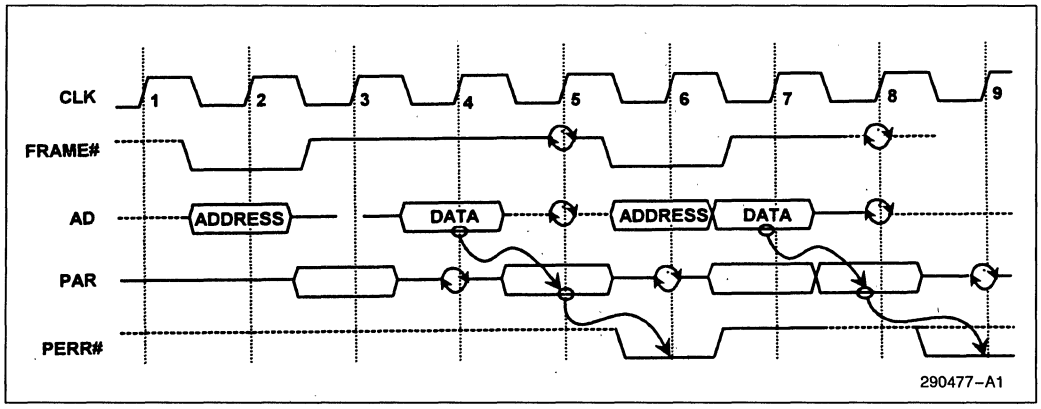


Figure 11. Configuration Cycle

5.1.7 INTERRUPT ACKNOWLEDGE CYCLE

The PCEB responds to an interrupt acknowledge cycle as decoded from the command during a valid address cycle (FRAME# asserted). The AD bus itself is a "don't care" to the PCEB during the address phase and, therefore, status of the internal PCI address decoder is not used for forwarding the cycle to the EISA Bus where the system interrupt controller resides.

The PCEB converts the PCI interrupt acknowledge cycle into an EISA I/O read access to the address 04h, with special semantics indicated to the ESC via the inter-chip signaling. Before the PCI interrupt acknowledge cycle can be converted into an EISA I/O read cycle, the EISA Bus must be owned. If the EISA Bus is not owned by the PCEB (EISAHLDA asserted), the PEREQ#/INTA# signal is asserted with PEREQ# semantics (PCI-to-EISA request). After the EISA Bus is acquired by the PCEB, the interrupt acknowledge sequence can proceed. The PCEB starts an I/O read cycle to address 04h and asserts PEREQ#/INTA# with INTA# semantics. The PEREQ#/INTA# remains asserted for the duration of the EISA I/O read cycle. Therefore, only a single pulse is generated on the PEREQ#/INTA# signal. Conversion of the single PCI interrupt acknowledge cycle into two interrupt acknowledge pulses (that is required for 8259 compatibility) occurs inside the ESC where the 8259-based interrupt controller resides. The ESC's EISA decoder uses the PEREQ#/INTA# signal (with INTA# semantics) to distinguish between normal I/O reads to the register located at address 04h (DMA1 Ch2 Base and Current Address) and the interrupt acknowledge sequence. The ESC holds the EISA Bus in wait states until the interrupt vector is returned to the PCEB (via SD[7:0]). The PCEB passes the vector to the PCI via AD[7:0] and then terminates the cycles both on EISA and PCI. Note that for compatibility reasons, only the ESC (containing the DMA controller) can respond to the EISA I/O read from 04h.

Figure 12 shows the PCI portion of the interrupt acknowledge sequence. The EISA portion of the sequence matches normal EISA I/O read timing, except that the PEREQ#/INTA# inter-chip signal is asserted during the bus cycle with INTA# semantics and, during the PCEB/ESC EISA Bus ownership exchange handshake, with PEREQ# semantics. Note that the PCEB responses to a PCI interrupt acknowledge cycle can be disabled by setting bit 5 in the PCI Control Register (PCICON) to 0.

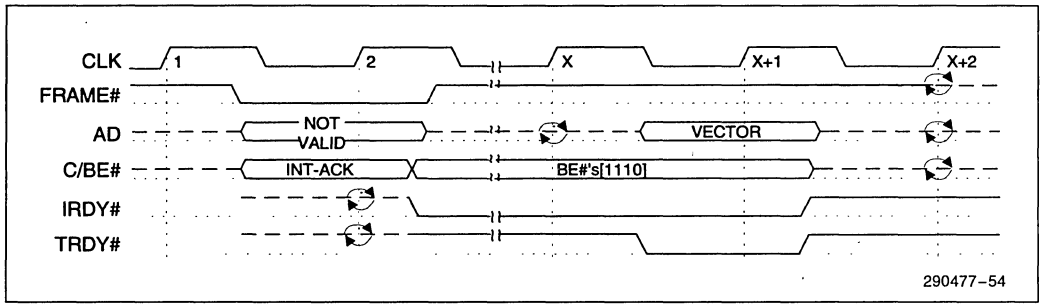


Figure 12. PCI Interrupt Acknowledge Cycle

1

5.1.8 EXCLUSIVE ACCESS

Refer to Figure 13, Figure 14, and Figure 15 for exclusive access timing relationships.

Target support:
PCI provides an exclusive access mechanism that allows non-exclusive accesses to proceed in the face of exclusive accesses. This is referred to as a Resource Lock. (Note that the exclusive access mechanism that locks the entire bus is Bus Lock.) The PCEB, as a resource, can be locked by any PCI initiator. In the context of locked cycles, the PCEB and entire EISA subsystem are considered a single resource. (EISA subsystem is indirectly locked during an exclusive access to the PCEB.) A locked access to any address contained within the EISA subsystem locks the entire subsystem from the PCI side. The PLOCK# signal is propagated to the EISA LOCK# signal. Note that write posting (PCI-to-EISA) is disabled for PCI locked cycles propagated to the EISA subsystem. The EISA Bus is not released to ESC until the locked sequence is complete. A subsequent PCI initiator access to the EISA subsystem, while it is locked, results in a retry. The PCEB becomes locked when it is the target of the access and PLOCK# is sampled negated during the address phase. The PCEB remains locked until FRAME# and PLOCK# are both sampled negated. When in a locked state, the PCEB only accepts requests when PLOCK# is sampled negated during the address phase. If PLOCK# is asserted during the address phase, the PCEB responds by asserting STOP# with TRDY# negated (RETRY).

As an unlocked target, the PCEB ignores PLOCK# when deciding if it should respond to a PCI address decoder hit. Also, if PLOCK# is sampled asserted during an address phase, the PCEB does not go into a locked state.

As a locked target, the PCEB responds to an initiator when it samples PLOCK# negated during the address phase of the cycle in which the PCEB is the target of the access. The locking master may negate PLOCK# at the end of the last data phase. When FRAME# and PLOCK# are both sampled negated, the PCEB goes to the unlocked state.

Note that the PCEB does not release the EISA Bus when it is in the locked state.

Initiator support:
When an EISA locked access to the PCI is encountered (EISA LOCK# asserted), the cycle is propagated to the PCI Bus as a PCI locked cycle. Line Buffers in the PCEB are bypassed. The PLOCK# signal must be negated (released) before an EISA agent can be granted the EISA Bus. Thus, when the PCEB acquires the PCI Bus on behalf of the EISA agent, a PCI LOCKED cycle can be performed, if needed.

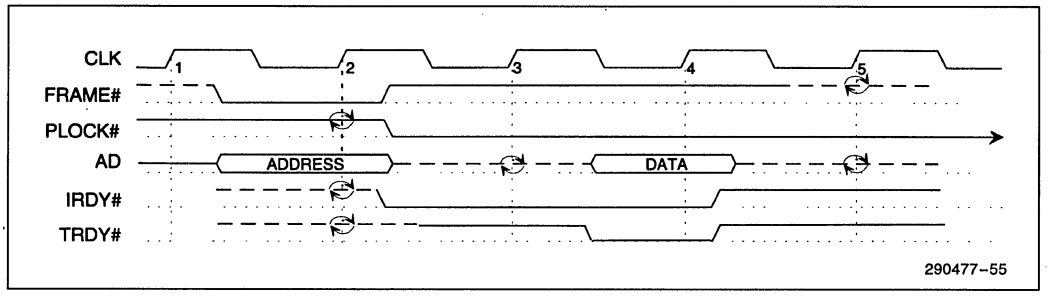


Figure 13. Beginning a Locked Cycle

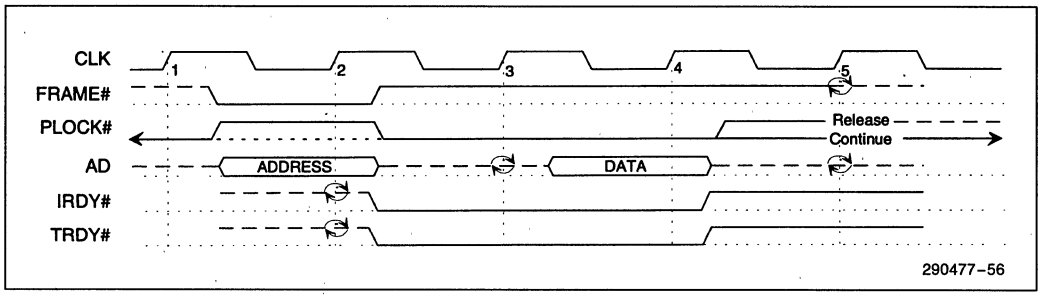


Figure 14. Continuing Locked Cycle

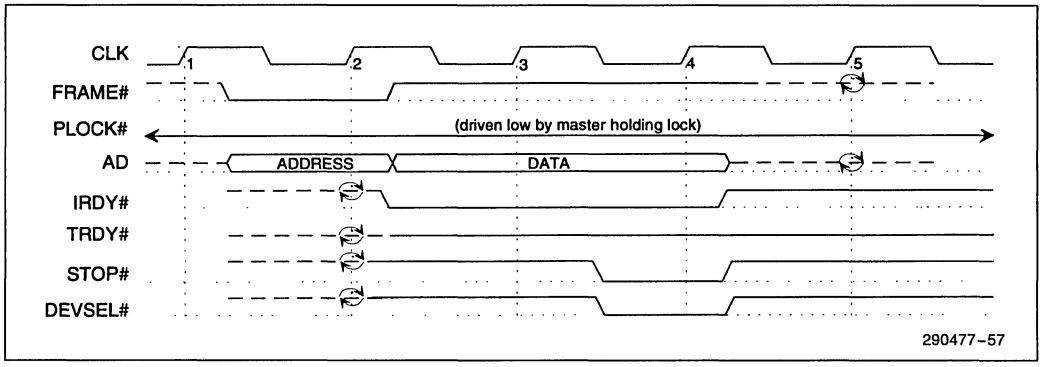


Figure 15. Access to Locked Target with PLOCK# Asserted During Address Phase

1

5.1.9 DEVICE SELECTION

The PCEB asserts DEVSEL# to indicate that it is the target of the PCI transaction. DEVSEL# is asserted when the PCEB, as a target, positively, subtractively, or negatively (82374SB only) decodes the PCI transaction. In all cases except one, once the PCEB asserts DEVSEL#, the signal remains asserted until FRAME# is negated (IRDY# is asserted) and either STOP# or TRDY# is asserted. The exception is a target abort, described in Section 5.1.10, Transaction Termination.

For most systems, PCI target devices are able to complete a decode and assert DEVSEL# within 2 or 3 clocks of FRAME# (medium and slow in the Figure 16). Accordingly, since the PCEB subtractively or negatively (82374SB only) decodes all unclaimed PCI cycles (except configuration cycles), it provides a configuration option to reduce by 1 clock the edge at which it samples DEVSEL#, allowing faster access to the expansion bus. Use of this option is limited by the slowest positive decode agent on the bus. This is described in more detail in Section 4.0, Address Decoding.

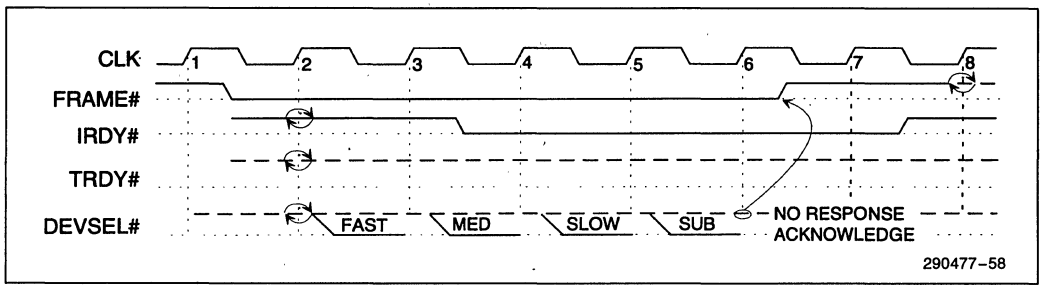


Figure 16. Device Selection (DEVSEL#)

5.1.10 TRANSACTION TERMINATION

Termination of a PCI cycle can be initiated by either a master or a target. The PCEB supports both master and target initiated termination. All transactions are concluded when FRAME# and IRDY# are both sampled negated, indicating that the PCI Bus is idle.

5.1.10.1 Master Initiated Termination

The PCEB supports three types of master initiated termination:

- **Completion:** Refers to the termination when the PCEB finishes the transaction normally. This is the most common type of termination.
- **Time-out:** Refers to termination when the PCEB's GNT# line is negated and its internal Master Latency Timer has expired. The intended transaction is not necessarily concluded. The timer may have expired because of a target-induced access latency, or because the intended operation was very long.
- **Abort:** Refers to termination when there is no target response (no DEVSEL# asserted) to a transaction within the programmed DEVSEL# response time.

Completions and time-outs are common while the abort is an abnormal termination. A normal termination of this type can be seen in Section 5.1.4 and 5.1.5 in the descriptions of the basic PCI read and write transaction.

The PCEB sends out a master abort (Figure 17) when the target does not respond to the PCEB-initiated transaction by asserting DEVSEL#. The PCEB checks DEVSEL# based on the programmed DEVSEL# sample point. If DEVSEL# is not asserted by the programmed sample point, the PCEB aborts the transaction by negating FRAME#, and then, one clock later, negating IRDY#. The master abort condition is abnormal and it indicates an error condition. The PCEB does not retry the cycle.

If the transaction is an EISA-to-PCI memory or I/O write, the PCEB terminates the EISA cycle with EXRDY. If the transaction is an EISA-to-PCI memory or I/O read, the PCEB returns FFFFFFFFh on the EISA Bus. This is identical to the way an unclaimed cycle is handled on the "normally ready" EISA Bus. If the Line Buffer is the requester of the PCI transaction, the master abort mechanism ends the PCI cycle, but no data is transferred into or out of the Line Buffer. The Line Buffer does not retry the cycle. The Received Master Abort Status bit in the PCI Status Register is set to 1 indicating that the PCEB issued a master abort.

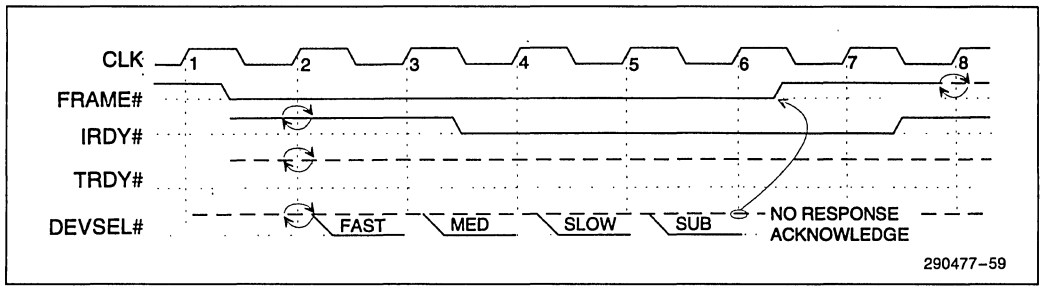


Figure 17. Master Initiated Termination (Master Abort)

1

5.1.10.2 Target Initiated Termination

The PCEB supports two forms of target-initiated termination:

- **Disconnect:** A disconnect termination occurs when the target is unable to respond within the latency guidelines of the PCI specifications. Note that this is not usually done on the first data phase.
- **Retry:** Retry refers to a termination requested because the PCEB is currently in a state that makes it unable to process the transaction.

Figure 18 and Figure 19 show four types of target-initiated terminations. The PCEB initiates a disconnect for PCI cycles destined to EISA after the first data phase due to incremental latency requirements. The difference between disconnect and retry is that the PCEB does not assert TRDY# for the retry case. This instructs the initiator to retry the transfer at a later time. No data is transferred in a retry termination since TRDY# and IRDY# are never both asserted. The PCEB retries a PCI initiator when:

- the PCEB buffers require management activity.
- the PCEB is locked and another PCI device attempts to select the PCEB without negating PLOCK# during the address phase.
- the EISA Bus is occupied by an EISA/ISA master or DMA.

Target abort is another form of target-initiated termination. Target abort resembles a retry, though the target must also negate DEVSEL#, along with assertion of STOP#. As a target, the PCEB never generates a target abort.

As a master, If the PCEB receives a target abort, it relinquishes the PCI Bus and sets the Received Target Abort Status bit in the PCI Status Register to a 1.

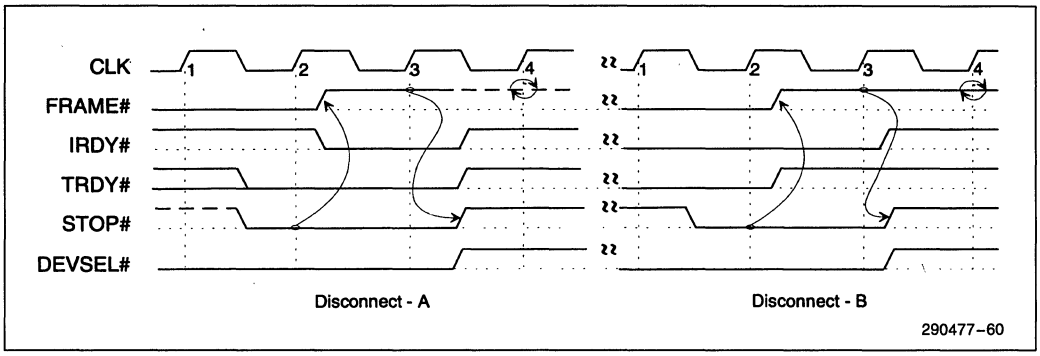


Figure 18. Target Initiated Termination

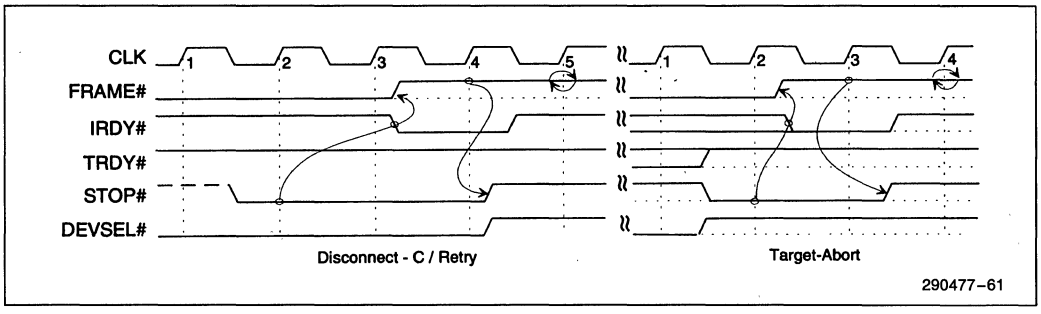


Figure 19. Target Initiated Termination

5.1.10.3 PCEB Target Termination Conditions

As a target, the PCEB terminates a transaction due to the following conditions:

Disconnect

- When a target, the PCEB always responds with a disconnect to a multiple data phase transaction (see the Incremental Latency Timer section),

Retry

- When the pending PCI cycle initiates buffer management activity.
- When the PCEB is locked, as a resource, and a PCI master tries to access the PCEB without negating the PLOCK# signal in the address phase.
- When the EISA Bus is occupied by an EISA/ISA master or DMA.

Target Abort

- The PCEB never generates a target abort.

5.1.10.4 PCEB Master Termination Conditions

As an initiator, the PCEB terminates a transaction due to the following conditions:

- Completion termination is always used by the PCEB signaling to the target that the PCEB is ready to complete the final data phase of the transaction.
- Master abort termination is issued if the PCEB does not receive a DEVSEL# from a target within five PCICLK's after FRAME# assertion. The PCEB sets the Received Master Abort Status bit in the PCI Status Register to a 1.
- Master initiated termination (disconnect) due to Master Latency Timer expiration when the PCEB's PCI Bus grant is removed (internal PCEBGNT# negated).

5.1.10.5 PCEB Responses/Results Of Termination

PCEB's response, as a target, to a master termination:

- Completion termination is the normal way of terminating a transaction.
- If a PCI initiator times out due to LT time-out and ends the current transaction, the PCEB cannot detect a difference between normal completion termination and time-out forced termination.

PCEB's response as a master to target termination:

- If the PCEB receives a target abort, it means that the target device is not capable of handling the transaction. The PCEB does not try the cycle again. If an EISA/ISA master or the DMA is waiting for the PCI cycle to terminate (EXRDY negated), the target abort condition causes the PCEB to assert EXRDY to terminate the EISA cycle. Note that write data is lost and the read data is meaningless. This is identical to the way an unclaimed cycle is handled on the "normally ready" EISA Bus. If the Line Buffer is the requester of the PCI transaction, the target abort mechanism ends the PCI cycle, but no valid data transfers are performed into or out of the Line Buffer. The Line Buffer does not try the cycle again. The Received Target Abort Status bit in the PCI Status Register is set to 1 indicating that the PCEB experienced a target abort condition.
- If the PCEB is retried as an initiator on the PCI Bus, it will remove it's request for 2 PCI clocks before asserting it again to retry the cycle.

- If the PCEB is disconnected as an initiator on the PCI Bus, it will respond very much as if it had been retried. The difference between retry and disconnect is that the PCEB did not see any data phase for the retry. Disconnect may be generated by a PCI slave when the PCEB is running a burst memory cycle to empty or to fill one line (16-byte) of the Line Buffers. In this case, the PCEB may need to finish a multi-data phase transfer and recycles through arbitration as required for a retry. An example is when an EISA agent (EISA/ISA master or DMA) issues a read request that the PCEB translates into a 16 byte prefetch (one line) and the PCEB is disconnected before the Line Buffer is completely filled.

5.1.11 PCI DATA TRANSFERS WITH SPECIFIC BYTE ENABLE COMBINATIONS

Non-Contiguous Combination of Byte Enables
As a master, the PCEB might generate non-contiguous combinations of data byte enables because of the nature of assembly operations in the Line Buffers.

As a target, the PCEB might need to respond to a non-contiguous combination of data byte enables. These cycles can not be passed directly to the EISA Bus; the EISA Bus specification does not allow non-contiguous combinations of byte enables. If this situation occurs, the PCEB splits the 32-bit transactions into two 16-bit transactions by first performing the lower word transfer (indicated by BE1# and BE0#) and then the upper word transfer (indicated by BE3# and BE2#).

BE[3:0]# = 1111
As a master, the PCEB might generate this combination of data byte enables during Line Buffer flush operations (burst write) to optimize the usage of the PCI Bus. Correct parity is driven during this transaction on the PCI Bus.

As a target, the PCEB might need to respond to this combination of data byte enables. If BE[3:0]# = 1111, the PCEB completes the transfer by asserting TRDY# and providing parity for read cycles. The PCEB does not forward the cycle to the EISA Bus.

5.2 PCI Bus Latency

The PCI specification provides two mechanisms that limit a master's time on the bus. They ensure predictable bus acquisitions when other masters are requesting bus access. These mechanisms are master-initiated termination supported by a Master Latency Timer (MLT) and a target-initiated termination (specifically, disconnect) supported by a target's incremental latency mechanism.

5.2.1 MASTER LATENCY TIMER (MLT)

The PCEB has a programmable Master Latency Timer (MLT). The MLT is cleared and suspended when the PCEB is not asserting FRAME#. The MLT is controlled via the MLT Register (see Section 4.1, PCEB Configuration Registers). When the PCEB, as a master, asserts FRAME#, it enables its MLT to count. If the PCEB completes its transaction (negates FRAME#) before the count expires, the MLT is ignored. If the count expires before the transaction completes (count = number clocks programmed into the MLT Register), the PCEB initiates a transaction termination as soon as its GNT# is removed. The number of clocks programmed into the MLT Register represents the guaranteed time slice (measured in PCICLKs) allotted to the PCEB; after which it surrenders the bus as soon as its GNT# is removed. (Actual termination does not occur until the target is ready.) Each master on PCI contains a master latency timer. The relative values programmed in each master timer determines how much of the PCI bandwidth is available to that master. Generally, if the EISA bus is heavily loaded with masters, the PCEB MLT register would be programmed with a relatively large value to give the PCEB a larger share of the PCI bus bandwidth.

5.2.2 INCREMENTAL LATENCY MECHANISM

As a target, the PCEB supports the Incremental Latency Mechanism for PCI-to-EISA cycles. The PCI specification states that for multi-data phase PCI cycles, if the incremental latency from current data phase (N) to the next data phase (N+1) is greater than eight PCICLKs, the target must manipulate TRDY# and STOP# to stop the transaction after the current data phase (N). All PCI-to-EISA cycles (memory read/write and I/O read/write) are automatically terminated (during a burst) after the first data phase because they require more than eight PCICLKs to complete on the EISA Bus.

Therefore, the PCEB does not need to specifically implement an 8 PCICLK timer and the PCEB handles a disconnect in a pre-determined fashion, based on the type of current transaction.

5.3 PCI Bus Parity Support And Error Reporting

PCI provides for parity and asynchronous system errors to be detected and reported separately. The PCEB/ESC chip set implements both mechanisms. The PCEB implements only parity generation and checking and it does not interface to the SERR# signal. Reporting of both PERR# and SERR# indicated errors is implemented in the ESC.

5.3.1 PARITY GENERATION AND CHECKING

The PCEB supports parity generation and checking on the PCI Bus. During the address and data phases, parity covers AD[31:0] and the C/BE[3:0]# lines, regardless of whether or not all lines carry meaningful information. Byte lanes that are not actually transferring data are still required to be driven with stable (albeit meaningless) data and are included in the parity calculation. Parity is calculated such that the number of 1s on AD[31:0], C/BE[3:0]#, and the PAR signals is an even number.

The role of the PCEB in parity generation/checking depends on the phase of the cycle (address or data), the type of bus cycle (read or write), and whether the PCEB is a master or target. The following paragraphs and Figure 20 summarize the behavior of the PCEB during the address and data phase of a PCI Bus cycle.

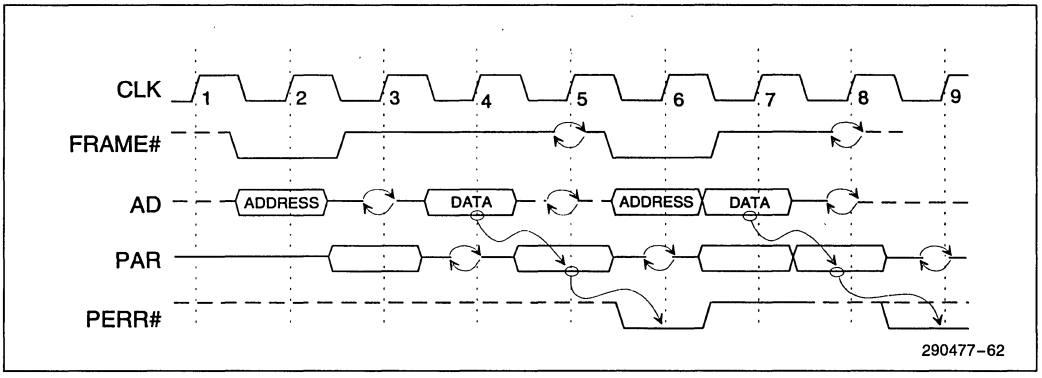


Figure 20. Parity Operation

5.3.1.1 Address Phase

As a master, the PCEB drives AD[31:0] and C/BE[3:0]# and calculates the corresponding parity value and drives it on the PAR signal, 1 clock later. As a target, the PCEB does not check parity during the address phase of a bus cycle.

5.3.1.2 Data Phase

As a master during a write cycle, the PCEB drives AD[31:0] and C/BE[3:0]# and calculates the corresponding parity value and drives it on the PAR signal, 1 clock later.

As a master during a read cycle, the PCEB only drives C/BE[3:0]#. The responding target drives AD[31:0] lines (data) and calculates parity based on the received C/BE[3:0]# and outgoing AD[31:0] signals. The target drives PAR during the following clock. The PCEB calculates parity based on the outgoing C/BE[3:0]# and the incoming AD[31:0] signals at the end of the data phase. It compares it with the incoming value of the PAR signal and asserts PERR# if there is no match.

As a target during a write cycle, the PCEB calculates parity on the incoming AD[31:0] and C/BE[3:0]# signals, and compares the result on the next clock with the incoming value on the PAR signal. If the value does not match, the PCEB asserts PERR#.

As a target during a read cycle, the PCEB calculates parity on the incoming C/BE[3:0]# and outgoing AD[31:0] signals. The PCEB drives the calculated parity value during the next clock. The master of the transaction receives the data, calculates parity on its outgoing C/BE[3:0]# and incoming AD[31:0] signals and compares its calculated value, on the next clock, with the parity value on the PAR signal (supplied by the PCEB). If the values do not match, the master asserts PERR#.

5.3.2 PARITY ERROR—PERR# SIGNAL

When the PCEB is involved in a bus transaction (master or target), it asserts the PERR# signal, if enabled via the PCICMD Register, to indicate a parity error for the bus cycle. PERR# is a sustained tri-state (s/t/s) type of signal (see Section 2.0, Signal Description). Note that PCI parity errors signaled by PERR#, are reported to the host processor via the ESC's system interrupt control logic. When the PCEB detects a parity error during one of its bus transactions, it sets the parity error status bit in the PCI Status Register, regardless of whether the PERR# signal is enabled via the PCICMD Register.

5.3.3 SYSTEM ERRORS

The PCEB does not generate system errors (SERR#). Thus, the PCEB does not have the capability of indicating parity errors during the address phase in which it is a potential target (i.e. not a master). Note that system errors are reported via the ESC (companion chip).

5.4 PCI Bus Arbitration

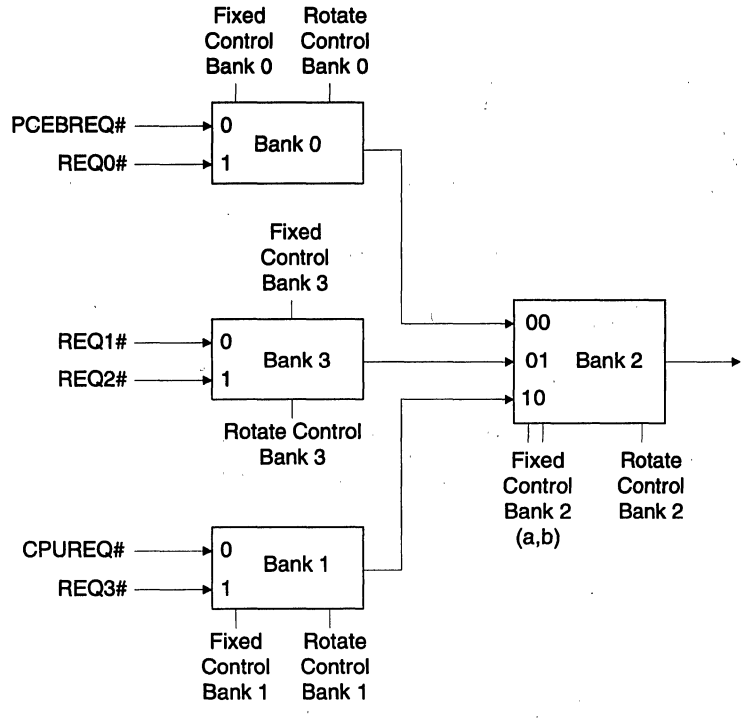
The PCEB contains a PCI Bus arbiter that supports six PCI Bus mastersThe Host/PCI Bridge, PCEB, and four other masters. The PCEB's REQ#/GNT# signals are internal. An external arbiter is not supported. Note that, for proper arbiter operation, CPUREQ# must be sampled high by the PCEB when PCIRST# makes a low-to-high transition. The internal arbiter contains several features that contribute to system efficiency:

- Use of the internal RESUME# signal to re-enable a backed-off initiator in order to minimize PCI Bus thrashing when the PCEB generates a retry.
- A programmable timer to re-enable retried initiators after a number of PCICLK's.
- A programmable PCI Bus lock or PCI resource lock function.
- The CPU Host/PCI can be optionally parked on the PCI Bus.

In addition, the PCEB has three PCI sideband signals (FLUSHREQ#, MEMREQ#, and MEMACK#) that are used to control system buffer coherency and control operations for the Guaranteed Access Time (GAT) mode.

5.4.1 PCI ARBITER CONFIGURATION

The PCI arbitration priority scheme is programmable through the configuration registers. The arbiter consists of four banks that can be configured so that the six masters to be arranged in a purely rotating priority scheme, one of 24 fixed priority schemes, or a hybrid combination (Figure 21).



NOTE:
PCEBREQ#/PCEBGNT# are PCEB internal signals.

Figure 21. Arbiter Conceptual Block Diagram

The PCEB implements PCI arbiter priority configuration registers ARBPRI and ARBPRIX mapped in the PCI's configuration space. Definition of the registers are as follows:

ARBPRI Register

Bit	Description
7	Bank 3 Rotate Control
6	Bank 2 Rotate Control
5	Bank 1 Rotate Control
4	Bank 0 Rotate Control
3	Bank 2 Fixed Priority Mode Select B
2	Bank 2 Fixed Priority Mode Select A
1	Bank 1 Fixed Priority Mode Select
0	Bank 0 Fixed Priority Mode Select

This register defaults to 04h at reset. This selects fixed mode #4 with the CPU the highest priority device guaranteeing that BIOS accesses can take place.

ARBPRIX Register

Bit	Description
7	Bank 3 Fixed Priority Mode select
6:0	Reserved

This register defaults to 00h at reset. The default value selects REQ1# as a higher priority request than REQ2# when Bank 3 operates in the fixed priority mode.

5.4.1.1 Fixed Priority Mode

The twelve selectable fixed priority schemes are listed in Table 6.

Table 6. Fixed Priority Mode Bank Control Bits

Mode	Bank					Priority				
	3	2b	2a	1	0	Highest				Lowest
0	0	0	0	0	0	PCEBREQ#	REQ0#	REQ1#/ REQ2#	CPUREQ#	REQ3#
1	0	0	0	0	1	REQ0#	PCEBREQ#	REQ1#/ REQ2#	CPUREQ#	REQ3#
2	0	0	0	1	0	PCEBREQ#	REQ0#	REQ1#/ REQ2#	REQ3#	CPUREQ#
3	0	0	0	1	1	REQ0#	PCEBREQ#	REQ1#/ REQ2#	REQ3#	CPUREQ#
4	0	0	1	0	0	CPUREQ#	REQ3#	PCEBREQ#	REQ0#	REQ1#/ REQ2#
5	0	0	1	0	1	CPUREQ#	REQ3#	REQ0#	PCEBREQ#	REQ1#/ REQ2#
6	0	0	1	1	0	REQ3#	CPUREQ#	PCEBREQ#	REQ0#	REQ1#/ REQ2#
7	0	0	1	1	1	REQ3#	CPUREQ#	REQ0#	PCEBREQ#	REQ1#/ REQ2#
8	0	1	0	0	0	REQ1#/ REQ2#	CPUREQ#	REQ3#	PCEBREQ#	REQ0#
9	0	1	0	0	1	REQ1#/ REQ2#	CPUREQ#	REQ3#	REQ0#	PCEBREQ#
A	0	1	0	1	0	REQ1#/ REQ2#	REQ3#	CPUREQ#	PCEBREQ#	REQ0#
B	0	1	0	1	1	REQ1#/ REQ2#	REQ3#	CPUREQ#	REQ0#	PCEBREQ#

Table 6. Fixed Priority Mode Bank Control Bits (Continued)

Mode	Bank					Priority				
	3	2b	2a	1	0	Highest				Lowest
	x	1	1	x	x	Reserved				
10	1	0	0	0	0	PCEBREQ#	REQ0#	REQ2#/ REQ1#	CPUREQ#	REQ3#
11	1	0	0	0	1	REQ0#	PCEBREQ#	REQ2#/ REQ1#	CPUREQ#	REQ3#
12	1	0	0	1	0	PCEBREQ#	REQ0#	REQ2#/ REQ1#	REQ3#	CPUREQ#
13	1	0	0	1	1	REQ0#	PCEBREQ#	REQ2#/ REQ1#	REQ3#	CPUREQ#
14	1	0	1	0	0	CPUREQ#	REQ3#	PCEBREQ#	REQ0#	REQ2#/ REQ1#
15	1	0	1	0	1	CPUREQ#	REQ3#	REQ0#	PCEBREQ#	REQ1#/ REQ2#
16	1	0	1	1	0	REQ3#	CPUREQ#	PCEBREQ#	REQ0#	REQ2#/ REQ1#
17	1	0	1	1	1	REQ3#	CPUREQ#	REQ0#	PCEBREQ#	REQ2#/ REQ1#
18	1	1	0	0	0	REQ2#/ REQ1#	CPUREQ#	REQ3#	PCEBREQ#	REQ0#
19	1	1	0	0	1	REQ2#/ REQ1#	CPUREQ#	REQ3#	REQ0#	PCEBREQ#
1A	1	1	0	1	0	REQ2#/ REQ1#	REQ3#	CPUREQ#	PCEBREQ#	REQ0#
1B	1	1	0	1	1	REQ2#/ REQ1#	REQ3#	CPUREQ#	REQ0#	PCEBREQ#
	x	1	1	x	x	Reserved				

Note that these two tables are permutations of the same table with different value of the Bank 3 fixed priority control bit. The fixed bank control bit(s) selects which requester is the highest priority device within that particular bank. Bits[7:4] must be programmed to all 0's (rotate mode disabled) to get these combinations. The selectable fixed priority schemes provide 24 of the 128 possible fixed mode permutations possible for the six masters.

5.4.1.2 Rotating Priority Mode

When any bank rotate control bit is set to a one, that particular bank rotates between the requesting inputs. Any or all banks can be set in rotate mode. If all four banks are set in rotate mode, the six supported masters are all rotated and the arbiter is in a pure rotating priority mode. If, within a rotating bank, the highest priority device (a) does not have an active request, the lower priority device (b or c) will be granted the bus. However, this does not change the rotation scheme. When the bank toggles, device b is the highest priority. Because of this, the maximum latency a device can encounter is two complete rotations.

5.4.1.3 Mixed Priority Mode

Any combination of fixed priority and rotate priority modes can be used in different arbitration banks to achieve a specific arbitration scheme.

5.4.1.4 Locking Masters

When a master acquires the PLOCK# signal, the arbiter gives that master highest priority until PLOCK# is negated and FRAME# is negated. This insures that a master that locked a resource will eventually be able to unlock that same resource.

5.4.2 ARBITRATION SIGNALING PROTOCOL

An agent requests the PCI Bus by asserting its REQ#. When the arbiter determines that an agent may use the PCI Bus, it asserts the agent's GNT#. Figure 22 shows an example of the basic arbitration cycle. Two agents (A and B) are used to illustrate how the arbiter alternates bus accesses. Note in Figure 22 that the current owner of the bus may keep its REQ# (REQ#-A) asserted when it requires additional transactions.

REQ#-A is asserted prior to or at clock 1 to request use of the PCI Bus. Agent A is granted access to the bus (GNT#-A is asserted) at clock 2. Agent A may start a transaction at clock 2 because FRAME# and IRDY# are negated and GNT#-A is asserted. Agent A's transaction starts when FRAME# is asserted (clock 3). Agent A requests another transaction by keeping REQ#-A asserted.

When FRAME# is asserted on clock 3, the arbiter determines that agent B has priority and asserts GNT#-B and negates GNT#-A on clock 4. When agent A completes its transaction on clock 4, it relinquishes the bus. All PCI agents can determine the end of the current transaction when both FRAME# and IRDY# are negated. Agent B becomes the PCI Bus owner on clock 5 (FRAME# and IRDY# are negated) and completes its transaction on clock 7. Note that REQ#-B is negated and FRAME# is asserted on clock 6, indicating that agent B requires only a single transaction. The arbiter grants the next transaction to agent A because its REQ# is still asserted.

5.4.2.1 REQ# And GNT# Rules

Figure 22 illustrates basic arbitration. Once asserted, GNT# may be negated according to the following rules:

1. If GNT# is negated at the same time that FRAME# is asserted, the bus transaction is valid and will continue.
2. One GNT# can be negated coincident with another being asserted, if the bus is not in the idle state. Otherwise, a one clock delay is incurred between the negation of the current master's GNT# and assertion of the next master's GNT#, to comply with the PCI specification.
3. While FRAME# is negated, GNT# may be negated, at any time, in order to service a higher priority master, or in response to the associated REQ# being negated.
4. If the MEMREQ# and MEMACK# are asserted, once the PCEB is granted the PCI Bus, the arbiter will not remove the internal grant until the PCEB removes it's request.

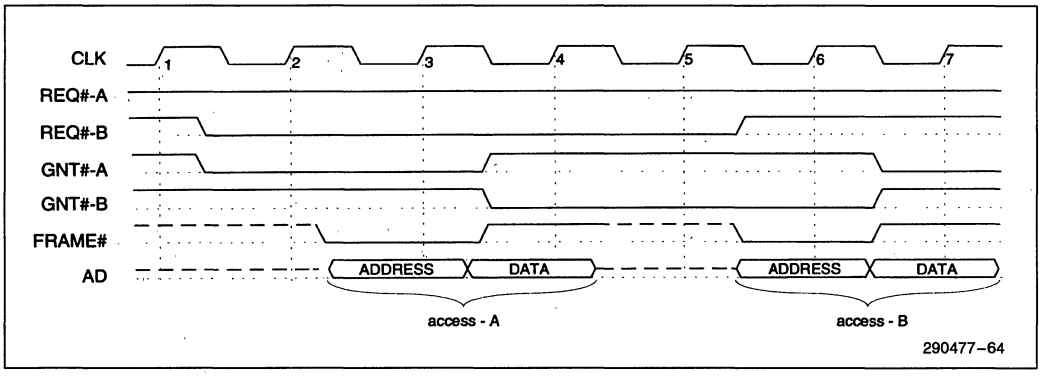


Figure 22. Basic Arbitration

5.4.2.2 Back-to-Back Transactions

Figure 23 illustrates arbitration for a back-to-back access. There are two types of back-to-back transactions by the same initiator; those that do not require a turn-around-cycle (see Section 5.1.3.1, Turn-Around-Cycle Definition) and those that do. A turn-around-cycle is not required when the initiator's second transaction is to the same target as the first transaction (to insure no TRDY# contention), and the first transaction is a write. This is a fast back-to-back. Under all other conditions, the initiator must insert a minimum of one turn-around-cycle.

During a fast back-to-back transaction, the initiator starts the next transaction immediately, without a turn-around-cycle. The last data phase completes when FRAME# is negated, and IRDY# and TRDY# are asserted. The current initiator starts another transaction on the same PCICLK that the last data is transferred for the previous transaction.

As a master, the PCEB does not know if it is accessing the same target, and, thus, does not generate fast back-to-back accesses. As a slave, the PCEB is capable of decoding fast back-to-back cycles.

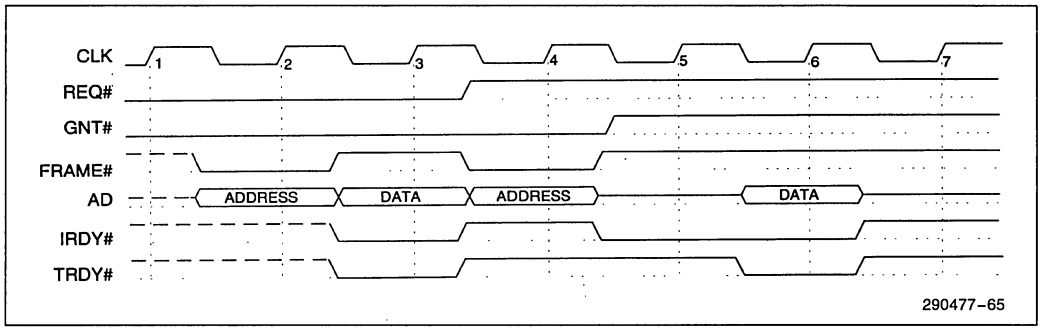


Figure 23. Arbitration for Back-to-Back Access

5.4.3 RETRY THRASHING RESOLVE

When a PCI initiator's access is retried, the initiator releases the PCI Bus for a minimum of two PCI clocks and then normally requests the PCI Bus again. To avoid thrashing of the bus with retry after retry, the PCI arbiter's state tracer provides REQ# masking. Tracking retried masters requires latching GNT# during FRAME# so that the correct retried master can be masked. The state tracer masks a REQ# after that particular agent is retried on the PCI Bus. The state tracer differentiates between two retry events. The two events include PCEB target retries and all other retries

For initiators that were retried by the PCEB as a target, the masked REQ# is flagged to be cleared upon RESUME# active. All other retries trigger the Master Retry Timer (described in Section 5.4.4.2, Master Retry Timer). When this timer expires, the mask is cleared.

5.4.3.1 Resume Function

The PCEB forces a retry to a PCI master (resulting in masking the REQ# of that master) for the following:

1. Buffer management activities (See Section 6.0, Data Buffering).
2. The EISA Bus is occupied by an EISA/ISA master or DMA.
4. The PCEB is locked as a resource and PLOCK# is asserted during the address phase.

5.4.3.2 Master Retry Timer

For any other retried PCI cycle, the arbiter masks the REQ# and flags it to be cleared by the expiration of a programmable timer. The first retry in this category triggers the programmable timer. Subsequent retries in this category are masked but do not reset the timer. Expiration of this programmable timer unmasks all REQ#s that are masked for this reason. The Retry Timer is programmable to 0 (disabled), 16, 32, or 64 PCICLKs.

If no other PCI masters are requesting the PCI Bus, all of the REQ#s masked for the timer are cleared and the timer is set to 0. Note that when there is a pending request that is internally masked, the PCEB does not park the CPU on the PCI Bus (i.e. PCI agent that uses CPUREQ#/CPUGNT# signal pair).

5.4.4 BUS LOCK MODE

As an option, the PCEB arbiter can be configured to run in Bus Lock Mode or Resource Lock Mode (default). The Bus Lock Mode is used to lock the entire PCI Bus. This may improve performance in some systems that frequently run quick read-modify-write cycles (i.e. access to the VGA frame buffer using the XCHG x86 instruction that automatically asserts the CPU LOCK# signal). Bus Lock Mode emulates the LOCK environment found in today's PC by restricting bus ownership when the PCI Bus is locked. While Bus Lock Mode improves performance in some systems, it may cause performance problems in other systems. With Bus Lock enabled, the arbiter recognizes a LOCK# being driven by an initiator and does not allow any other PCI initiator to be granted the PCI Bus until LOCK# and FRAME# are both negated, indicating the master released lock. When Bus Lock is disabled, the default resource lock mechanism is implemented (normal resource lock) and a higher priority PCI initiator could intervene between the cycles that are part of the locked sequence and run non-exclusive accesses to any unlocked resource.

CAUTION:

Bus Lock mode should not be used with non-GAT mode. If the system is initialized for both Bus Lock mode and non-GAT mode a deadlock situation might occur in the case where the first access to the locked device is a write instead of a read and the locked device has data in its internal posted write buffer. In GAT mode and/or Resource Lock mode this condition can not happen. If it is absolutely necessary to operate the system in the above mentioned combination of modes, then the posted write buffers of the device that might be involved in locked operations (typically semaphore in main memory) must be disabled.

5.4.5 MEMREQ#, FLSHREQ#, AND MEMACK# PROTOCOL

Before an EISA master or DMA can be granted the PCI Bus, it is necessary that all PCI system posted write buffers be flushed. Also, since the EISA-originated cycle could access memory on the Host/PCI Bridge, it is possible that the EISA master or DMA could be held in wait states (via EXRDY) waiting for the Host/PCI Bridge arbitration for longer than the 2.1 μs EISA/ISA specification. The PCEB has an optional mode called Guaranteed Access Time mode (GAT) that ensures that this timing specification is not violated. This is accomplished by delaying the EISA grant signal to the requesting master or DMA until the EISA Bus, PCI Bus, and the system memory bus are arbitrated for and owned.

The three sideband signals, MEMREQ#, FLSHREQ#, and MEMACK# are used to support the system Posted Write Buffer flushing and Guaranteed Access Time mechanism. The MEMACK# signal is the common acknowledge signal for both mechanisms. Note that, when MEMREQ# is asserted, FLSHREQ# is also asserted. Table 7 shows the relationship between MEMREQ# and FLSHREQ#.

Table 7. FLSHREQ# and MEMREQ#

FLSHREQ#	MEMREQ#	Meaning
1	1	Idle
0	1	Flush buffers pointing towards the PCI Bus to avoid EISA deadlock
1	0	Flush buffers pointing towards main memory for buffer coherency in APIC systems
0	0	GAT mode. Guarantees PCI Bus immediate access to main memory

5.4.5.1 Flushing System Posted Write Buffers

Once an EISA Bus owner (EISA/ISA master or the DMA) begins a cycle on the EISA Bus, the cycle can not be backed-off. It can only be held in wait states via EXRDY. In order to know the destination of EISA master cycles, the cycle needs to begin. After the cycle is started, no other device can intervene and gain ownership of the EISA Bus until the cycle is completed and arbitration is performed. A potential deadlock condition exists when an EISA-originated cycle to the PCI Bus forces a mandatory transaction to EISA, or when the PCI target is inaccessible due to an interacting event that also requires the EISA Bus. To avoid this potential deadlock, all PCI posted write buffers in the system must be disabled and flushed, before an EISA/ISA master or DMA can be granted the EISA Bus. The buffers must remain disabled while the EISA Bus is occupied. The following steps indicate the PCEB (and ESC) handshake for flushing the system posted write buffers.

1. When an EISA/ISA master, DMA or refresh logic requests the EISA Bus, the ESC component asserts EISAHOLD to the PCEB.
2. The PCEB completes the present cycle (and does not accept any new cycle) and gives the EISA Bus to the ESC by floating its EISA interface and asserting EISAHLDA. Before giving the bus to the ESC, the PCEB checks to see if it itself is locked as a PCI resource. It can not grant the EISA Bus as long as the PCEB is locked.

   At this point the PCEB's EISA-to-PCI Line Buffers and other system buffers (Host/PCI Bridge buffers) that are pointing to PCI are not yet flushed. The reason for this is that the ESC might request the bus in order to run a refresh cycle that does not require buffer flushing. That is not known until the EISA arbitration is frozen (after EISAHLDA is asserted).

   a. If the ESC needs to perform a refresh cycle, then it negates NMFLUSH# (an ESC-to-PCEB flush control signal). ESC drives the EISA Bus until it completes the refresh cycle and then gives the bus to the PCEB by negating EISAHOLD.

   b. If the ESC requested the EISA Bus on behalf of the EISA master, DMA or ISA master, then it asserts NMFLUSH# and tri-states the EISA Bus. The PCEB asserts the FLSHREQ# signal to the Host/PCI Bridge (and other bridges) to disable and flush posted write buffers.
3. When the Host/PCI Bridge completes its buffer disabling and flushing, it asserts MEMACK# to the PCEB. Other bridges in the system may also need to disable and flush their posted write buffers pointing towards PCI. This means that other devices may also generate MEMACK#. All of the MEMACK#s need to be "wire-OR'd". When the PCEB receives MEMACK# indicating that all posted write buffers have been flushed, it asserts NMFLUSH# to the ESC and the ESC gives the bus grant to the EISA device.
4. The PCEB continues to assert FLSHREQ# while the EISA/ISA master or DMA owns the EISA Bus. While FLSHREQ# is asserted the Host/PCI Bridge must keep its posted write buffers flushed.
5. MEMACK# should be driven inactive as soon as possible by the Host/PCI Bridge and other bridges after FLSHREQ# is negated. The PCEB waits until it detects MEMACK# negated before it can generate another FLSHREQ#.

5.4.5.2 Guaranteed Access Time Mode

When the PCEB's Guaranteed Access Time Mode is enabled (via the ARBCON Register), MEMREQ# and MEMACK# are used to guarantee that the ISA 2.1 μs CHRDY specification is not violated. Note that EISA's 2.5 μs maximum negation time of the EXRDY signal is a subset of the ISA requirement. Thus, 2.1 μs satisfies both bus requirements.

When an **EISA/ISA master or DMA slave** requests the EISA Bus (MREQ# or DREQ# active), the EISA Bus, the PCI Bus, and the memory bus must be arbitrated for and all three must be owned before the EISA/ISA master or DMA is granted the EISA Bus. The following lists the sequence of events:

1. An EISA/ISA master, DMA, or refresh logic requests the EISA Bus. The ESC asserts EISAHOLD signal to the PCEB.
2. The PCEB completes the present cycle (i.e. does not accept any new cycle) and gives the bus to the ESC by floating its EISA interface and asserting EISAHLDA. Before giving the bus to the ESC, the PCEB checks to see if it is locked as a PCI resource. It can not grant the EISA Bus as long as the PCEB is locked.

At this point, the PCEB's EISA-to-PCI Line Buffers and other system buffers (e.g., Host/PCI Bridge buffers) that are pointing to the PCI Bus are not flushed. The reason is that the ESC might request the bus to run a refresh cycle that does not require buffer flushing. This is not known until the EISA arbitration is frozen (after EISAHLDA is asserted).

3. Depending on whether the pending cycle is a refresh, the ESC initiates one of the following two actions:
   a. If the ESC needs to perform a refresh cycle, then it asserts NMFLUSH# (an ESC-to-PCEB flush control signal). The ESC drives the EISA Bus until it completes the refresh cycle and then gives the bus to the PCEB by negating EISAHOLD.
   b. If the ESC requested the EISA Bus on behalf of the EISA master, DMA or ISA master, then it asserts NMFLUSH# and tri-states the EISA Bus. If the PCEB is programmed in GAT (Guaranteed Access Time mode), the MEMREQ# and FLSHREQ# signals are asserted simultaneously to indicate request for direct access to main memory and a request to flush the system's posted write buffers pointing towards the PCI (including the PCEB's internal buffers). These requirements are necessary to insure that once the PCI and EISA Buses are dedicated to the PCEB, the cycle generated by the PCEB will not require the PCI or EISA Buses, thus creating a deadlock. MEMREQ# and FLSHREQ# are asserted as long as the EISA/ISA master or DMA owns the EISA Bus.
4. Once the Host/PCI Bridge has disabled and flushed its posted write buffers, and the memory bus is dedicated to the PCI interface, it asserts MEMACK#. Other bridges in the system may also need to disable and flush their posted write buffers pointing towards PCI due to the FLSHREQ# signal. This means that other devices may also generate a MEMACK#. All of the MEMACK#s need to be "wire-OR'd". When the PCEB receives MEMACK#, it assumes that all of the critical posted write buffers in the system have been flushed and that the PCEB has direct access to main memory, located behind the Host/PCI Bridge.
5. When MEMACK# is asserted by the PCEB, it will request the PCI Bus (internal PCEBREQ# signal). Before requesting the PCI Bus, the PCEB checks to see that the PCI Bus does not have an active lock. The PCI Bus is granted to the PCEB when it wins the bus through the normal arbitration mechanism. Once the PCEB is granted the PCI Bus (internalPCEBGNT#), the PCEB checks to see if PLOCK# is negated before it grants the EISA Bus. If the PCI Bus is locked when the PCEB is granted the PCI Bus, the PCEB releases the REQ# signal and waits until the PLOCK# is negated before asserting REQ# again. Once the PCEB owns the PCI Bus (internal PCEBGNT#), and the MEMACK# and MEMREQ# signals are asserted, the PCI arbiter will not grant the PCI Bus to any other PCI master except the PCEB until the PCEB releases its PCI REQ# line.
6. When the PCEB is granted the PCI Bus (internal PCEBGNT#) and LOCK# is inactive, it asserts NMFLUSH# to the ESC and the ESC gives the bus grant to the EISA device.

7. When the EISA Bus is no longer owned by an EISA master or DMA, the PCEB negates MEMREQ# and FLSHREQ# and the PCI request signal (internal PCEBREQ#. The negation of MEMREQ# and FLSHREQ# indicates that direct access to the resource behind the bridge is no longer needed and that the posted write buffers may be enabled. Note that MEMACK# should be driven inactive as soon as possible by the Host/PCI Bridge and other bridges after MEMREQ# is negated. The PCEB waits until it detects MEMACK# negated before it can generate another MEMREQ# or FLSHREQ#.

The use of MEMREQ#, FLSHREQ#, and MEMACK# does not guarantee GAT mode functionality with ISA masters that don't acknowledge CHRDY. These signals just guarantee the CHRDY inactive specification.

5.4.5.3 Interrupt Synchronization-Buffer Flushing

The ESC contains the system interrupt controller consisting of an 8259 compatible interrupt controller and an I/O APIC. For the 8259 compatible interrupt controller, the PCEB/ESC chip set is the default destination of the PCI interrupt acknowledge cycles. Interrupts in the system are commonly used as a synchronization mechanism. If interrupts are used by the EISA agents to notify the Host CPU that data is written to main memory, then posted data buffers must be flushed before the vector is returned during the interrupt acknowledge sequence. The PCEB handles this transparently to the rest of the system hardware/software. It retries the PCI interrupt acknowledge cycles and flushes the PCEB Line Buffers, if necessary.

The Advanced Programmable Interrupt Controller (APIC) uses a private message passing bus to send interrupt information to the companion APIC(s) residing at the host CPU(s). To support interrupts as a synchronization mechanism, system buffer coherency must be guaranteed before interrupts can be processed. With ESC's interrupt controller operating in APIC mode the PCEB and ESC use the PCEB/ESC interchip signal AFLUSH# to maintain system buffer coherency.

5.4.6 BUS PARKING

PCI Bus parking can be enabled/disabled via the ARBCON Register. Parking is only allowed for the device that is connected to CPUREQ# (i.e. the Host/PCI Bridge). REQ[3:0]#, and the internal PCEBREQ# are not allowed to park on the PCI Bus. When bus parking is enabled, CPUGNT# is asserted when no other agent is currently using or requesting the bus. This achieves the minimum PCI arbitration latency possible.

Arbitration Latency

Parked: 0 PCICLKs for parked agent, 2 PCICLKs for all other.

Not Parked: 1 PCICLK for all agents.

Upon assertion of CPUGNT# due to bus parking enabled and the PCI Bus idle, the CPU (i.e., parked agent) must ensure AD[31:0], C/BE[3:0]#, and (one PCICLK later) PAR are driven. If bus parking is disabled, then the PCEB drives these signals when the bus is idle.

5.4.7 PCI ARBITRATION AND PCEB/ESC EISA OWNERSHIP EXCHANGE

There are two aspects of PCEB/ESC EISA Bus ownership exchange that are explained in this section. They are related to GAT mode and RESUME/RETRY operations.

The PCEB is the default owner of the EISA Bus. When control of the EISA Bus is given to the ESC, all PCI operations targeted to the EISA subsystem (including the PCEB) are retried. Retry causes assertion of the PEREQ#/INTA# signal with PEREQ# semantics. In this way, the PCEB indicates to the ESC that it needs to obtain ownership of the EISA Bus.

5.4.7.1 GAT Mode And PEREQ# Signaling

In GAT mode, the PCEB owns the PCI Bus on behalf of the EISA master and other PCI agents (e.g., the Host/PCI Bridge) can not generate PCI cycles. Therefore, the PCEB never generates a back-off (i.e. retry), as long as the EISA Bus is controlled by the ESC. This might cause starvation of the PCI agents (including the Host/PCI Bridge i.e., CPU) even in the case of a moderately loaded EISA subsystem. The solution is that PEREQ#, in the GAT mode, is generated when any of the PCI Bus request signals are asserted. For particular Host/PCI Bridge designs (e.g. PCMC) this will be not be an adequate solution since their PCI request can be activated only based on the CPU generated cycle directed to PCI. This will not be possible since the Host Bus (CPU bus) in the GAT mode is controlled by the Host/PCI Bridge and not by the CPU. The solution to this type of design is to generate PEREQ# immediately after entering the GAT mode. This feature is controlled via ARBCON Register (bit 7).

5.4.7.2 PCI Retry And EISA Latency Timer (ELT) Mechanism

When a PCI cycle is retried by the PCEB (in non-GAT mode) because the EISA Bus is controlled by the ESC (EISAHLDA asserted), an internal flag is set for the corresponding PCI master. This flag masks the request of a particular master until the PCEB acquires the ownership of EISA and the RESUME condition clears the flag. If the PCI master, which is now unmasked, does not acquire the ownership of the PCI Bus within the time period before ESC asserts EISAHOLD again, the EISA Bus can be surrendered to the ESC. Unmasked masters will eventually gain the access to the PCI Bus but the EISA Bus will not be available and the master will be retried again. This scenario can be repeated multiple times with one or more PCI masters and starvation will occur.

To solve this situation, the PCEB arbitration logic incorporates an EISA Latency Timer mechanism. This mechanism is based on the programmable timer that is started each time that the ESC requires the bus (EISAHOLD asserted) and there is a PCI agent that has been previously retried because of activity on the EISA Bus. As soon as the ELT timer expires, any PCI cycle which is currently in progress is retried and the EISA Bus is given back to the ESC after the current PCI-to-EISA transaction completes. If all the PCI requesters, masked because of EISAHLDA, are serviced before the ELT timer expires, the EISA Bus is immediately surrendered to the ESC. The ELT provides a minimum time slice for PCI masters to access the EISA bus even if EISA masters, ISA masters or DMA devices are attempting to acquire the EISA bus.

Generally, the ELT is set to a larger value if latency sensitive PCI masters which typically access EISA are present in the system. Larger ELT values, however, do increase the worst case latency for EISA devices which typically access devices on PCI (e.g. main memory).

The EISA Latency Timer (ELT) is controlled by the ELTCR Register. The value written into ELTCR is system dependent. It is typically between 1 and 3 μs.

6.0 DATA BUFFERING

The PCEB contains data buffers (Figure 24) to isolate the PCI Bus from the EISA Bus and to provide concurrent EISA and PCI Bus operations and APIC operations. The Line Buffers are used for EISA-to-PCI memory reads and writes. A control bit in the EPMRA Registers permis the Line Buffers to be enabled (accesses are buffered) or disabled (accesses are non-buffered). Non-buffered accesses use the bypass path. Note that PCI and EISA I/O read/write cycles and PCI configuration cycles are always non-buffered and use the bypass path.

When data is temporarily stored in the buffers between the EISA Bus and PCI Bus, there are potential data coherency issues. The PCEB guarantees data coherency by intervening when data coherency could be lost and either flushing or invalidating the buffered data, as appropriate.

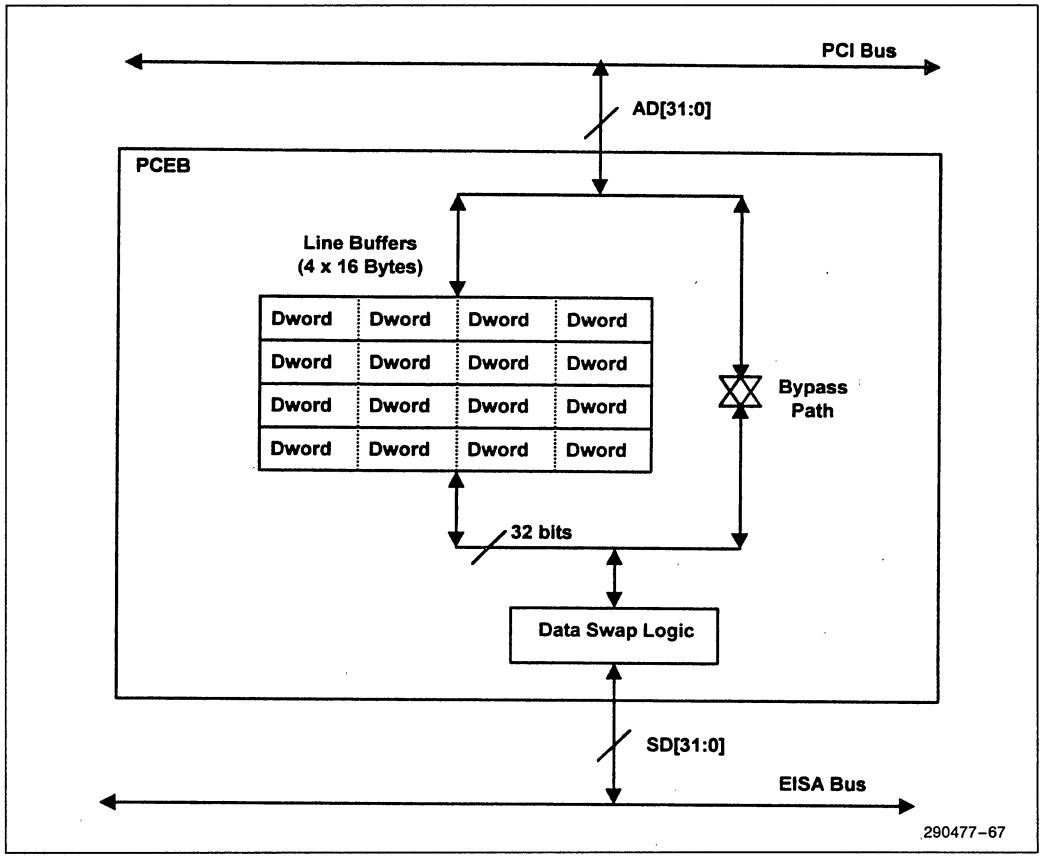


Figure 24. PCEB Data Buffers

6.1 Line Buffers

The PCEB contains four Line Buffers that are each four Dwords wide (16 bytes). The Line Buffers are bi-directional and are used by the EISA/ISA master and DMA to assemble/disassemble data. The data in each Line Buffer is aligned on 16 byte boundaries. When data is placed in one of the Line Buffers, the PCEB maintains the corresponding 16-byte boundary address until the data in the line is transferred to its destination or invalidated.

The Line Buffers can be enabled/disabled by writing to the PCICON Register. In addition, when the Line Buffers are enabled via the PCICON Register, buffering for accesses to the four programmable EISA-to-PCI memory regions (Region [4:1]) can be selectively disabled via the EPMRA Register.

During buffer operations, the four Line Buffers, collectively, are either in a write state or in a read state. These states are described in the following sections.

6.1.1 WRITE STATE

If a Line Buffer contains valid write data, it is in a *write state*. In the write state, data from the EISA/ISA master or DMA is posted in the Line Buffers. Posting means that the write operation on the EISA Bus completes when the data is latched in the buffer. The EISA master does not have to wait for the write to complete to its destination (memory on the PCI Bus). Posting permits the EISA Bus cycle to complete in a minimum time and permits concurrent EISA and PCI Bus operations. During posting, data accumulates in the Line Buffer until it is flushed (written to PCI memory) over the PCI Bus. A Line Buffer is scheduled for flushing by the PCEB when:

- the line becomes full.
- a subsequent write is a line miss (not within the current line boundary address range).
- the write is to an address of a lower Dword than the previous write. Note that writes to lower addresses within the same Dword do not cause a flush. Note also, that if two (or more) consecutive EISA Bus cycles are writes to the same Dword (i.e., the same byte or word locations within the Dword, or the same Dword for Dword writes) , the accessed buffer data is overwritten. However, if any of the flush conditions described in this list occur between the writes, the line is flushed before the next write and data is not overwritten.
- the last address location in the Line Buffer is accessed.
- a subsequent cycle is a read.
- the EISA Bus changes ownership.
- an interrupt acknowledge cycle is encountered.
- the ESC performs an EISA refresh cycle.
- the ESC's I/O APIC receives an interrupt request.

When a line is scheduled for flushing, the PCEB begins arbitration for the PCI Bus. If more than one line is scheduled to be flushed, the Line Buffers are flushed in a "first scheduled, first to be flushed" order. If the line to be flushed contains valid data in only one Dword, the PCEB uses a single data transfer cycle on the PCI Bus. Otherwise, flushing operations use burst transfers.

During flushing, write data within a Line Buffer is packetized into Dword quantities, when possible, for a burst transfer over the 32-bit PCI Bus. Packetizing occurs at two levels - Dwords within a line and bytes/words within a Dword. When a Line Buffer is flushed, all of the valid Dwords within the line are packetized into a single PCI burst write cycle. In addition, all valid data bytes within a Dword boundary are packetized into a single data phase of the burst cycle. Packetizing reduces the PCI arbitration latency and increases the effective PCI Bus bandwidth. When multiple Line Buffers are schedule for flushing, each Line Buffer is packetized separately. Packetizing across Line Buffer boundaries is not permitted.

During flushing, strong ordering is preserved at the Dword level (i.e., the Dwords are flushed to PCI memory in the same order that they were written into the Line Buffer). Note, however, that strong ordering is not preserved at the byte or word levels (i.e., even if byte or word transfers were used by the EISA/ISA master or DMA to sequentially write to a Dword within a Line Buffer, all of the bytes in the resulting Dword boundary are simultaneously flushed to PCI memory).

Because strong ordering is not preserved within a Dword boundary, care should be used when accessing memory-mapped I/O devices. If the order of byte or word writes to a memory-mapped I/O device needs to be preserved, buffered accesses should not be used. By locating memory-mapped I/O devices in the four programmable EISA-to-PCI memory regions, buffering to these devices can be selectively disabled.

6.1.2 READ STATE

If a Line Buffer contains valid read data, it is in a *read state*. Read data is placed in the Line Buffer by two PCEB mechanisms - fetching and prefetching. Data is placed in the Line Buffer on demand (fetching) when the data is requested by a read operation from the EISA/ISA master or DMA. The PCEB also prefetches data that has not been explicitly requested but is anticipated to be requested. Once in the Line Buffer, data is either read by the EISA/ISA master or DMA (and then invalidated) or invalidated without being read. Read data is invalidated when:

- data in the Line Buffer is read (transferred to the EISA/ISA master or DMA). This prevents reading of the same data more than once.
- a subsequent read is a line miss (not to the previously accessed Line Buffer). Valid data in the current Line Buffer is invalidated. If a new line had been prefetched during access to the current line, data in the prefetched line is not invalidated, unless the access also misses this line. In this case, the data in the prefetched line is invalidated.
- a subsequent cycle is a write. Data in all Line Buffers are invalidated.

If the requested data is in the Line Buffer, a line hit occurs and the PCEB transfers the data to the EISA/ISA master or DMA (and invalidates the hit data in the buffer). If EISA Bus reads hit two consecutive line addresses, the PCEB prefetches the next sequential line of data from PCI memory (using a PCI Bus burst transfer). This prefetch occurs concurrently with EISA Bus reads of data in the already fetched Line Buffer. If consecutive addresses are not accessed, the PCEB does not prefetch the next line.

A line miss occurs if the requested data is not in the Line Buffer. If a line miss occurs, the PCEB invalidates data in the missed Line Buffer. If the requested data is in a prefetched line, the read is serviced. If a line was not prefetched or the read missed the prefetched line, the PCEB invalidates any prefetched data and fetches the Dword containing the requested data. During this fetch, the PCEB holds off the EISA/ISA master or DMA with wait states (by negating EXRDY). When the requested data is in the Line Buffer, it is transferred to the EISA Bus. Simultaneously with the EISA Bus transfer, the PCEB prefetches the rest of the line data (Dwords whose addresses are within the line and above the Dword address of the requested data). The Dword containing the requested data and the rest of the Dwords in the line (located at higher addresses) are fetched from PCI memory using a burst transfer, unless the requested data is in the last Dword of a line. In this case, a single cycle read occurs on the PCI Bus.

For purposes of data read operations, all four 4-Dword buffers are used to form two 8-Dword lines (32 bytes each). There are only tow address pointers, one for each line. Fetching fractions of a line is accomplished as described above (i.e., starting from the first requested Dword).

The MSBURST# input signal is used to supplement control of the prefetch sequence. The MSBURST# signal is activated only when an EISA master desires to do burst transfers to access sequential data (although this is not an absolute EISA rule, i.e., theoretically the data can be non-sequential after an EISA slave indicates its ability via SLBURST#). This will occur during the first data transfer.

The Line Buffer control logic dynamically switches between two prefetch modes—Half Line Prefetch (16 bytes fetch) and Full Line Prefetch (32 bytes fetch)

The prefetch control logic has implemented a Sequential Access Flag which is cleared before the initial prefetch. Initial prefetch (first data fetch) starts in the Half Line Prefetch mode and is extended to Full Line Prefetch mode immediately after MSBURST# is sampled asserted at which time the Sequential Access Flag is automatically set (this is done on-the-fly during the first line fetch). If after the initial prefetch the Sequential Access Flag has not been set (MSBURST# remained not asserted) and the control logic recognizes two consecutive hits (in incrementally sequential Dwords including the first one which is originally requested), the Sequential Access Flag is set and the prefetch control logic switches to Full Line Prefetch mode. An additional 32-byte line (or fraction depending on alignment) will be fetched.

When the Sequential Access Flag is set, prefetching is accomplished using the Full Line Prefetch mode. Each time a line buffer (32 bytes) is available, an additional line will be fetched as long as the Sequential Access Flag remains set.

When out-of-order access is recognized within the prefetched data or a miss occurs when there is valid fetched data, the Sequential Access Flag is cleared and the prefetch mode changes to Half Line Prefetch. Also, the Sequential Access Flag is cleared when MSBURST# transitions from active to inactive.

When the Sequential Access Flag is not asserted, the prefetch control logic operates in Half Line Prefetch mode during which only 16 bytes of data is fetched at a time. The same test for sequential access is repeated, and if access is recognized, the Sequential Access Flag is set and the control switches to Full Line Prefetch mode.

6.2 Buffer Management Summary

Table 8 shows Line Buffer for different cycles. Note that the first three columns together define the cycles that may trigger buffer activity.

Table 8. Buffer Management Summary

Master (Origin)	Cycle Type	Slave (Destination)	Line Buffer Data in Write State	Line Buffer Data in Read State
PCI	Memory Read	EISA	Flush	No Action
PCI	Memory Write	EISA	No Action	Invalidate
PCI	I/O Read	EISA	Flush	No Action
PCI	I/O Write	EISA	No Action	Invalidate
PCI	Interrupt Acknowledge	PCEB/ESC	Flush	No Action
PCI	Configuration Cycle	PCEB Registers	No Action	No Action
PCI	Memory Read/Write	PCI	No Action	No Action
PCI	I/O Read/Write	PCI	No Action	No Action
EISA	Bus Ownership Change	—	Flush	No Action
EISA	Memory Read/Write	EISA	No Action	No Action

Table 8. Buffer Management Summary (Continued)

Master (Origin)	Cycle Type	Slave (Destination)	Line Buffer Data in Write State	Line Buffer Data in Read State
EISA	Memory Read/Write	PCI	(Note 1)	(Note 1)
EISA	I/O Read/Write	EISA	No Action	No Action
EISA	I/O Read/Write	PCI	Flush	Invalidate
ESC's I/O APIC	APIC Bus Message Transfer	Local APIC	Flush	No Action

NOTES:
1. Change from write to read operation or from read to write causes the Line Buffers to be flush or invalidate, respectively.
2. LOCKed cycles (both from PCI and EISA) are not buffered within the PCEB. They are processed using the bypass path.

7.0 EISA INTERFACE

The PCEB provides a fully EISA Bus compatible master and slave interface. This interface provides address and data signal drive capability for eight EISA slots and supports the following types of cycles:

- PCI-initiated memory and I/O read/write accesses to an EISA/ISA device.
- EISA/ISA/DMA-initiated memory and I/O read/write accesses to a PCI device (i.e. via the Line Buffers, if necessary).
- Accesses contained within the EISA Bus (only data swap buffers involved).

For transfers between the EISA Bus and PCI Bus, the PCEB translates the bus protocols. For PCI master-initiated cycles to the EISA Bus, the PCEB is a slave on the PCI Bus and a master on the EISA Bus. For EISA master-initiated cycles to the PCI Bus, the PCEB is a slave on the EISA Bus and a master on the PCI Bus.

NOTE:

1. The PCEB is not involved in refresh cycles on the EISA Bus. When the REFRESH# signal is asserted, the PCEB disables EISA Bus address decoding.
2. Wait state generation on the EISA Bus is performed by the ESC. ISA memory slaves (8 or 16 bits) and ISA I/O slaves can shorten their default or standard cycles by asserting the NOWS# signal line. It is the responsibility of the ESC to shorten these cycles when NOWS# is asserted. Note that ISA I/O 16-bit devices can shorten their cycles by asserting NOWS#. If CHRDY and NOWS# are driven low during the same cycle, NOWS# will not be used and wait states are added as a function of CHRDY. For more details on the wait state generation and the NOWS# signal, refer to the ESC data sheet.
3. All locked PCI cycles (PLOCK# asserted) destined to the EISA Bus are converted to EISA locked cycles using the LOCK# signal protocol. The PCEB is a locked resource during these cycles and maintains control of the EISA Bus until the locked PCI sequence is complete.
4. All locked EISA cycles (LOCK# asserted) destined to PCI are converted to PCI locked cycles using the PLOCK# signal protocol. The PLOCK# signal remains active as long as the EISA LOCK# signal is asserted.

5. The PCEB contains EISA data swap buffers for data size translations between mismatched PCI Bus and EISA Bus transfers and between mismatched devices contained on the EISA Bus. Thus, if data size translation is needed, the PCEB is involved in cycles contained to the EISA Bus, even if the PCEB is neither the master or slave. For data size translation operations, see Section 8.0, EISA Data Swap Buffers.
6. For ISA master cycles to PCI memory or I/O, the ESC translates the ISA signals to EISA signals. The PCEB, as an EISA slave, forwards the cycle to the PCI Bus.
7. For ISA master cycles to ISA/EISA slaves, the PCEB is not involved, except when the cycle requires data size translations. See the ESC data sheet for cycles that are contained within the EISA Bus (i.e., EISA-to-EISA, EISA-to-ISA, ISA-to-ISA, and ISA-to-EISA device cycles).
8. In this section, LA[31:24]# and LA[23:2] are collectively referred to as LA[31:2].

7.1 PCEB As An EISA Master

The PCEB is an EISA master for PCI-initiated cycles targeted to the EISA Bus. When the PCEB decodes the PCI cycle as a cycle destined to the EISA Bus (via subtractive or negative (82374SB only) decoding, as described in Section 4.0, Address Decoding), the PCEB becomes a slave on the PCI Bus. If the PCEB owns the EISA Bus, the cycle is forwarded to the EISA/ISA device. If the PCEB does not own the EISA Bus (EISAHOLDA is asserted to the ESC), the PCI master is retried and the PCEB issues an EISA Bus request to the ESC. For PCI-to-EISA I/O and memory read/write accesses, the PCEB runs standard EISA Bus cycles.

When cycles are forwarded to a matched EISA/ISA slave, the PCEB is the EISA master and controls the transfer until the cycle is terminated. For mismatched cycles to an EISA/ISA slave, the PCEB backs off the EISA Bus as described in Section 7.1.3, Back-Off Cycle.

7.1.1 STANDARD EISA MEMORY AND I/O READ/WRITE CYCLES

The standard EISA cycle completes one transfer each two BCLK periods (zero wait states). The standard EISA memory or I/O cycle begins when the PCEB presents a valid address on LA[31:2] and drives M/IO# high for a memory cycle and low for an I/O cycle. The address can become valid at the end of the previous cycle to allow address pipelining. The EISA slave decodes the address and asserts the appropriate signals to indicate the type of slave and whether it can perform any special timings. The slave asserts EX32# or EX16# to indicate support of EISA cycles.

For extended cycles, the EISA slave introduces wait states using the EXRDY signal. Wait states allow a slower slave to get ready to complete the transfer. The slave negates EXRDY after it decodes a valid address and samples START# asserted. The slave may hold EXRDY negated for a maximum of 2.5 μs to complete a transfer, and must release EXRDY synchronous to the falling edge of BCLK to allow a cycle to complete. Note that the PCEB, as an EISA master, never introduces wait states.

Figure 25 shows three data transfer cycles between an EISA master and an EISA slave. The first transfer is an extended transfer (EXRDY negated), followed by two standard cycles. For PCI cycles that are forwarded to the EISA Bus, the PCEB is the EISA master. The PCEB asserts START# to indicate the start of a cycle. The PCEB also drives W/R# to indicate a read or write cycle and BE[3:0]# to indicate the active bytes. The LA[31:2] and the BE[3:0] remain valid until after the negation of START#. A slave that needs to latch the address does so on the trailing edge of START#.

The ESC asserts CMD# simultaneously with the negation of START# to control data transfer to or from the slave. If a read cycle is being performed, the slave presents the requested data when CMD# is asserted and holds it valid until CMD# is negated by the ESC. For a write cycle, the PCEB presents the data prior to the assertion of CMD# and the slave latches it on or before the trailing edge of CMD#.

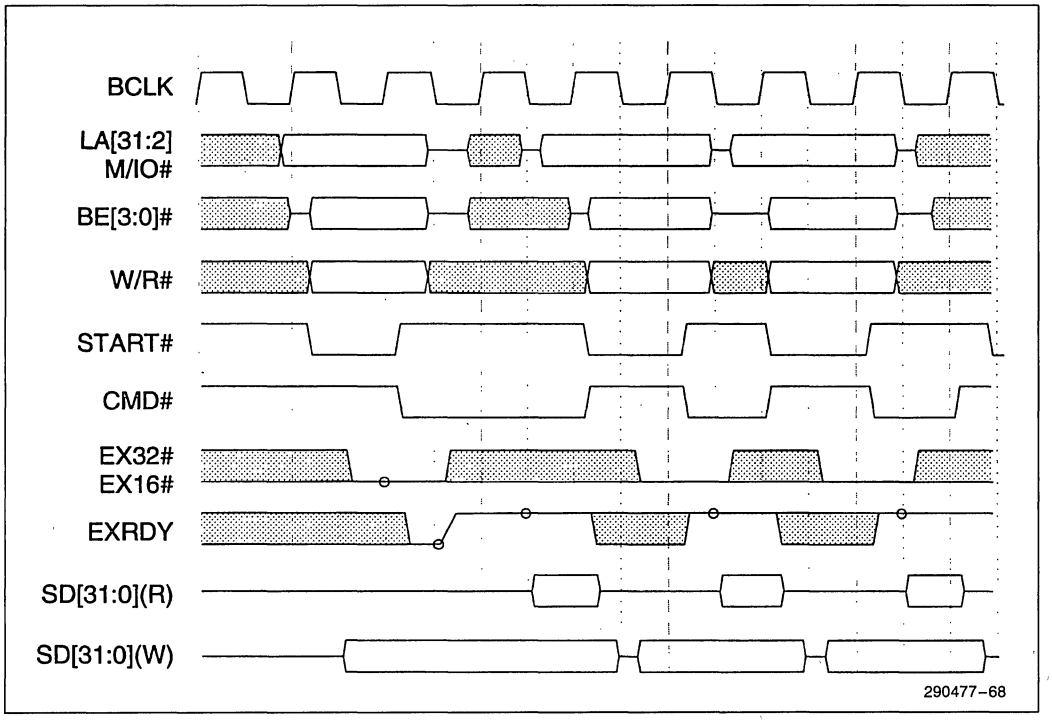


Figure 25. EISA Memory and I/O Read/Write Cycle (One Extended and Two Standard Cycles)

7.1.2 EISA BACK-OFF CYCLE

For mismatched cycles to an EISA/ISA slave, the PCEB, as a master, backs off the EISA Bus by floating the START#, BE[3:0]# and SD[31:0] signals one and half BCLKs after START# has been asserted. The ESC controls the EISA Bus for the duration of the cycle. This allows the ESC to perform data translation, if necessary. At the end of the cycle, the ESC transfers control back to the PCEB by asserting EX16# and EX32# on the falling edge of BCLK, before the rising edge of BCLK that the last CMD# is negated. Refer to the ESC data sheet for further details on master back-off and the cycle transfer control operations.

Figure 26 shows an example of a back-off sequence during a 32-bit EISA master to 16-bit EISA slave Dword read and write operation. The thick lines indicate the change of control between the master and the ESC.

PCEB Reading From a 16-bit EISA Slave

As a 32-bit EISA master, the PCEB begins by placing the address on LA[31:2] and driving M/IO#. The 16-bit EISA slave decodes the address and asserts EX16#. The PCEB asserts START#, W/R#, and BE[3:0]#. The ESC samples EX32# and EX16# on the rising edge of BCLK following the assertion of START# and asserts CMD#. At the same time, the PCEB negates START# and samples EX32#. When EX32# is sampled negated, the PCEB floats START# and BE[3:0]#. Note that, the PCEB continues to drive a valid address on LA[31:0].

The ESC negates CMD# after one BCLK period unless the slave adds wait states (negates EXRDY). The ESC latches SD[15:0] into the PCEB's data swap buffer on the trailing edge of CMD#. The ESC controls the PCEB data swap buffers via the PCEB/ESC Interface. The ESC then asserts START# and presents BE[3:0] (upper word enabled). The ESC negates START# and asserts CMD#. The slave latches the address on the trailing edge of START# and presents data on SD[15:0]. The ESC negates CMD# after one BCLK, unless the slave negates EXRDY. The ESC latches SD[15:0] into the PCEB data swap buffers on the trailing edge of CMD# and instructs the PCEB data swap buffer to copy D[15:0] to D[31:0] and asserts EX32#. Note that, since the transfer is intended for the PCEB, the data is not re-driven back out onto the EISA Bus. The ESC floats the START# and BE[3:0]#. The PCEB regains control of the EISA Bus after sampling EX32# and EX16# asserted.

PCEB Writing To a 16-bit EISA Slave

As a 32-bit EISA master, the PCEB begins by placing the address on LA[31:2] and driving M/IO#. The 16-bit EISA slave decodes the address and asserts EX16#. The PCEB asserts START#, W/R#, BE[3:0]#, and SD[31:0]. The ESC samples EX32# and EX16# on the rising edge of BCLK following the assertion of START# and asserts CMD#. At the same time, the PCEB negates START# and samples EX32#. When EX32# is sampled negated, the PCEB floats START#, SD[31:0], and BE[3:0]#. The data is latched in the PCEB's data swap buffers. Note that the PCEB continues to drive a valid address on LA[31:2].

The ESC instructs the PCEB to drive the data out on SD[31:0] and asserts CMD# after sampling EX32# negated. The slave may sample SD[15:0] while CMD# is asserted. The ESC negates CMD# after one BCLK, unless the slave adds wait states (negates EXRDY). The ESC then presents BE[3:0] (upper word enabled) and asserts START#. The ESC instructs the PCEB to copy SD[31:0] to SD[15:0], negates START# and asserts CMD#. The ESC negates CMD# after one BCLK, unless the slave negates EXRDY. The slave latches the address on the trailing edge of START# and samples SD[15:0] on the trailing edge of CMD#. The ESC returns control of the EISA Bus to the PCEB by floating BE[3:0]# and START#, then asserting EX32#. The PCEB samples EX32# and EX16# asserted on the rising edge of BCLK.

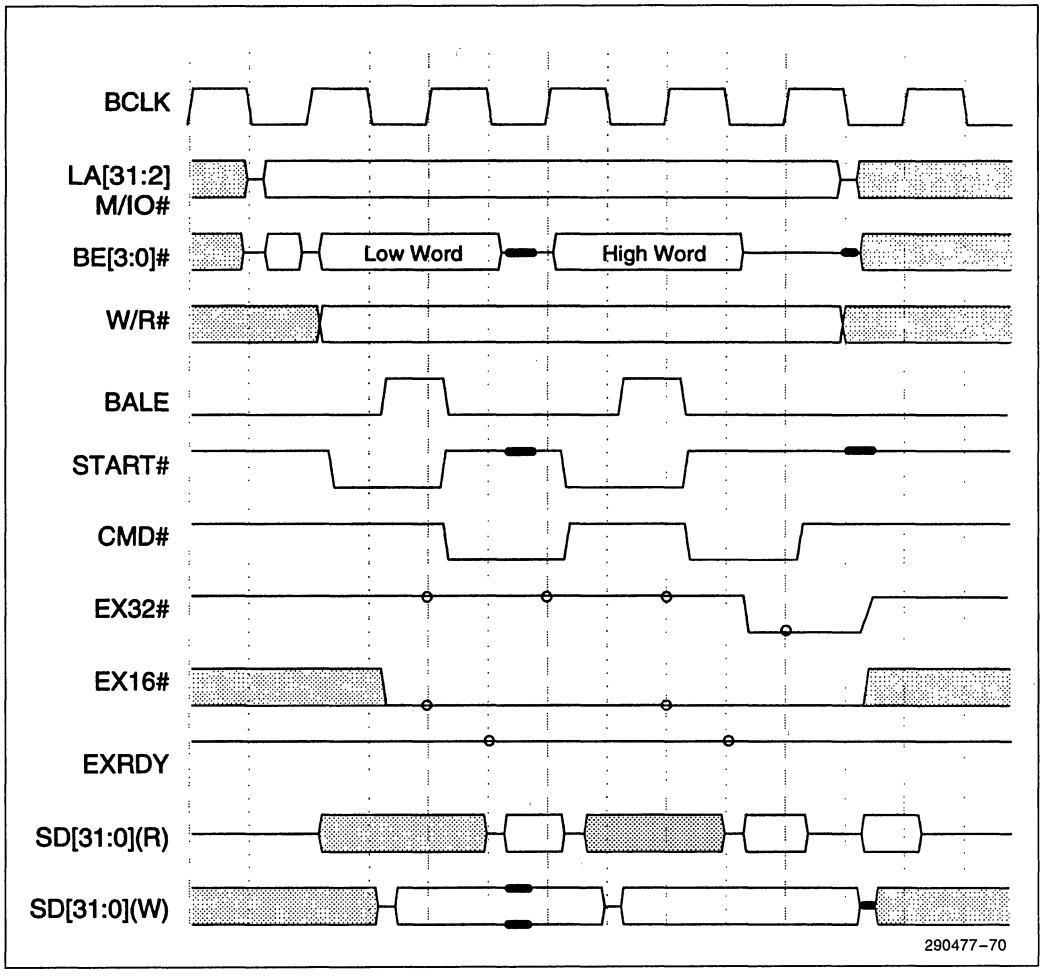


Figure 26. EISA Back-Off Cycle

7.2 PCEB As An EISA Slave

The PCEB is an EISA slave for EISA/ISA/DMA-initiated cycles targeted to the PCI Bus. If the PCEB positively decodes the address (access to one of the EISA programmed main memory segments or access to one of the programmable EISA-to-PCI memory or I/O regions), the PCEB becomes an EISA slave and the cycle is forwarded to the PCI Bus. If the PCEB does not positively decode the address, the cycle is contained to the EISA Bus. For cycles contained to the EISA Bus (i.e., EISA-to-EISA, EISA-to-ISA, ISA-to-ISA, and ISA-to-EISA device cycles), the PCEB is only involved when data size translation is needed.

The PCEB responds as a 32-bit EISA slave. If the EISA master size is not 32 bits, the cycle is a mismatch and invokes data size translation. For details on data size translation, refer to Section 8.0, EISA Data Swap Buffers.

All EISA master memory read cycles to PCI memory start as extended cycles, unless the cycle triggers a read hit to one of the four Line Buffers. If the data is available in the Line Buffers, the PCEB supplies the data to the EISA master without adding wait states. Otherwise, the cycle is extended (wait states added via EXRDY) until the data is available. Note that for non-buffered accesses, the EISA cycle is always extended until data is available from the PCI Bus.

If the Line Buffers are enabled, write cycles to PCI memory are posted in the Line Buffers. If the write can be immediately posted, wait states are not generated on the EISA Bus. Otherwise, the cycle is extended (via wait states) until the data can be posted. Note that writes can be posted to available Line Buffers concurrently with other Line Buffers being flushed to the PCI Bus.

All EISA master I/O read/write accesses to PCI I/O space are non-buffered and always start as extended cycles. Data transfer on the EISA Bus occurs when the requested data is available from the PCI Bus.

For mismatched cycles to the PCEB, the EISA/ISA master backs off the EISA Bus as described in Section 7.1.3, Back-Off Cycle.

7.2.1 EISA MEMORY AND I/O READ/WRITE CYCLES

The standard EISA cycle completes one transfer each two BCLK periods (zero wait states). The standard EISA memory or I/O cycle begins with the EISA master presenting a valid address on LA[31:2] and driving M/IO# high for a memory cycle and low for an I/O cycle. The address can become valid at the end of the previous cycle to allow address pipelining. When the PCEB positively decodes the address, it asserts EX32# to indicate 32-bit support. For memory cycles, the PCEB also asserts SLBURST# to indicate support for burst transfers.

For extended cycles, the PCEB introduces wait states using the EXRDY signal. The PCEB may hold EXRDY negated for a maximum of 2.5 μs to complete a transfer, and releases EXRDY synchronous to the falling edge of BCLK to allow a cycle to complete.

Figure 27 shows three data transfers between an EISA master and an EISA slave. The first transfer is an extended transfer (EXRDY negated), followed by two standard cycles. For EISA cycles that are forwarded to the PCI Bus, the PCEB is an EISA slave. The EISA master asserts START# to indicate the start of a cycle. The EISA master also drives W/R# to indicate a read or write cycle and BE[3:0]# to indicate the active bytes. The LA[31:2] and the BE[3:0] remain valid until after the negation of START#. The PCEB latches the address on the trailing edge of START#.

The ESC asserts CMD# simultaneously with the negation of START# to control data transfer to or from the PCEB. If a read cycle is being performed, the PCEB presents the requested data when CMD# is asserted and holds it valid until CMD# is negated by the ESC. For a write cycle, the EISA master must present the data prior to the assertion of CMD# and the PCEB latches it on the trailing edge of CMD#.

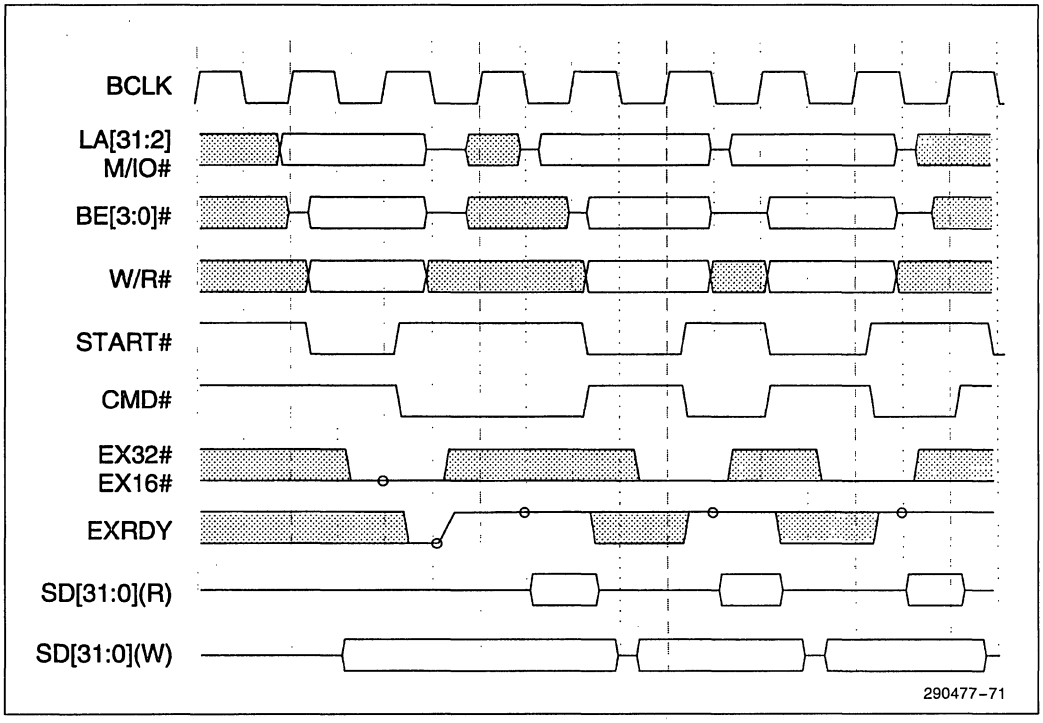


Figure 27. EISA Memory and I/O Read/Write Cycles (One Extended and Two Standard Cycles)

7.2.2 EISA MEMORY BURST CYCLES

The EISA burst cycles permit a continuous sequence of read or write cycles in zero wait-states (1 BCLK per transfer). A burst transfer is either all reads or all writes. Mixed cycles are not allowed. As an EISA slave, the PCEB supports burst memory reads and burst memory writes from/to its Line Buffers. Figure 28 shows an example of a burst sequence for both memory reads and writes on the EISA Bus. During the particular burst sequence, five data transfers occur with a wait state added on the third data transfer.

The first transfer in a burst transfer begins like the standard cycle described above. The EISA master presents a valid address on LA[31:2]. The PCEB, after decoding the address and M/IO#, responds by asserting SLBURST#. The EISA master must sample SLBURST# on the rising edge of BCLK at the trailing edge of START#. The EISA master asserts MSBURST# on the falling edge of BCLK and presents a second address to the PCEB. The ESC holds CMD# asserted while the burst is being performed. If MSBURST# is not asserted by the master, the cycle is run as a standard cycle.

If the cycle is a burst read, the EISA master presents burst addresses on the falling edge of every BCLK. The PCEB presents the data for that address, which is sampled one and half BCLKs later. If the cycle is a burst write, the EISA master presents the data on the rising edge of BCLK, a half cycle after presenting the address. The PCEB samples memory write data on the rising BCLK edge when CMD# is asserted (regardless of the state of MSBURST#). The EISA master terminates the burst cycles by negating MSBURST# and completing the last transfer.

To add wait states during a burst sequence, the PCEB negates EXRDY before the falling edge of BCLK (with CMD# asserted). The EISA master samples EXRDY on the falling edge of BCLK and extends the cycle until EXRDY is asserted. The EISA master can still change the next address even though EXRDY is negated.

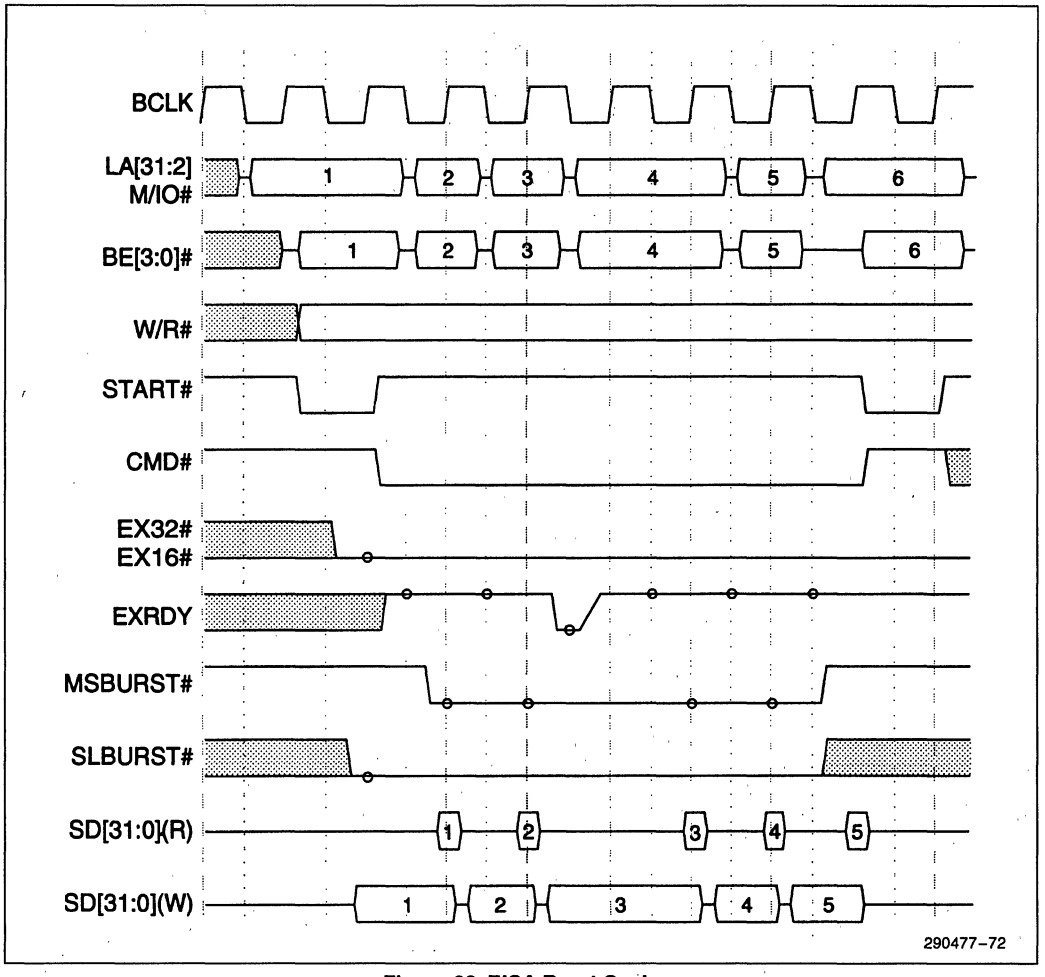


Figure 28. EISA Burst Cycle

7.3 I/O Recovery

The I/O recovery mechanism in the PCEB guarantees a minimum amount of time between back-to-back 8-bit and 16-bit PCI cycles to ISA I/O slaves. Delay times (in BCLKs) for 8-bit and 16-bit cycles are individually programmed via the IORT Register. Accesses to an 8-bit device followed by an access to a 16-bit device use the 8-bit recovery time. Similarly, accesses to a 16-bit device followed by an access to an 8-bit device use the 16-bit recovery time. The PCEB cycles to EISA I/O, DMA cycles, and EISA/ISA bus masters to I/O slaves do not require any delay between back-to-back I/O accesses.

Note that I/O recovery is only required for ISA I/O devices. However, since the PCEB does not distinguish between 8-bit ISA and 8-bit EISA, the delay is also applied to 8-bit EISA I/O accesses (i.e. the ESC).

8.0 EISA DATA SWAP BUFFERS

The PCEB contains a set of buffers/latches that perform data swapping and data size translations on the EISA Bus when the master and slave data bus sizes do not match (e.g., 32-bit EISA master accessing a 16-bit EISA slave). During a data size translation, the PCEB performs one or more of the following operations, depending on the master/slave type (PCI/EISA/ISA), transfer direction (read/write), and the number of byte enables active (BE[3:0]#):

- Data assembly or disassembly
- Data copying (up or down)
- Data re-drive

These operations are described in this section. An example is provided in Section 8.3, The Re-Drive Operation, that shows a cycle where all three functions are used.

The PCEB performs data size translations on the EISA Bus using the data swap buffer control signals generated by the ESC. These signals are described in Section 10.0, PCEB/ESC Interface.

8.1 Data Assembly And Disassembly

The data assembly/disassembly process occurs during PCI, EISA/ISA, and DMA cycles when the master data size is greater than the slave data size. For example, if a 32-bit PCI master is performing a 32-bit read cycle to an 8-bit ISA slave, the ESC intervenes and performs four 8-bit reads. The data is assembled in the PCEB (Figure 29). Once assembled, the PCEB transfers the data as a single Dword to the 32-bit PCI master during the fourth cycle. For a 32-bit write cycle, the PCEB disassembles the Dword by performing four write cycles to the slave. The actual number of cycles required to perform an assembly/disassembly process and make a transfer is a function of the number of bytes (BE[3:0]#) requested and the master/slave size combination.

During EISA master assembly/disassembly transfers, cycle control is transferred from the master to the ESC. The master relinquishes control by backing off the bus (i.e., by floating its START#, BE[3:0], and SD[31:0] signals on the first falling edge of BCLK after START# is negated). The ESC controls the assembly/disassembly process in the PCEB via the data swap buffer control signals on the PCEB/ESC interface. At the end of the assembly/disassembly process, cycle control is transferred back to the bus master (by the ESC asserting EX16# and EX32#). An additional BCLK is added at the end of the transfer to allow the exchanging of cycle control to occur. During DMA transfers, cycle control is maintained by the ESC for the entire cycle.

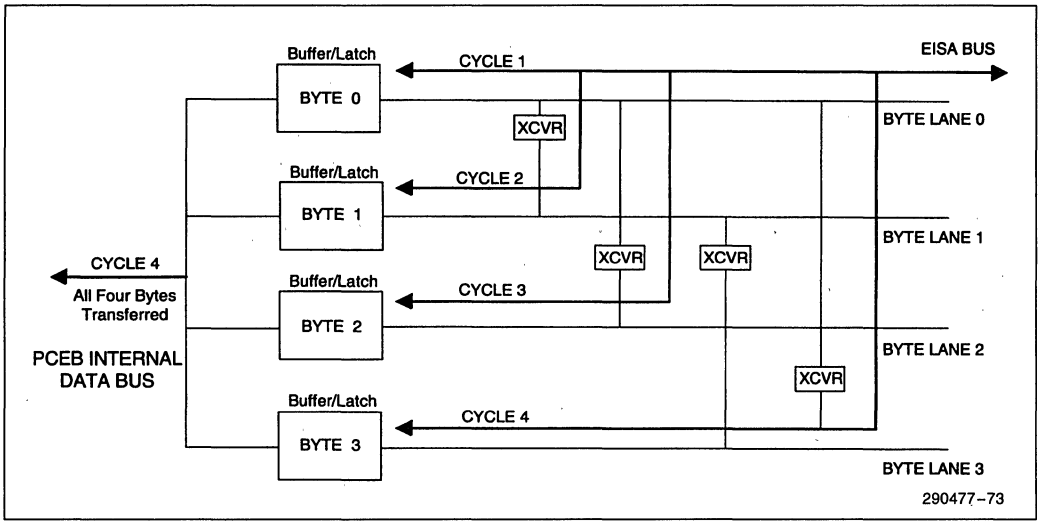


Figure 29. Assembly Function: PCI 32-bit Read from an 8-bit EISA or ISA SlaveBE[3:0]# = 0000

8.2 The Copy Operation (Up Or Down)

The copy operation (Figure 30) is invoked during data transfers between byte lanes. This operation allows the assembly/ disassembly of the data pieces during the cycles between mismatched master/slave combinations. For example, Section 8.1, Data Assembly and Disassembly, describes a 32-bit master read from an 8-bit slave where the data is copied up during the assembly process. Copy-up is used for data assembly and copy-down is used for data disassembly.

The copy-up and copy-down operations are also used during transfers where assembly or disassembly are not required. These transfers are:

- When the master size is smaller than the slave size (e.g. 16-bit EISA master cycle to a 32-bit EISA slave).
- Between a mis-matched master/slave combination when only a byte or a word needs to be transferred (e.g. 32-bit EISA master cycle to an 8-bit ISA slave and only a byte needs to be transferred).

The number of bytes copied up or down is a function of the number of bytes requested (BE[3:0]#) and the master/slave size combinations. During EISA master cycles where the data copying is performed, cycle control is transferred from the bus master to the ESC, except during transfers where the master's data size is smaller than the slave's data size. During DMA transfers, bus control is maintained by the ESC throughout the transfer.

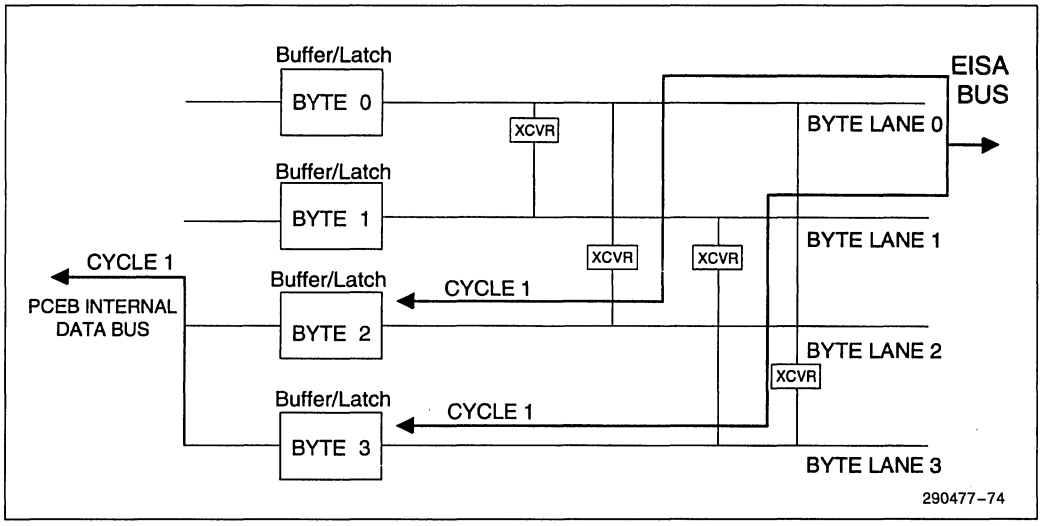


Figure 30. Copy Function: PCI 16-bit Read from a 16-bit EISA or ISA Slave—BE[3:0]# = 0011

8.3 The Re-Drive Operation

The re-drive operation (Figure 31) is used when both the master and the slave, other than PCEB, are on the EISA Bus and the master/slave size combination is mis-matched. Specifically, re-drive occurs:

- during EISA master and DMA cycles (excluding DMA compatible cycles) where the master's data size is greater than the slave's data size.
- during EISA master cycles to ISA slaves where the master/slave match in the size.
- during DMA burst write cycles to a non-burst memory slave.

During a re-drive cycle, the data is latched from the EISA Bus, and then driven back onto the appropriate EISA byte lanes. During a read cycle, the re-drive occurs after the necessary sub-cycles have been completed and the read data has been assembled. For example, when a 32 bit EISA master (other than PCEB) performs 32 bit read from an 8 bit EISA slave, the following sequence of events occurs:

1. The 32-bit EISA master initiates the read cycle. Since the master/slave combination is a mis-match, the master backs off the bus. The EISA master floats its START#, BE[3:0]# and SD[31: 0] lines. The cycle control is then transferred to the ESC.
2. The ESC brings in the first 8 bit data (byte 0) in the first cycle. The ESC asserts SDLE0# to the PCEB.
3. When SDLE0# is asserted, the PCEB latches byte 0 into the least significant byte lane.
4. In the second cycle, the ESC reads the next 8 bit data (byte 1). The PCEB uses SDLE1#, SDCPYUP and SDCPYEN0-1# to latch byte 1 and copy it to the second least significant byte lane (copy-up). This process continues for byte 2 and byte 3. On the fourth cycle, the Dword assembly is complete. During each of the 4 cycles, the ESC generates BE[3:0]# combinations.

5. The ESC instructs the PCEB to re-drive the assembled word to the master by asserting SDOE[2:0]#. In this case, all three SDOE[2:0]# signals are asserted.
6. When SDOE[2:0]# are asserted, the PCEB drives the 32 bit assembled data on SD[31:0] to be latched by the master. The ESC generates the byte enables (BE[3:0]#).
7. The ESC completes the transfer.
8. At the end of the cycle, The ESC transfers control of the EISA Bus back to the EISA master.

During a write cycle, the re-drive occurs after the write data from the master has been latched, and before the data has been disassembled. For example, during a 32-bit write by a 32-bit EISA master to an 8-bit EISA slave, in the first cycle of transfer, the data swap buffers latch the write data (Dword) from the master and drives the first byte back onto the lower byte lane of the EISA Bus. The EISA slave uses the byte enable (BE[3:0]#) combination put out by the EISA master during the first cycle to latch the least significant byte. For the subsequent cycles, the BE[3:0]# combination is generated by the ESC. The PCEB re-drives the second, third and the fourth byte on the second, third and the fourth cycles of the transfer. The number of cycles run is a function of the number of bytes requested (BE[3:0]#), and the master/slave size combinations.

During EISA master and DMA write cycles between master and slave combinations on the EISA/ISA Bus, where only copying is required and no assembly/disassembly is required, the data swap buffer treats this as a re-drive cycle. For example, during a write transfer between a 32-bit EISA master and a 16-bit EISA or ISA slave, where the master is driving data on the upper two byte lanes (BE[3:0]# = 0011), the data swap buffers latch the data on the byte lanes 2 and 3 (Figure 32). The data swap buffers will then re-drive the data onto byte lanes 2 and 3 while copying the data down to byte lanes 0 and 1, for latching by the slave device.

When the PCEB is involved as a master or slave, the re-drive function is disabled. When the PCEB reads 32-bit data from an 8-bit slave the following sequence of events occurs:

1. Same steps as steps 1-4 in the previous example.
2. Once the assembly is complete, the PCEB internally latches the data.
3. The control is transferred back to the PCEB.

NOTE:

During EISA master cycles that require re-driving, the control is transferred from the EISA master to the ESC before the data is re-driven on the data bus. However; during the DMA cycles, the cycle control is maintained by the ESC throughout the entire cycle.

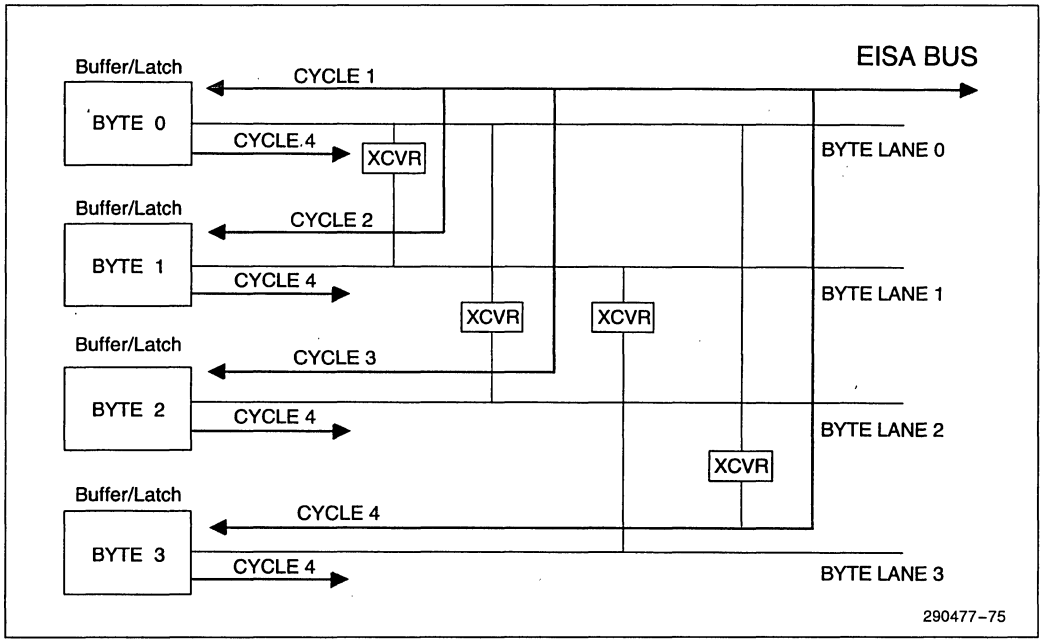


Figure 31. Re-Drive Function: 32-bit EISA Master Accessing an 8-bit EISA or ISA Slave 32-bit Read—BE[3:0]# = 0000

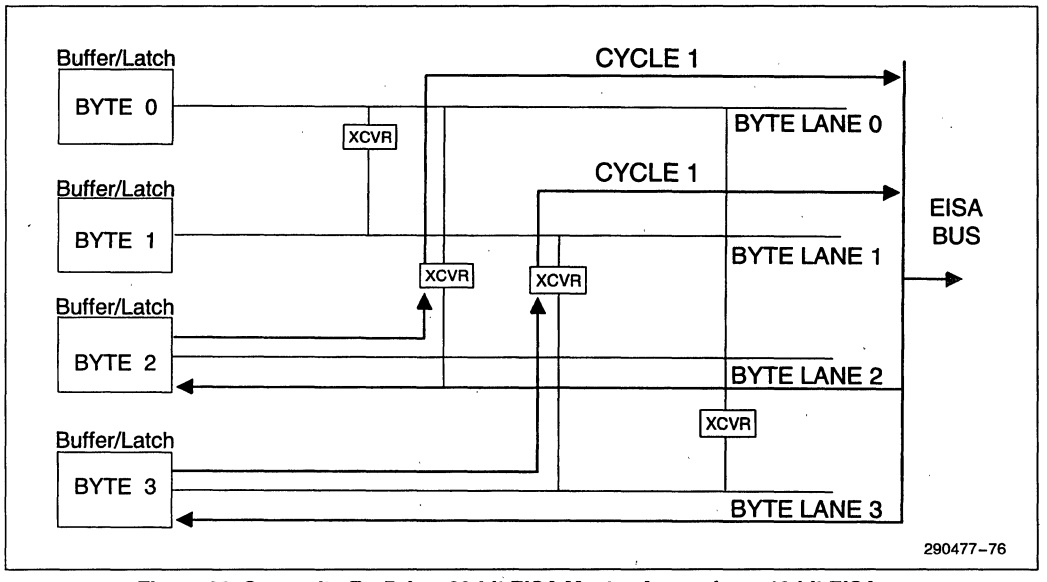


Figure 32. Copy witn Re-Drive: 32-bit EISA Master Accessing a 16-bit EISA or ISA Slave-One Word Write—BE[3:0] # = 0011

9.0 BIOS TIMER

The PCEB provides a system BIOS Timer that decrements at each edge of its 1.03/1.04 MHz clock (derived from the 8.25/8.33 MHz BCLK). Since the state of the counter is undefined at power-up, the BIOS Timer Register must be programmed before it can be used. The timer can be enabled/disabled by writing to the BIOS Timer Address Register.

The BIOS Timer Register can be accessed as a single 16-bit quantity or as 32-bit quantity. For 32-bit accesses, the upper 16 bits are don't care (reserved). The BIOS Timer I/O address location is software programmable. The address is determined by the value programmed into the BTMR Register and can be located on Dword boundaries anywhere in the 64 KByte PCI I/O space.

The BIOS Timer clock has a frequency of 1.03 or 1.04 MHz, depending on the value of BCLK (derived either from 25 MHz or 33 MHz PCICLK). This allows time intervals to be counted from 0 to approximately 65 ms. The accuracy of the counter is ± 1 μs.

9.1 BIOS Timer Operations

A write operation (either 16-bit or 32-bit) to the BIOS Timer Register initiates the counting sequence. After initialization, the BIOS timer starts decrementing until it reaches zero. When the value in the timer reaches zero, the timer stops decrementing and register value remains at zero until the timer is re-initialized.

After the timer is initialized, the current value can be read at any time. The timer can be re-programmed (new initial value written to the BIOS Timer Register) before the register value reaches zero. All write and read operations to the BIOS Timer Register should include all 16 counter bits. Separate accesses to the individual bytes of the counter must be avoided since this can cause unexpected results (incorrect count intervals).

10.0 PCEB/ESC INTERFACE

The PCEB/ESC interface (Figure 33) provides the inter-chip communications between the PCEB and ESC. The interface provides control information between the two components for PCI/EISA arbitration, data size translations (controlling the PCEB's EISA data swap buffers), and interrupt acknowledge cycles.

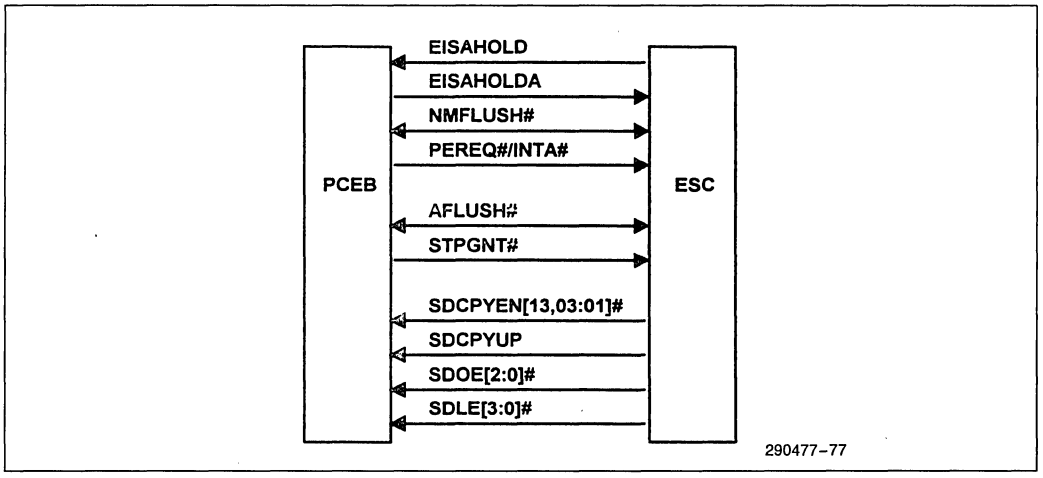


Figure 33. PCEB/ESC Interface Signals

10.1 Arbitration Control Signals

The PCEB contains the arbitration circuitry for the PCI Bus and the ESC contains the arbitration circuitry for the EISA Bus. The PCEB/ESC Interface contains a set of arbitration control signals (EISAHOLD, EISAHOLDA, NMFLUSH#, and PEREQ#/INTA#) that synchronize bus arbitration and ownership changes between the two bus environments. The signals also force PCI device data buffer flushing, if needed, to maintain data coherency during EISA Bus ownership changes.

The PCEB is the default owner of the EISA Bus. If another EISA/ISA master or DMA wants to use the bus, the ESC asserts EISAHOLD to instruct the PCEB to relinquish EISA Bus ownership. The PCEB completes any current EISA Bus transaction, tri-states its EISA Bus signals, and asserts EISAHOLDA to inform the ESC that the PCEB is off the bus.

For ownership changes, other than for a refresh cycle, the ESC asserts the NMFLUSH# signal to the PCEB (for one PCICLK) to instruct the PCEB to flush its Line Buffers pointing to the PCI Bus (Figure 34). The assertion of NMFLUSH# also instructs the PCEB to initiate flushing and to temporarily disable system buffers on the PCI Bus (via MEMREQ#, MEMACK#, and FLSHREQ#). The buffer flushing maintains data coherency, in the event that the new EISA Bus master wants to access the PCI Bus. Buffer flushing also prevents dead-lock conditions between the PCI Bus and EISA Bus. Since the ESC/PCEB do not know ahead of time, whether the new master is going to access the PCI Bus or a device on the EISA Bus, buffers pointing to the PCI Bus are always flushed when there is a change of EISA Bus ownership, except for refresh cycles. For refresh cycles, the ESC controls the cycle and, thus, knows that the cycle is not an access to the PCI Bus and does not initiate a flush request to the PCEB. After a refresh cycle, the ESC always surrenders control of the EISA Bus back to the PCEB.

NMFLUSH# is a bi-directional signal that is negated by the ESC when buffer flushing is not being requested. The ESC asserts NMFLUSH# to request buffer flushing. When the PCEB samples NMFLUSH# asserted, it starts driving the signal in the asserted state and begins the buffer flushing process. (The ESC tri-states NMFLUSH# after asserting it for the initial 1 PCICLK period.) The PCEB keeps NMFLUSH# asserted until all buffers are flushed and then it negates the signal for 1 PCICLK. When the ESC samples NMFLUSH# negated, it starts driving the signal in the negated state, completing the handshake. When the ESC samples NMFLUSH# negated, it grants ownership to the winner of the EISA Bus arbitration (at the time NMFLUSH# was negated). Note that for a refresh cycle, NMFLUSH# is not asserted by the ESC.

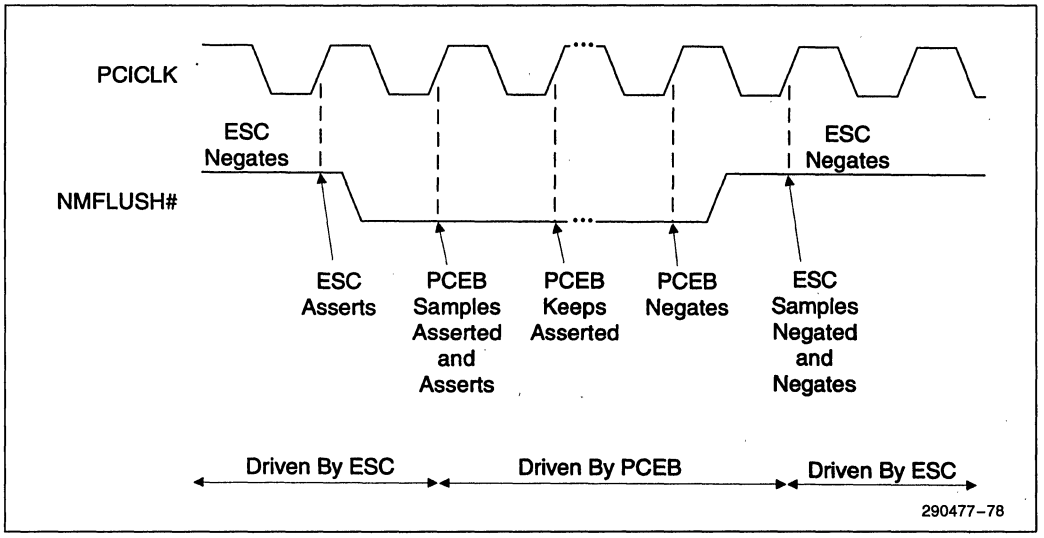


Figure 34. NMFLUSH# Protocol

When the EISA master completes its transfer and gets off the bus (i.e., removes its request to the ESC), the ESC negates EISAHOLD and the PCEB, in turn, negates EISAHOLDA. At this point, the PCEB resumes its default ownership of the EISA Bus.

If a PCI master requests access to the EISA Bus while the bus is owned by a master other than the PCEB, the PCEB retries the PCI cycle and requests ownership of the EISA Bus by asserting PEREQ#/INTA# to the ESC. PEREQ#/INTA# is a dual function signal that is a PCEB request for the EISA Bus (PEREQ# function) when EISAHOLDA is asserted. In response to the PCEB request for EISA Bus ownership, the ESC removes the grant to the EISA master. When the EISA master completes its current transactions and relinquishes the bus (removes its bus request), the ESC negates EISAHOLD and the PCEB, in turn, negates EISAHOLDA. At this point, a grant can be given to the PCI device for a transfer to the EISA Bus. Note that the INTA# function of the PEREQ#/INTA# signal is described in Section 10.3, Interrupt Acknowledge Control.

10.2 System Buffer Coherency Control-APIC

During an interrupt sequence, the system buffers must be flushed before the ESC's I/O APIC can send an interrupt message to the local APIC (CPU's APIC). The ESC and PCEB maintain buffer coherency when the ESC receives an interrupt request for its I/O APIC using the AFLUSH# signal.

10.3 Power Management (82375SB)

In response to the 82375SB ESC's STPCLK# assertion, the CPU sends out a stop grant bus cycle to indicate that it has entered the stop grant state. The PCEB uses the STPGNT# signal to inform the ESC of the stop grant cycle.

10.4 EISA Data Swap Buffer Control Signals

The cycles in the EISA environment may require data size translations before the data can be transferred to its intermediate or final destination. As an example, a 32-bit EISA master write cycle to a 16-bit EISA slave requires a disassembly of a 32-bit Dword into 16-bit words. Similarly, a 32-bit EISA master read cycle to a 16-bit slave requires an assembly of two 16 bit words into a 32-bit Dword. The PCEB contains EISA data swap buffers to support data size translations on the EISA Bus. The operation of the data swap buffers is described in Section 8.0, EISA Data Swap Buffers. The ESC controls the operation of the PCEB's data swap buffers with the following PCEB/ESC interface signals. These signals are outputs from the ESC and an inputs to the PCEB.

- SDCPYEN[13,03:01]#
- SDCPYUP
- SDOE[2:0]#
- SDLE[3:0]#

Copy Enable Outputs (SDCPYEN[13,03:01]#)

These signals enable the byte copy operations between data byte lanes 0, 1, 2 and 3 as shown in the Table 9. ISA master cycles do not perform assembly/disassembly operations. Thus, these cycles use SDCPYEN[13,03:01]# to perform the byte routing and byte copying between lanes. EISA master cycles however, can have assembly/ disassembly operations. These cycles use SDCPYEN[13,03:01]# in conjunction with SDCPYUP and SDLE[3:0]#.

Table 9. Byte Copy Operations

Signal	Copy between Byte Lanes
SDCPYEN01#	Byte 0 (bits[7:0]) and Byte 1 (bits[15:8])
SDCPYEN02#	Byte 0 (bits[7:0]) and Byte 2 (bits[23:16])
SDCPYEN03#	Byte 0 (bits[7:0]) and Byte 3 (bits[31:24])
SDCPYEN13#	Byte 1 (bits[15:8]) and Byte 3 (bits[31:24])

System Data Copy Up (SDCPYUP)

SDCPYUP controls the direction of the byte copy operations. When SDCPYUP is asserted (high), active lower bytes are copied onto the higher bytes. The direction is reversed when SDCPYUP is negated (low).

System Data Output Enable (SDOE[2:0]#)

These signals enable the output of the data swap buffers onto the EISA Bus (Table 10). SDOE[2:0] are re-drive signals in case of mis-matched cycles between EISA to EISA, EISA to ISA, ISA to ISA and the DMA cycles between the devices on EISA.

Table 10. Output Enable Operations

Signal	Byte Lane
SDOE0#	Applies to Byte 0 (bits[7:0])
SDOE1#	Applies to Byte 1 (bits[15:8])
SDOE2#	Applies to Byte 2 and Byte 3 (bits[31:16])

System Data to Internal (PCEB) Data Latch Enables (SDLE[3:0]#)

These signals latch the data from the EISA Bus into the data swap latches. The data is then either sent to the PCI Bus via the PCEB or re-driven onto the EISA Bus. SDLE[3:0]# latch the data from the corresponding EISA Bus byte lanes during PCI Reads from EISA, EISA writes to PCI, DMA cycles between an EISA device and the PCEB. These signals also latch data during mismatched cycles between EISA to EISA, EISA to ISA, ISA to ISA, the DMA cycles between the devices on EISA, and any cycles that require copying of bytes, as opposed to copying and assembly/disassembly.

10.5 Interrupt Acknowledge Control

PEREQ#/INTA# (PCI to EISA Request or Interrupt Acknowledge) is a dual function signal and the selected function depends on the status of EISAHLDA. When EISAHLDA is negated, this signal is an interrupt acknowledge (INTA#) and supports interrupt processing. If interrupt acknowledge is enabled via the PCEB's PCICON Register and EISAHOLDA is negated, the PCEB asserts PEREQ#/INTA# when a PCI interrupt acknowledge cycle is being serviced. This informs the ESC that the forwarded EISA I/O read from location 04h is an interrupt acknowledge cycle. Thus, the ESC uses this signal to distinguish between a request for the interrupt vector and a read of the ESC's DMA register located at 04h. The ESC responds to the read request by placing the interrupt vector on SD[7:0].

11.0 ELECTRICAL CHARACTERISTICS

11.1 Absolute Maximum Ratings

Case Temperature Under Bias ..... −65°C to 110°C

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

Supply Voltages with Respect to Ground ..... −0.5V to V_{CC} + 0.5V

Voltage On Any Pin ..... −0.5V to V_{CC} + 0.5V

Power Dissipation (fully loaded) 0.95W

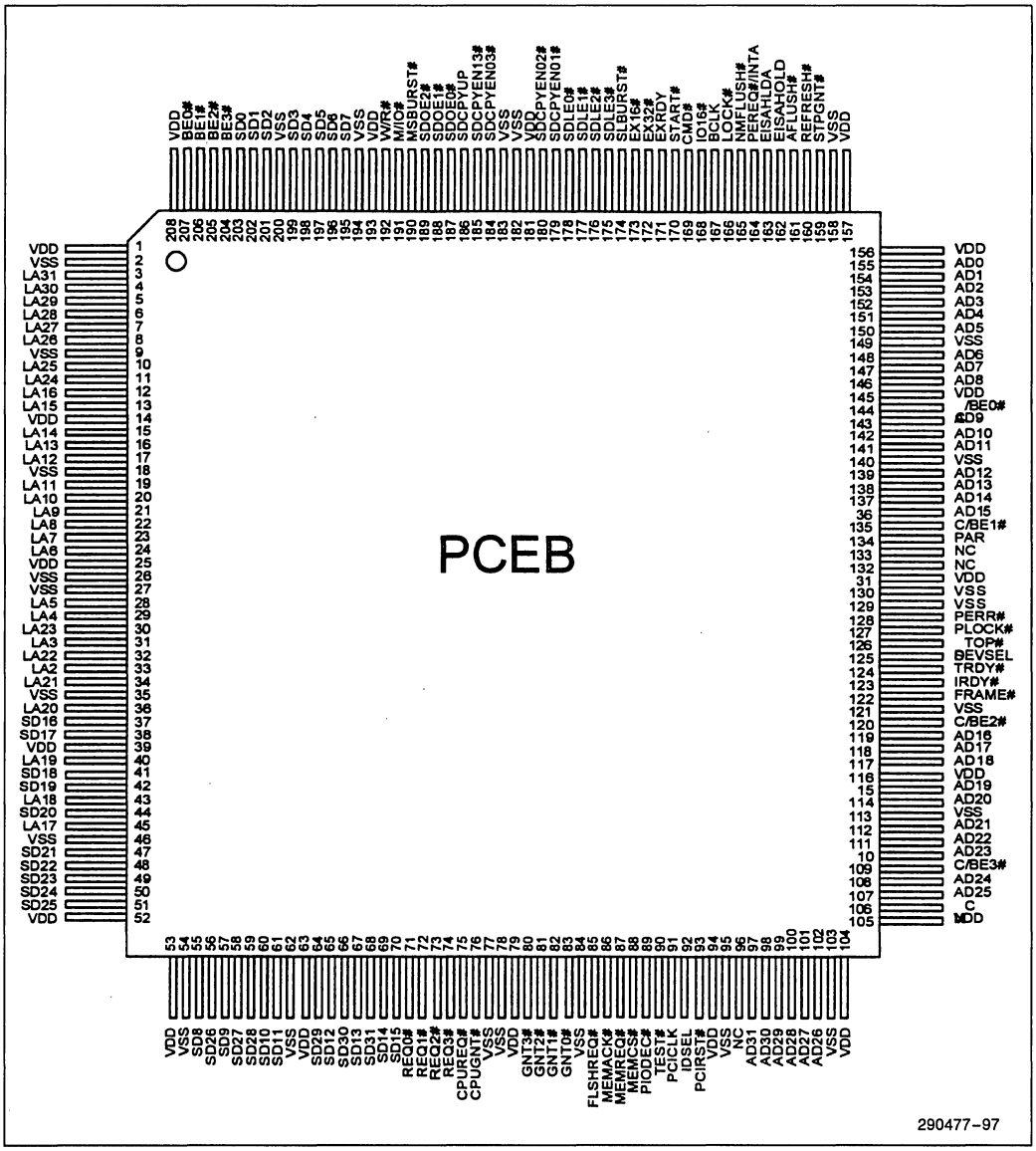
Power Dissipation (four slots) 0.75W

NOTICE: This data sheet contains information on products in the sampling and initial production phases of development. 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.*

12.0 PINOUT AND PACKAGE INFORMATION

12.1 Pin Assignment



290477-97

Figure 35. Pinout

Table 11. Alphabetical PCEB Pin Assignment

Name	Pin #	Type
AD0	155	t/s
AD1	154	t/s
AD2	153	t/s
AD3	152	t/s
AD4	151	t/s
AD5	150	t/s
AD6	148	t/s
AD7	147	t/s
AD8	146	t/s
AD9	143	t/s
AD10	142	t/s
AD11	141	t/s
AD12	139	t/s
AD13	138	t/s
AD14	137	t/s
AD15	136	t/s
AD16	119	t/s
AD17	118	t/s
AD18	117	t/s
AD19	115	t/s
AD20	114	t/s
AD21	112	t/s
AD22	111	t/s
AD23	110	t/s
AD24	108	t/s
AD25	107	t/s
AD26	102	t/s
AD27	101	t/s
AD28	100	t/s
AD29	99	t/s
AD30	98	t/s

Name	Pin #	Type
AD31	97	t/s
AFLUSH#	161	t/s
BCLK	167	in
BE0#	207	t/s
BE1#	206	t/s
BE2#	205	t/s
BE3#	204	t/s
C/BE0#	144	t/s
C/BE1#	135	t/s
C/BE2#	120	t/s
C/BE3#	109	t/s
CMD#	169	in
CPUGNT#	76	out
CPUREQ#	75	in
DEVSEL#	125	s/t/s
EISAHLDA	163	out
EISAHOLD	162	in
EX16#	173	in
EX32#	172	o/d
EXRDY	171	o/d
FLSHREQ#	85	out
FRAME#	122	s/t/s
GNT0#	83	out
GNT1#	82	out
GNT2#	81	out
GNT3#	80	out
IDSEL	92	in
IO16#	168	o/d
IRDY#	123	s/t/s
LA2	33	t/s
LA3	31	t/s
LA4	29	t/s

Table 11. Alphabetical PCEB Pin Assignment (Continued)

Name	Pin #	Type
LA5	28	t/s
LA6	24	t/s
LA7	23	t/s
LA8	22	t/s
LA9	21	t/s
LA10	20	t/s
LA11	19	t/s
LA12	17	t/s
LA13	16	t/s
LA14	15	t/s
LA15	13	t/s
LA16	12	t/s
LA17	45	t/s
LA18	43	t/s
LA19	40	t/s
LA20	36	t/s
LA21	34	t/s
LA22	32	t/s
LA23	30	t/s
LA24	11	t/s
LA25	10	t/s
LA26	8	t/s
LA27	7	t/s
LA28	6	t/s
LA29	5	t/s
LA30	4	t/s
LA31	3	t/s
LOCK#	166	t/s
M/IO#	191	t/s
MEMACK#	86	in
MEMCS#	88	out
MEMREQ#	87	out

Name	Pin #	Type
MSBURST#	190	t/s
NC	96	NC
NC	106	NC
NC	132	NC
NC	133	NC
NMFLUSH#	165	t/s
PAR	134	t/s
PCICLK	91	in
PCIRST#	93	in
PEREQ#/INTA#	164	out
PERR#	128	s/t/s
PIODEC#	89	in
PLOCK#	127	s/t/s
REFRESH#	160	in
REQ0#	71	in
REQ1#	72	in
REQ2#	73	in
REQ3#	74	in
SD0	203	t/s
SD1	202	t/s
SD2	201	t/s
SD3	199	t/s
SD4	198	t/s
SD5	197	t/s
SD6	196	t/s
SD7	195	t/s
SD8	55	t/s
SD9	57	t/s
SD10	60	t/s
SD11	61	t/s
SD12	65	t/s
SD13	67	t/s

Table 11. Alphabetical PCEB Pin Assignment (Continued)

Name	Pin #	Type
SD14	69	t/s
SD15	70	t/s
SD16	37	t/s
SD17	38	t/s
SD18	41	t/s
SD19	42	t/s
SD20	44	t/s
SD21	47	t/s
SD22	48	t/s
SD23	49	t/s
SD24	50	t/s
SD25	51	t/s
SD26	56	t/s
SD27	58	t/s
SD28	59	t/s
SD29	64	t/s
SD30	66	t/s
SD31	68	t/s
SDCPYEN01#	179	in
SDCPYEN02#	180	in
SDCPYEN03#	184	in
SDCPYEN13#	185	in
SDCPYUP	186	in
SDLE0#	178	in
SDLE1#	177	in
SDLE2#	176	in
SDLE3#	175	in
SDOE0#	187	in
SDOE1#	188	in
SDOE2#	189	in
SLBURST#	174	t/s
START#	170	t/s

Name	Pin #	Type
STOP#	126	s/t/s
STPGNT#	159	out
TEST#	90	in
TRDY#	124	s/t/s
V_{DD}	1	V
V_{DD}	14	V
V_{DD}	25	V
V_{DD}	39	V
V_{DD}	52	V
V_{DD}	53	V
V_{DD}	63	V
V_{DD}	79	V
V_{DD}	94	V
V_{DD}	104	V
V_{DD}	105	V
V_{DD}	116	V
V_{DD}	131	V
V_{DD}	145	V
V_{DD}	156	V
V_{DD}	157	V
V_{DD}	181	V
V_{DD}	193	V
V_{DD}	208	V
V_{SS}	2	V
V_{SS}	9	V
V_{SS}	18	V
V_{SS}	26	V
V_{SS}	27	V
V_{SS}	35	V
V_{SS}	46	V
V_{SS}	54	V
V_{SS}	62	V

Table 11. Alphabetical PCEB Pin Assignment (Continued)

Name	Pin #	Type
V_{SS}	77	V
V_{SS}	78	V
V_{SS}	84	V
V_{SS}	95	V
V_{SS}	103	V
V_{SS}	113	V
V_{SS}	121	V
V_{SS}	129	V
V_{SS}	130	V

Name	Pin #	Type
V_{SS}	140	V
V_{SS}	149	V
V_{SS}	158	V
V_{SS}	182	V
V_{SS}	183	V
V_{SS}	194	V
V_{SS}	200	V
W/R#	192	t/s

Table 12. Numerical PCEB Pin Assignment

Pin #	Name	Type
1	V_{DD}	V
2	V_{SS}	V
3	LA31	t/s
4	LA30	t/s
5	LA29	t/s
6	LA28	t/s
7	LA27	t/s
8	LA26	t/s
9	V_{SS}	V
10	LA25	t/s
11	LA24	t/s
12	LA16	t/s
13	LA15	t/s
14	V_{DD}	V
15	LA14	t/s
16	LA13	t/s
17	LA12	t/s
18	V_{SS}	V
19	LA11	t/s
20	LA10	t/s
21	LA9	t/s

Pin #	Name	Type
22	LA8	t/s
23	LA7	t/s
24	LA6	t/s
25	V_{DD}	V
26	V_{SS}	V
27	V_{SS}	V
28	LA5	t/s
29	LA4	t/s
30	LA23	t/s
31	LA3	t/s
32	LA22	t/s
33	LA2	t/s
34	LA21	t/s
35	V_{SS}	V
36	LA20	t/s
37	SD16	t/s
38	SD17	t/s
39	V_{DD}	V
40	LA19	t/s
41	SD18	t/s
42	SD19	t/s
43	LA18	t/s

Table 12. Numerical PCEB Pin Assignment (Continued)

Pin #	Name	Type
44	SD20	t/s
45	LA17	t/s
46	V_{SS}	V
47	SD21	t/s
48	SD22	t/s
49	SD23	t/s
50	SD24	t/s
51	SD25	t/s
52	V_{DD}	V
53	V_{DD}	V
54	V_{SS}	V
55	SD8	t/s
56	SD26	t/s
57	SD9	t/s
58	SD27	t/s
59	SD28	t/s
60	SD10	t/s
61	SD11	t/s
62	V_{SS}	V
63	V_{DD}	V
64	SD29	t/s
65	SD12	t/s
66	SD30	t/s
67	SD13	t/s
68	SD31	t/s
69	SD14	t/s
70	SD15	t/s
71	REQ0#	in
72	REQ1#	in
73	REQ2#	in
74	REQ3#	in
75	CPUREQ#	in

Pin #	Name	Type
76	CPUGNT#	out
77	V_{SS}	V
78	V_{SS}	V
79	V_{DD}	V
80	GNT3#	out
81	GNT2#	out
82	GNT1#	out
83	GNT0#	out
84	V_{SS}	V
85	FLSHREQ#	out
86	MEMACK#	in
87	MEMREQ#	out
88	MEMCS#	out
89	PIODEC#	in
90	TEST#	in
91	PCICLK	in
92	IDSEL	in
93	PCIRST#	in
94	V_{DD}	V
95	V_{SS}	V
96	NC	NC
97	AD31	t/s
98	AD30	t/s
99	AD29	t/s
100	AD28	t/s
101	AD27	t/s
102	AD26	t/s
103	V_{SS}	V
104	V_{DD}	V
105	V_{DD}	V
106	NC	NC
107	AD25	t/s

Table 12. Numerical PCEB Pin Assignment (Continued)

Pin #	Name	Type
108	AD24	t/s
109	C/BE3#	t/s
110	AD23	t/s
111	AD22	t/s
112	AD21	t/s
113	V_{SS}	V
114	AD20	t/s
115	AD19	t/s
116	V_{DD}	V
117	AD18	t/s
118	AD17	t/s
119	AD16	t/s
120	C/BE2#	t/s
121	V_{SS}	V
122	FRAME#	s/t/s
123	IRDY#	s/t/s
124	TRDY#	s/t/s
125	DEVSEL#	s/t/s
126	STOP#	s/t/s
127	PLOCK#	s/t/s
128	PERR#	s/t/s
129	V_{SS}	V
130	V_{SS}	V
131	V_{DD}	V
132	NC	NC
133	NC	NC
134	PAR	t/s
135	C/BE1#	t/s
136	AD15	t/s
137	AD14	t/s
138	AD13	t/s
139	AD12	t/s

Pin #	Name	Type
140	V_{SS}	Vv
141	AD11	t/s
142	AD10	t/s
143	AD9	t/s
144	C/BE0#	t/s
145	V_{DD}	V
146	AD8	t/s
147	AD7	t/s
148	AD6	t/s
149	V_{SS}	V
150	AD5	t/s
151	AD4	t/s
152	AD3	t/s
153	AD2	t/s
154	AD1	t/s
155	AD0	t/s
156	V_{DD}	V
157	V_{DD}	V
158	V_{SS}	V
159	STPGNT#	out
160	REFRESH#	in
161	AFLUSH#	t/s
162	EISAHOLD	in
163	EISAHLDA	out
164	PEREQ#/INTA#	out
165	NMFLUSH#	t/s
166	LOCK#	t/s
167	BCLK	in
168	IO16#	o/d
169	CMD#	in
170	START#	t/s
171	EXRDY	o/d

Table 12. Numerical PCEB Pin Assignment (Continued)

Pin #	Name	Type
172	EX32#	o/d
173	EX16#	in
174	SLBURST#	t/s
175	SDLE3#	in
176	SDLE2#	in
177	SDLE1#	in
178	SDLE0#	in
179	SDCPYEN01#	in
180	SDCPYEN02#	in
181	V_{DD}	V
182	V_{SS}	V
183	V_{SS}	V
184	SDCPYEN03#	in
185	SDCPYEN13#	in
186	SDCPYUP	in
187	SDOE0#	in
188	SDOE1#	in
189	SDOE2#	in
190	MSBURST#	t/s

Pin #	Name	Type
191	M/IO#	t/s
192	W/R#	t/s
193	V_{DD}	V
194	V_{SS}	V
195	SD7	t/s
196	SD6	t/s
197	SD5	t/s
198	SD4	t/s
199	SD3	t/s
200	V_{SS}	V
201	SD2	t/s
202	SD1	t/s
203	SD0	t/s
204	BE3#	t/s
205	BE2#	t/s
206	BE1#	t/s
207	BE0#	t/s
208	V_{DD}	V

12.2 Package Characteristics

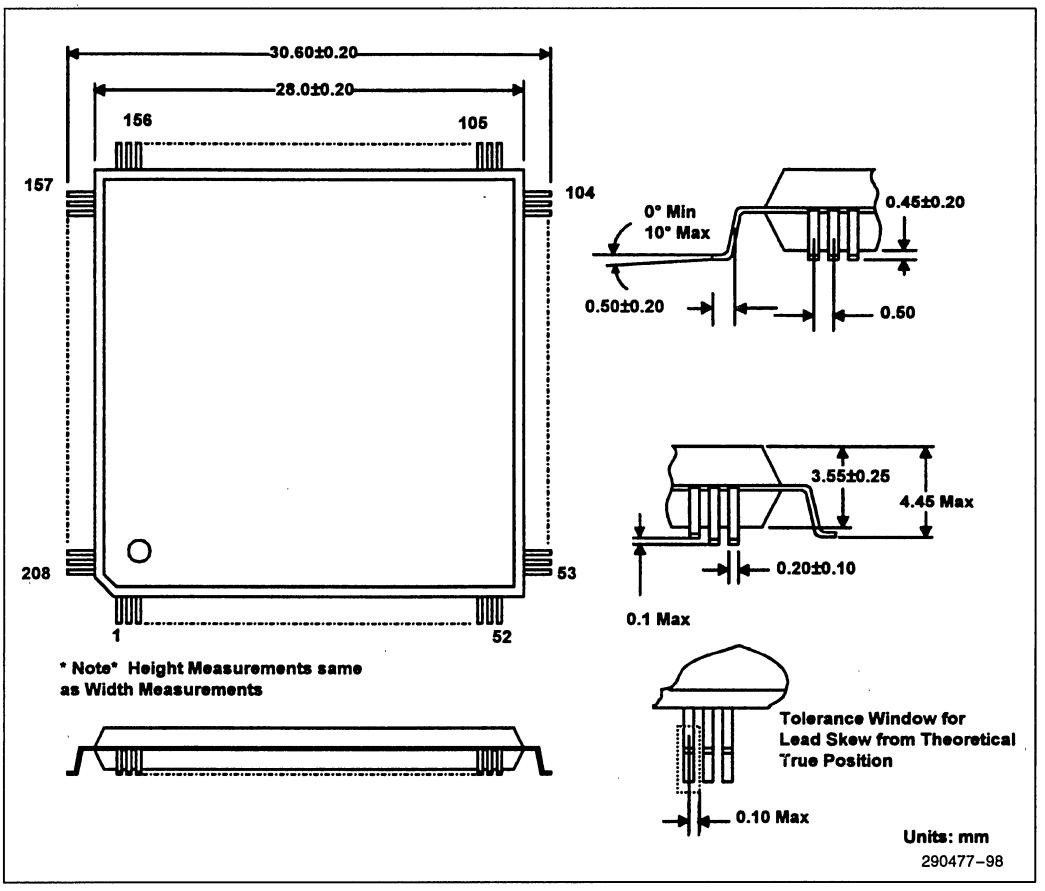


Figure 36. 208-Pin Quad Flat Pack (QFP) Dimensions

13.0 TESTABILITY

13.1 NAND Tree

A NAND Tree is provided primarily for VIL/VIH testing. The NAND Tree is also useful for Automated Test Equipment (ATE) at board level testing. The NAND Tree allows the tester to test the solder connections for each individual signal pin.

The TEST# pin, along with BCLK, PIODEC# and EX16#, activates the NAND Tree. The following combinations of PIODEC#, EX16#, and TEST# causes each buffer to be tri-stated.:

PIODEC# = 1 and EX16# = 0 and TEST# = 0
or
PIODEC# = 0 and EX16# = 1

Care must be taken as the test is in progress to ensure that one of the preceding combinations is valid. Otherwise, the test mode will be exited.

Asserting TEST# causes the output pulse train to appear on the EISAHLDA pin. BCLK must be driven low in order to enable the NAND Tree.

The sequence of the ATE test is as follows:

1. Drive TEST# low, EX16# high, PIODEC# low, and BCLK low.
2. Drive each pin high, except for the pins mentioned in the above discussion (TEST#, PIODEC#, and BCLK).
3. Starting at pin 168 (IO16#) and continuing with pins 169, 170, etc., individually drive each pin low, remembering to toggle PIODEC# from low to high when EX16# is toggled from high to low. Also, when PIODEC# is driven low, EX16# must be driven high. Expect EISAHLDA to toggle after each corresponding input pin is toggled. The final pin in the tree is pin 166 (LOCK#). BCLK is not part of the tree, and EISAHLDA is operated only as an output. Also, note that no-connect (NC), Vcc, and Vxx pins are not part of the NAND Tree.
4. Turn off tester drivers before enabling the PCEB's buffers (via PIODEC#, TEST#, and EX16#).
5. Reset the PCEB prior to proceeding with further testing.

Table 13. NAND Tree Cell Order

Pin#	Name
168	IO16#(1)
169	CMD#
170	START
171	EXRDY
172	EX32#
173	EX32#
174	SLBURST#
175	SDLE3#
176	SDLE2#
177	SDLE1#
178	SDLE0#
179	SDCPYEN01#
180	SDCPYEN02#
184	SDCPYEN03#
185	SDCPYEN13#
186	SDCPYUP
187	SDOE0#
188	SDOE1#
189	SDOE2#
190	MSBURST#
191	M/IO#
192	W/R#
195	SD7
196	SD6
197	SD5
198	SD4
199	SD3
201	SD2
202	SD1
203	SD0
204	BE3#

Pin#	Name
205	BE2#
206	BE1#
207	BE0#
3	LA31#
4	LA30#
5	LA29#
6	LA28#
7	LA27#
8	LA26#
10	LA25#
11	LA24#
12	LA16
13	LA15
15	LA14
16	LA13
17	LA12
18	V_{SS}
19	LA11
20	LA10
21	LA9
22	LA8
23	LA7
24	LA6
28	LA5
29	LA4
30	LA23
31	LA3
32	LA22
33	LA2
34	LA21
36	LA20
37	SD16

Pin#	Name
38	SD17
40	LA19
41	SD18
42	SD19
43	LA18
44	SD20
45	LA17
47	SD21
48	SD22
49	SD23
50	SD24
51	SD25
55	SD8
56	SD26
57	SD9
58	SD27
59	SD28
60	SD10
61	SD11
64	SD29
65	SD12
66	SD30
67	SD13
68	SD31
69	SD14
70	SD15
71	REQ0#
72	REQ1#
73	REQ2#
74	REQ3#
75	CPUREQ#
76	CPUGNT

Table 13. NAND Tree Cell Order

Pin#	Name
80	GNT3#
81	GNT2#
82	GNT1#
83	GNT0#
85	FLSHREQ#
86	MEMACK#
87	MEMREQ#
88	MEMCS#
89	PIODEC#(3)
91	PCICLK
92	IDSEL
93	PCIRST#
97	AD31
98	AD30
99	AD29
100	AD28
101	AD27
102	AD26
107	AD25
108	AD24
109	C/BE3#

Pin#	Name
110	AD23
111	AD22
112	AD21
114	AD20
115	AD19
117	AD18
118	AD17
119	AD16
120	C/BE2#
122	FRAME#
123	IRDY#
124	TRDY#
125	DEVSEL#
126	STOP#
127	PLOCK#
128	PERR#
134	PAR
135	C/BE1#
136	AD15
137	AD14
138	AD13

Pin#	Name
139	AD12
141	AD11
142	AD10
143	AD9
144	C/BE0#
146	AD6
147	AD7
148	AD6
150	AD5
151	AD4
152	AD3
153	AD2
154	AD1
155	AD0
160	REFRESH#
161	AFLUSH#
162	EISAHOLD
164	PEREQ#/INTA#
165	NMFLUSH#
166	LOCK#

NOTES:
1.Start of NAND Tree.
2.Must be 1 when PIODEC# is 0 and must be 0 when PIODEC# is 1.
3.Must be 0 when EX16# is 1 and must be 1 when EX16# is 0.

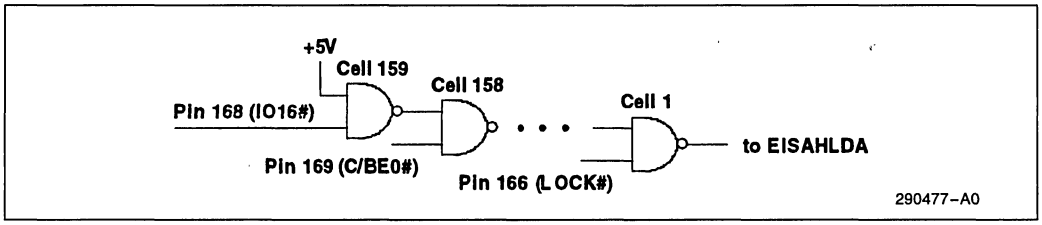


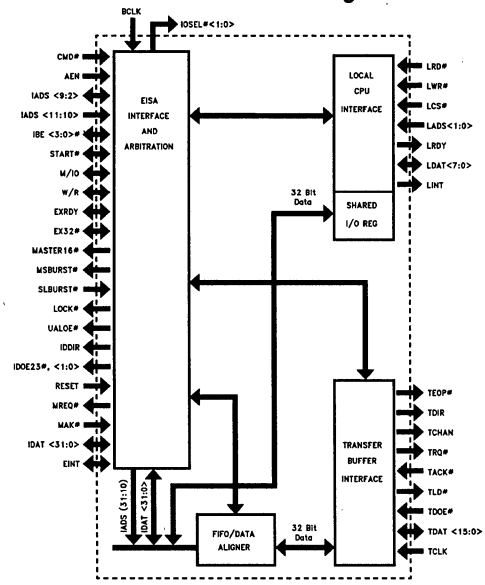
Figure 37. NAND Tree

82355
BUS MASTER INTERFACE CONTROLLER (BMIC)

- **Designed for use in 32-Bit EISA Bus Master Expansion Board Designs**
  - **— Integrates Three Interfaces (EISA, Local CPU, and Transfer Buffer)**
- **Supports 16- and 32-Bit Burst Transfers**
  - **— 33 Mbytes/Sec Maximum Data Transfers**
- **Supports 32-Bit Non-Burst and Mismatched Data Size Transfers**
- **Supports 32-Bit EISA Addressability (4 Gigabyte)**
- **Two independent Data Transfer Channels with 24-Byte FIFOs**
  - **— Expansion Board Timing and EISA Timing Operate Asynchronously**
- **Supports Peek/Poke Operation with the Ability to Access Individual Locations in EISA Memory or I/O space**
- **Automatically Handles Misaligned Doubleword Data Transfers with No Performance Penalty**
- **Supports Automatic Handling of Complete EISA Bus Master Protocol**
  - **— EISA Arbitration/Preemption**
  - **— Cycle Timing and Execution**
  - **— Byte Alignment**
  - **— 1K Boundary Detection**
- **Supports Local Data Transfer Protocol Similar to Traditional DMA**
- **Supports a General Purpose Command and Status Interface**
  - **— Local and EISA System Interrupt Support**
  - **— General Purpose Information Transfers**
  - **— Set-and-Test-Functions in I/O Space (Semaphore Function)**
  - **— Supports the EISA Expansion Board ID Function**
- **Supports Decode of Slot Specific and General I/O Addresses**
- **132-Pin JEDEC PQFP Package**

  (See Packaging Specification Order #240800, Package Type NG)

82355 Internal Block Diagram



290255-1

May 1994
Order Number: 290255-008

82355 Bus Master Interface Controller (BMIC)

CONTENTS

CONTENTS

CONTENTS

1.0 INTRODUCTION

The 82355 Bus Master Interface Controller (BMIC) is a highly integrated Bus Master designed for use in 32-Bit EISA Bus Master expansion board designs and supports all of the enhancements defined in the EISA specifications required for EISA bus master applications. The BMIC provides a simple, yet, powerful and flexible interface between the functions on the expansion board and the EISA bus. With the help of external buffer devices, the BMIC provides all EISA control, address, and data signals necessary to interface to the EISA bus.

The primary function of the 82355 is to support 16- and 32-bit burst data transfers between functions on the EISA expansion board and the EISA bus. Data transfer rates of up to 33 Mbytes/sec are supported (the fastest transfer rate available on an EISA bus). The following logic on the BMIC supports efficient burst transfers:

- Arbitration logic, for gaining control of the EISA bus
- Two transfer-address and byte counters
- Two data FIFOs, which allow expansion board and EISA bus timing to operate asynchronously
- Data shifters, which align data to specific byte boundaries
- A transfer buffer interface, for the data transfers on the expansion board
- General-purpose command and status interface logic
- Local processor interface, to allow programming by an on-board processor
- EISA slave interface, to allow communication with the EISA system

The BMIC greatly simplifies the design of EISA expansion boards. With the 82355, a board can be implemented with simple logic similar to that used in traditional ISA DMA designs. The EISA standard allows designs with 32-bit data and address buses, burst transfers, and automatic handling of the full EISA bus master protocol.

To maximize system throughput, the 82355 BMIC incorporates three fully concurrent interfaces: EISA interface, Transfer Buffer interface, and Local Processor interface. The EISA interface incorporates two 24-byte FIFOs, and implements the full EISA protocol. The Transfer Buffer interface is optimized for high speed static RAM buffers, and can operate at a maximum frequency of 20 MHz. The Local Processor interface supports a generic slave interface, and allows the local processor to fully program the BMIC for operation. Local processors are supported with the ability to access individual locations in system memory or I/O space; this peek-and-poke feature allows the expansion board to communicate easily with other devices in the system. All three interfaces can operate simultaneously, thus maximizing overall system performance.

Address-generation support for the data transfer buffer logic on the expansion board is provided on-chip. The transfer logic on the expansion board can use a high-speed asynchronous transfer clock. The BMIC handles all synchronization with the EISA bus. A FIFO within the BMIC eliminates performance degradation on burst transfers caused by synchronization delays. The BMIC also provides a set of programmable address comparators that drive external chip selects on the expansion board to assist local devices in decoding I/O address ranges.

1.1 BMIC Terminology/Definitions

EISA BUS MASTER—A 32- or 16-bit device that uses the extended part of the EISA bus to generate memory or I/O cycles.

Downshifting Bus Master—A "downshifting" master is a 32-bit master which can convert to a 16-bit master "on the fly". The BMIC will only downshift from a 32-bit master to a 16-bit master if programmed for burst mode (refer to Section 4.2.1).

EISA READ—A data transfer (burst, non-burst (two BCLK), or mismatched) from system to the expansion board across transfer channel 1.

EISA WRITE—A data transfer (burst, non-burst (two BCLK), or mismatched) from the expansion board to system memory across one of the two transfer channels.

I/O ADDRESS DECODE SUPPORT—Refers to slot specific or general I/O address decoding.

Slot Specific Address Decoding—Refers to the decoding of unique addresses allocated to EISA slot specific expansion boards. These addresses are: X000h–X0FFh, X400h–X4FFh, X800h–X8FFh, and XC00h–XCFFh, where X represents the EISA slot number. EISA slot number "0" is reserved for the EISA system board.

General I/O Address Decoding—Refers to the decoding of addresses allocated to ISA expansion boards. These addresses are: 0100h–03FFh.

LOCAL PROCESSOR—A processor located on the expansion board.

SYSTEM CPU—Processor located on the motherboard.

SYSTEM MEMORY—Memory located on the EISA bus or motherboard.

TRANSFER INTERRUPTION—A transfer interruption is defined as an occurrence resulting in a break in a transfer caused by one of the following conditions: A FIFO pause, a FIFO stall, a channel preemption, a channel clear or suspension, a 1K page break, or a transfer complete (EOP).

FIFO Pause—This is a condition where the EISA bus does not provide or take data at a rate fast enough to keep up with the expansion board transfer buffer logic. During an EISA read, this condition is defined as an empty FIFO. During an EISA write, this condition is defined as a full FIFO. A FIFO pause is considered a preferred condition and under normal operations should occur frequently. A FIFO pause will result in the BMIC negating TRQ# until the FIFO becomes not full during an EISA write or not empty during an EISA read.

FIFO Stall—This is a condition where the transfer buffer logic on the expansion board does not provide or take data at a rate fast enough to keep up with the EISA bus. During an EISA read, this condition is defined as a full FIFO. During an EISA write, this condition is defined as an empty FIFO. Under normal operations, a FIFO stall is expected to be a rare and exceptional event. For additional information regarding a FIFO stall, refer to Section 6.2.

Channel Clear—A channel clear results in the immediate termination of the current transfer and the flushing of the channel's corresponding FIFO. A channel clear is initiated by setting the CFGCL bit in the corresponding channel's Configuration register to a 1. For additional information regarding channel clear, refer to Section 8.2.4.2.

Channel Suspension—This temporarily prevents a channel from proceeding with a transfer. A transfer can be temporarily suspended by setting the CFGSU bit in the corresponding channel's Configuration register to a 1.

Channel Preemption—The BMIC can be preempted from the EISA bus by the 82357 (ISP). The 82357 negates MAK# indicating to the BMIC that it must finish the current bus cycle and relinquish control of the EISA bus by negating MREQ# within 64 BCLK periods. The BMIC is programmable to relinquish the bus within 0, 32, or 64 BCLKs from the negation of MAK# (refer to Section 4.4.2).

1K Page Break—The temporary termination of a burst, non-burst (two BCLK), or mismatched data transfer due to a 1K page address boundary crossing (refer to Section 4.2.2).

Transfer Complete (EOP)—End of process due to the transfer byte count being exhausted or a channel being cleared (channel clear). A transfer complete (EOP) will result in the BMIC asserting TEOP# with the last cycle (refer to Section 5.4).

TRANSFER BUFFER LOGIC—Logic located on the expansion board used to support the transfer and storage of data during BMIC EISA master mode transfers between the expansion board and system memory.

The transfer buffer logic interfaces to the Transfer Buffer Interface of the BMIC. Refer to Section 5.2 for additional information regarding transfer buffer logic.

2.0 BMIC INTERFACE ILLUSTRATION

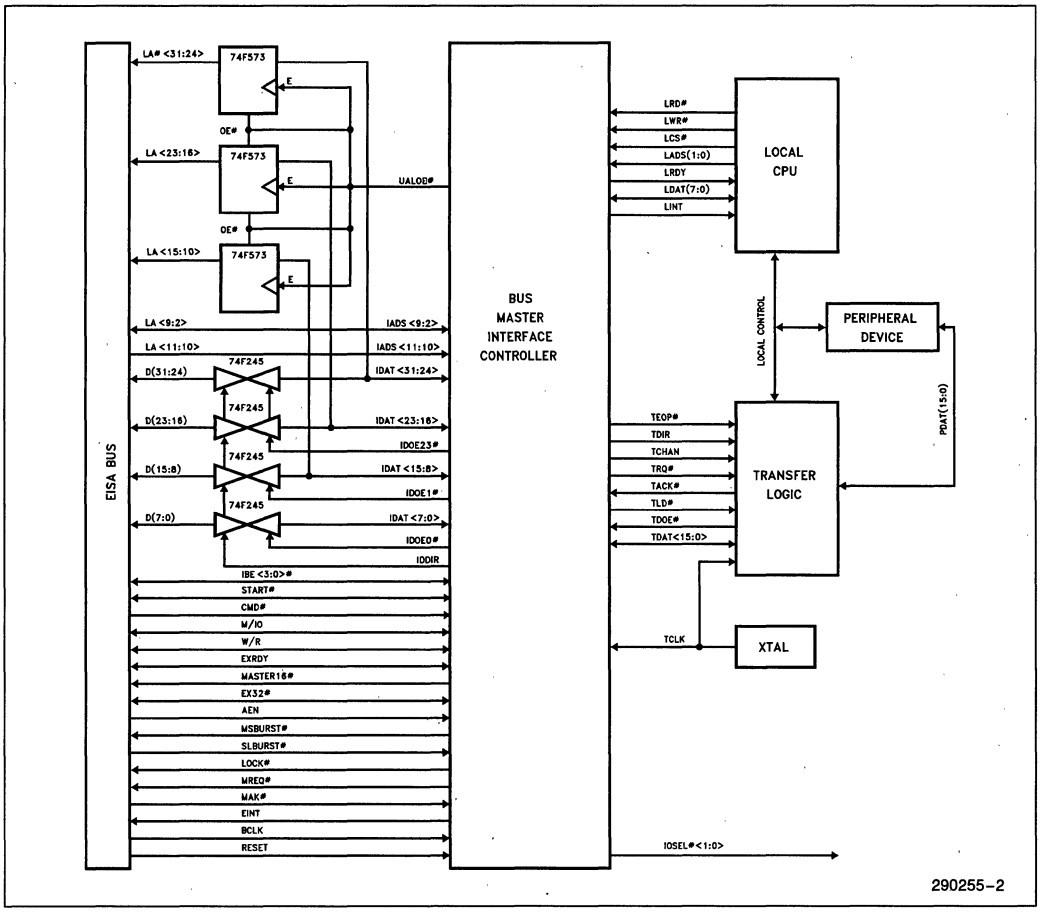


Figure 2-1. BMIC System Interface

3.0 FUNCTIONAL OVERVIEW

The following is a brief discussion of the functional blocks and features of the 82355. The EISA interface, Transfer Buffer interface, FIFO/Data Aligner, and Local interface each have a corresponding detailed section later in this data sheet.

3.1 EISA Master and EISA Slave Operations

In EISA slave mode, the 82355 monitors the EISA address lines <11:2> for general I/O address decoding, slot-specific address decoding, and Shared register accessing. During slave mode operations, all internal registers are accessible through the Local Processor interface, and all Shared registers are accessible through either the Local Processor interface or the EISA interface of the BMIC.

In EISA master mode, the 82355 becomes the master of the EISA bus. It may perform burst, non-burst (two BCLK), mismatched, or Peek/Poke data transfers at this time. During master mode operations, all internal registers are accessible through the Local Processor interface of the BMIC.

The arbiter portion of the BMIC determines which mode the device is in, performs the EISA arbitration, and provides the control signals necessary to regulate the slave and master activities internal to the chip. In slave mode, the arbiter also mediates between the EISA side and the local side during Shared register accesses.

The following is a table of the functions that can be performed during master and slave operations:

	Shared Reg. Accessing	Local CPU Only Reg. Accessing	EISA I/O Address Decoding	Data Transfers
EISA Slave Mode	YES (1, 2)	YES	YES	NO
EISA Master Mode	YES (2)	YES	NO	YES

NOTE:
Shared Reg. Accessing refers to the registers that are accessible through either the EISA interface or Local Processor interface.
Local Processor Only Reg. Accessing refers to the registers that are accessible through the Local Processor interface only.
EISA I/O Address Decoding refers to either general or slot specific I/O decoding support for the expansion board.
Data Transfers refer to either burst, non-burst (two BCLK), mismatched, or peek/poke data transfers.
YES = Can Be Performed
NO = Can Not Be Performed
1 = EISA interface
2 = Local interface

3.2 82355 Internal Architecture Description

The 82355 contains four blocks of control logic. The EISA interface block, Transfer Buffer interface block, FIFO/Data Aligner block, and the Local Processor interface block.

3.2.1 EISA INTERFACE BLOCK

The EISA interface block provides the following functions:

- generates the 32-bit EISA address for burst, non-burst (two BCLK), and peek/poke data transfers
- generates the EISA control signals necessary to implement an EISA 16-bit or 32-bit bus master, and a 32-bit EISA slave
- generates the control signals necessary to enable and disable the external buffer devices
- performs the EISA arbitration and provides the internal control signals required to regulate the slave and master activities of the BMIC
- integrates the registers necessary for the above operations as well as the registers required to provide the configuration and status of the data transfers between the EISA bus and the memory buffer on the expansion board

The EISA memory address range of the 82355 covers the 4 Gigabytes and supports the detection of 1K page address boundaries during burst, non-burst (two BCLK), and mismatched data cycles to and from system memory.

During slave mode, the EISA interface also supports slot specific and general I/O address decode necessary for Shared Register accesses and general decode as required by the expansion board. The shared register addresses are mapped into the slot specific I/O range (C80h-C9Fh).

The EISA interface block contains 43 registers necessary to execute the above functions. A detailed description of the registers and their functions can be found under Register Description (Sections 8.1 and 8.2).

3.2.2 TRANSFER BUFFER INTERFACE BLOCK

The Transfer Buffer interface block provides the group of signals that are required to perform 16-bit data transfers to and from the memory buffer on the expansion board. The protocol used is similar to that found in standard DMA designs. The interface includes a 16-bit data bus (TDAT), seven control signals and a transfer clock (TCLK). The transfer clock can run completely asynchronous to the EISA BCLK signal.

The Transfer Buffer interface block also provides a 16-bit transfer start address which is generated at the beginning of all new data transfers to and from the memory buffer on the expansion board. The 16 TDAT data lines are used to transfer the address.

The Transfer Buffer interface block contains eight registers. A detailed description of the registers and their functions can be found under Register Description (Section 8.2).

3.2.3 FIFO/DATA ALIGNER BLOCK

The FIFO/Data Aligner block is used to isolate and simplify the timing relationships between the EISA bus and the bus master expansion board. This allows the transfer buffer logic and EISA bus timing to operate asynchronously. The FIFO provides the data channel between the EISA bus and the expansion board during BMIC master data transfers and the Data Aligner provides the byte alignment and assembly necessary for the EISA bus.

There are two dual-port, six doubleword wide (24 byte) FIFOs on-board, one per transfer channel. The data is written into the FIFO from either the EISA bus side or the expansion board side, depending on the direction of the transfer. The transfer direction is controlled by a bit in the Transfer Base Count register set.

3.2.4 LOCAL PROCESSOR INTERFACE BLOCK

The Local Processor interface block provides the interface between the BMIC and the local processor. If a local processor is not present, the processor interface can be connected to the ISA bus. The Local Processor interface block is based on an 8086 style slave mode and provides an 8-bit data path for BMIC programming. All of the BMICs internal registers are accessible through this interface.

The Local Processor interface block contains a group of Shared registers used to support general-purpose command and status interactions between the system CPU or EISA bus master and the local processor. In addition to the command/status registers, the CPU interface includes a set of ID registers for EISA expansion board ID support, and a set of Peek/Poke data registers used to hold the data during peek/poke operations.

The local interface portion of the BMIC also contains three 8-bit registers which are used by the local processor to access all of the BMICs internal registers. These registers are mapped into the local processor interface and include a local status register, local data register, and a local index register (refer to Section 3.2.6.1).

The Local Processor block contains 31 registers. A detailed description of the registers and their functions can be found under Register Description (Sections 8.1 and 8.2).

3.2.5 DATA TRANSFER TYPES

The BMIC supports four types of data transfers on the EISA bus: Burst, non-burst (two BCLK), peek/poke or locked exchange, and mismatched. For all of the above transfer types, the addressed slave device can negate EXRDY if wait state timing is required (each wait state is one BCLK).

The primary function of the BMIC is to support 16- and 32-bit burst data transfers between functions on the expansion board and the EISA memory. If the addressed memory is not capable of supporting burst transfers, the BMIC will run either 32-bit non-burst (two BCLK) cycles or, with the support of the 82358 EISA Bus Controller, run mismatched data cycles.

The burst cycle type provides a continuous sequence of one BCLK read or write cycles to and from 16- or 32-bit EISA memory. Burst cycles can not be used with I/O devices or ISA devices (slaves or masters).

The non-burst cycle type provides a continuous sequence of two BCLK read or write transfers to and from 32-bit EISA memory. The BMIC will only respond as a 32-bit master when configured for two BCLK transfers (refer to Section 4.2.1).

The peek/poke and locked exchange feature allows local processor accesses to and from individual I/O space or system memory locations on the EISA bus. The BMIC responds as a 32-bit master and generates two BCLK cycles when configured for peek/poke transfers (refer to Section 4.3). A locked exchange transfer consists of six BCLKs (peek followed by a poke). A peek/poke data transfer has the same timings as a non-burst (two BCLK) data transfer.

The mismatched cycle type provides a means of communicating with 8- or 16-bit EISA or ISA devices. In the event the I/O or memory slave device that has been addressed requires a data size translation, the BMIC will back-off the bus and allow the 82358 EISA Bus Controller to perform the necessary data size translations (refer to Section 4.2.1). The BMIC will generate mismatched cycles as required for all data transfers (burst, non-burst, peek, poke, or locked-exchange).

The following table identifies the BMIC cycle types, master sizes, slave types accessible (memory-I/O), and BCLKs per cycle.

Transfer Type	BMIC Master Size		Slave Type Accessible		BCLKs per Cycle
	16-Bit	32-Bit	I/O	Memory	
Burst	X	X		X	1
Mismatched		X		X	*
Non-Burst		X		X	2
Mismatched		X		X	*
Peek/Poke		X	X	X	2
Mismatched		X	X	X	*
Locked Exchange		X	X	X	6
Mismatched		X	X	X	*

*Depends on slave type/size (EISA/ISA, I/O/Memory, 8-bit/16-bit)

For all of the above transfer types, the addressed slave device can negate EXRDY if wait-state timing is required (each wait-state is one BCLK).

3.2.6 REGISTER ACCESSING

The BMIC provides three distinct groups of registers; the Shared register set, the Local Processor Only register set, and the Index register set. The Shared register set is used by the system CPU or EISA bus master and the local processor for general-purpose command and status interactions and expansion board ID support. The Local Processor only registers are used by the local processor to program the BMIC and provide status for data transfers across the EISA bus and Transfer Buffer interface. The Local Processor Only register set also provides address range decode support for slot specific and general I/O address ranges of interest to the expansion board. The Index register set is used by the local processor as a means of accessing all of the above registers through an indexing scheme.

The Shared register set is accessible through either the EISA interface or the Local Processor interface, the remaining two register sets are accessible through the Local Processor interface only. In the case of contention between the EISA bus and the local processor accessing a Shared register simultaneously, the local processor on the expansion board will have initial priority. Consecutive multiple accesses to the BMIC's shared registers result in a rotational arbitration between the EISA bus and the local processor.

3.2.6.1 Register Accessing through the Local Processor Interface

Register accessing on the local side of the BMIC is accomplished using an indexing procedure. The local interface portion of the BMIC contains two 8-bit registers which are used by the local processor to access all of the BMIC's internal registers. These registers are mapped into the Local Processor interface and include a local data register and a local index register. The registers are selected using the two local address lines (LADS<1:0>). The BMIC's internal register set is read by writing the address of the register to be accessed into the local index register. The register contents are then read through the Local Data register. To write to one of the BMIC's internal registers, the local processor must first write the address of the register to be accessed into the local index register, same as a read, then write the new data value to the Local Data register.

An optional auto-increment mode is supported by the BMIC, which automatically increments the index register after each register read or write. This allows for efficient programming of the register set by using byte string moves. If the Local Index register is given a local index address with bit (7) set high, the local index address will automatically increment each time the Local Data register is read or written.

The Local Status/Control register is directly mapped into the Local Processor interface and is also accessible using the two address lines (LADS<1:0>).

3.2.6.2 Register Accessing through the EISA Interface

The shared registers are mapped directly into the EISA slot-specific I/O space XC80–XC9F. The EISA address lines <11:2> and the byte enables <3:0> are used for decode during shared register accesses.

A standard slave read or write access to the BMIC consists of two BCLKs + one wait-state (one wait-state = one BCLK period). During a slave cycle where the EISA access loses the internal register access through arbitration to the local processor, the cycle will consist of two BCLKs + two wait-states. The BMIC will negate EXRDY for one BCLK for each wait-state required.

3.2.7 INTERRUPTS

The BMIC provides two interrupt request lines, one for the EISA side (EINT), and one for the local side (LINT). The EISA interrupt (EINT) can be programmed for either edge or level-triggered operations. During edge-triggered operations the EINT signal will transition from a low level to a high level. In level-triggered mode, the EISA interrupt signal is an active low open collector output. The local interrupt signal (LINT) can be programmed for either active low or active high level operations and will default to active low operation upon reset. The LINT signal is not an open collector output during active low operations and will require external logic if interrupts need to be tied together on the local side. The EINT and LINT modes of operation are programmed through the Global Configuration register.

3.2.7.1 Interrupt Sources

Several events can trigger each of the two interrupt request signals, and the events can be enabled or disabled on an individual or global basis (refer to Sections 8.1.1.3 and 8.2.2). The system CPU or EISA bus master can only be interrupted by an I/O write from the local processor to the BMIC EISA System Doorbell register. However, the local processor can be interrupted by several sources which are listed below:

— An I/O write from the system CPU or EISA bus master to the BMIC Local Doorbell register.

— The completion of a data transfer on one of the transfer channels.

3.2.7.2 Interrupt Handling

To prevent the BMIC from allowing undetected interrupts from occurring, when servicing an interrupt initiated by the BMIC, all additional interrupts must be disabled prior to reading the Local or EISA System Doorbell Status registers. The interrupts are disabled by writing to the Local or EISA System Doorbell Enable registers, depending on the source of the interrupt.

This is required due to the nature of the interrupt mechanism of the BMIC. All interrupt sources have an edge triggered nature internal to the BMIC, with each event being 'OR'ed together. Additional interrupt sources occurring after the first interrupt will set their appropriate bit in the Status register, but they will not generate an external interrupt until the initial event has been cleared. Thus if the Status register was read first, and another interrupt occurred after this read, the second interrupt would remain undetected in the status register until another event occurred. Disabling of the interrupts prior to reading the status register will prevent this from occurring.

4.0 EISA INTERFACE

4.1 EISA Interface Signals

The BMIC provides a complete interface to the EISA bus and supplies all of the control signals, data lines, and address lines necessary to implement a 16- or 32-bit EISA bus master and a 32-bit EISA slave. This includes a 32-bit data path, a 32-bit address path, and 20 EISA control signals. The BMIC also provides five control signals used to enable and disable the external data buffers and address latches, as shown in Figure 2-1.

The BMIC uses four 74F245 external bidirectional buffers to drive and receive the 32 EISA data and three 74F573 external latches to latch and drive the upper 22 EISA address lines. The external data buffers and address latches should be comprised of "F" or "AS" type logic to meet EISA speed requirements.

The upper 22 EISA addresses are multiplexed through the 22 upper EISA data lines of the BMIC. They are latched externally by the 74F573's. EISA address lines <11:2> and byte enable lines <3:0> are tied directly to the EISA bus. Address lines 10 and 11 are input directly to the BMIC for slave mode address decode. During EISA master operation, lines 10 and 11 are driven indirectly through the external latches.

As a slave, the BMIC receives address lines IADS<11:2> and byte enable lines IBE<3:0># for I/O address decode. Address lines <11:2> are used for slot specific decode and address lines <9:2> are used for general I/O address decode. Address lines <11:2> along with IBE<3:0># are used by the BMIC during Shared register accesses. Address lines <31:12> are not used by the BMIC in slave mode.

The following address lines are used during I/O decoding as shown:
Slot specific I/O address decoding (expansion board)—IADS<11:2>
Slot specific I/O address decoding (shared registers)—IADS<11:2>/IBE<3:0>#
General I/O address decoding (expansion board)—IADS<9:2>

All of the BMIC EISA control signals function as defined in the EISA bus specification. The signals are used to support the following cycles:

BMIC as a Master

	(Cycle Type Performed)			
Master Type	**Burst**	**Non-Burst**	**Mismatched**	**Peek/Poke/Locked Exchange**
32-Bit	X	X	X	X
16-Bit	X			

BMIC as a Slave

1. Responds to EISA shared register accesses as 32-bit slave.
2. Responds to slot specific and general I/O accesses (refer to Section 4.8).

4.2 Transfer Channels

The BMIC contains two identical independent transfer channels which are configurable to run either burst or non-burst (two BCLK) cycles to and from system memory. The BMIC will automatically run non-burst (two BCLK) or mismatched cycles if the memory the BMIC has addressed cannot run burst cycles. Mismatched cycles will be run if data size translation is required.

Channel 0 must be used for EISA READ operation only. Channel 1 can be used for both EISA READ and EISA WRITE operations.

Each channel has three sets of registers to regulate data transfers. These are the Base register group, the Current register group, and the Data Status/Control register group. This implementation of a triple register set allows a processor to begin programming the next transfer on the channel while the current transfer is being executed.

The Base register set contains seven 8-bit registers. These registers are programmed by the local processor when a transfer is required across one of the channels. Four Transfer Channel Base Address registers are combined to form the starting 32-bit EISA address to be used during the transfer. The remaining three registers are the Transfer Channel Base Count Registers. The Base Count registers are combined to determine the number of transfers (in bytes) to be performed. The number of bytes which can be transferred ranges from 1 byte to 4 Mbytes. The most significant bit of the Transfer Channel Base Count register group is used to control the start of the transfer and the second most significant bit is used to control the direction of the transfer (refer to Section 8.2.3.3).

The Current register set contains seven registers each of which corresponds to a Base register. These registers are loaded from the Base registers. The Transfer Channel Current Address registers contain the 32-bit real-time EISA memory address. The Transfer Channel Current Count registers contain the number of bytes remaining to be transferred on the channel. The current register set is readable by the local processor. However, there are possible coherency problems involved with reading multiple bytes while the current registers are being updated during a transfer. To avoid these problems, a channel's transfer should be temporarily suspended (using the channel's Configuration Register) before trying to read the channel's current register set.

The Status/Control register set contains three registers: the Transfer Channel Strobe register, Transfer Channel Configuration register, and the Transfer Channel Status register. The Transfer Channel Strobe register is used to initiate the transfer of data from the Base register set to the associated Current register set. A transfer request for that channel will be generated following the Current register load. The Transfer Channel Configuration register is used to program the mode of the transfer. The Transfer Channel Status register provides current FIFO and transfer channel status.

To initialize a transfer over either of the two transfer channels, the following steps must be completed:

1. Verify that the Base registers for the desired transfer channel are available.

   The Transfer Channel Base Address and Base Count registers must be available before they can be programmed. This is determined by the status of bits 0 and 1 in the Local Status/Control register. A "1" in either of the two bits indicates that the corresponding channel is currently running a transfer and the Base registers are busy. A "0" indicates that the Base registers are free and available for programming. In the event that the Base registers are not available, the local processor must wait until the data transfer executing on the requested channel has completed, at which time bits "0" or "1" (depending on which channel was programmed) in the Local Status/Control registers will be reset to 0. Programming the Base registers during a Base register Busy state, is illegal and will corrupt the Base register data of the pending transfer. Programming the Transfer Configuration register during a cycle in progress may cause the termination of the transfer, depending on which bit in the register was changed.
2. Program the transfer channel's associated Transfer Base register set with the desired transfer information (Base registers must be available).
3. Initiate the Base register to Current register load and schedule a transfer request by writing to the channel's Transfer Strobe register.

   If a transfer is in progress on the requested channel and a write to the associated channel's Strobe register is done, the Base to Current register load will take place immediately after the data transfer on the requested channel has completed.

4.2.1 BURST AND NON-BURST MODES OF OPERATION

The BMIC can be programmed for burst or non-burst (two BCLK) data transfers to and from EISA memory. This is determined by a write to the Channel Configuration Register.

If burst mode is enabled, the BMIC will look for the SLBURST# signal at the beginning of the transfer to determine if the slave device that was addressed is

capable of running burst cycles. If the slave device does not respond with an active SLBURST# signal, the BMIC will not activate the MSBURST# signal and will proceed with either non-burst (two BCLK) bus cycles or mismatched cycles.

In burst mode, the BMIC can respond as a 16- or 32-bit master. The BMIC informs the system of this capability by driving MASTER16# low from the same BCLK rising edge that START# is asserted. MASTER16# will remain low for one BCLK. The BMIC will automatically "downshift" from a 32- to a 16-bit master if the EX32# signal is sampled inactive and the SLBURST# signal is sampled active at the beginning of a transfer. If EX32# and SLBURST# are sampled active at the beginning of the transfer, the BMIC will proceed with a 32-bit burst transfer.

In non-burst mode, the BMIC will respond as a 32-bit master. The BMIC will look for the EX32# signal at the beginning of the transfer to determine if the system memory it has addressed has the same bus width. If the EX32# signal is not returned (mismatched cycle indicated), the BMIC will "back-off" the bus by floating START#, IBE#<3:0>, and IDAT<31:0> to allow the 82358 EISA Bus Controller to take control of the transfer. The EISA Bus Controller will then proceed to assemble or disassemble the data as needed. The EISA Bus Controller will return the EX32# signal after the mismatched cycle is complete, indicating to the BMIC that a new address can be placed on the bus. If the EX32# signal is sampled active at the beginning of the transfer, the BMIC will proceed with a 32-bit non-burst (two BCLK) transfer.

4.2.2 1K PAGE ADDRESS BOUNDARY DETECTION

During burst, non-burst (two BCLK), and mismatched data cycles, the BMIC provides the support to detect 1K page address boundary crossings. If the BMIC detects that the current cycle is about to cross a 1K page boundary, the transfer will be temporarily terminated on the next cycle. The BMIC will then arbitrate between restarting the transfer on the current channel, selecting the second channel, doing a peek/poke cycle, or preempting the channel (refer to Section 4.4 for information regarding BMIC arbitration).

Example: Transfer = 32-bit transfer and page address boundary is at location 400h = 1024

1. The BMIC detects that the current cycle is about to cross a 1K page address boundary—current address (3FCh = 1020).
2. Address after BMIC has executed the current cycle (400h = 1024).
3. Transfer is temporarily terminated (interrupted).
4. BMIC will now arbitrate between restarting the transfer on a new page, selecting the second channel, doing a peek/poke cycle, or preempting the channel.

4.3 Peek/Poke, Locked Exchange Transfers

To allow the local processor to communicate with other devices in the main system, the BMIC allows the local processor to execute individual I/O or memory cycles over the EISA bus. These cycles can be thought of as being similar to "peek" and "poke" statements in the Basic programming language. These cycles may be reads, writes, or locked exchanges in 8-, 16-, 24-, or 32-bit values. All cycles must be contained within a single doubleword.

The Peek/Poke operation requires the following set of registers: Four 8-bit Peek/Poke Address registers which are combined to provide the 32-bit Peek/Poke address; One 8-bit Peek/Poke Control register which contains the bits defining whether the cycle is I/O or memory, peek (read)/poke (write) or locked exchange, and which byte enables are to be active during the cycle; and four 8-bit Peek/Poke Data registers which are used to hold the data for the Peek/Poke cycle. During all peek/poke or locked exchange cycles, byte enables IBE<3:0># are derived from bits 0–3 in the Peek/Poke Control register set. The lower two bits of the Peek/Poke Address register are ignored. Peek, poke, or locked exchange cycles will not be generated for illegal combinations of byte enables (i.e., 1111, 1010, 0110, 0101, 0100, 0010).

To do an individual write cycle (poke), the local processor must first write to the Peek/Poke Address register set to specify the 32-bit memory address or the 16-bit I/O address. It must then write the data to be transferred into the Peek/Poke Data register set. The data must be placed in the appropriate byte positions in the Data register set so that it goes out on the correct byte lanes during a 32-bit bus master transfer.

Once the appropriate data and address have been programmed, the local processor must write to the Peek/Poke Control register to specify the cycle type and initiate the cycle. After this write to the Peek/Poke Control register, bit 2 in the Local Status/Control register will be set to a 1 by the BMIC to indicate that a peek/poke request is pending and that the peek/poke registers are busy. When the poke cycle has finished executing on the EISA bus, the Peek/Poke status bit 2 in the Local Status/Control register will return to normal (0).

To do an individual read cycle (peek), the local processor must write to the Peek/Poke Address registers, then to the Peek/Poke control register to initiate the read cycle. The Peek/Poke status bit 2 in the Local Status/Control register will be set high by the BMIC and remain active until the peek cycle finishes on the EISA bus. The local processor can then read the data from the Peek/Poke data registers.

NOTE:
When running consecutive peek transfers, the data must be read from the Peek/Poke data registers before each new peek transfer is generated. The BMIC will read the data off the EISA bus from all four byte lanes regardless of which Byte enables (IBE<3:0>#) are active. (Although all bytes are read, the value of the byte enables are important to the system and must be programmed for the peek transfer).

When a locked exchange cycle is requested by the local processor, a peek cycle is scheduled first and then immediately followed by a poke cycle. The LOCK# signal is active during the locked exchange cycle to indicate to the system that no other accesses to the addressed location can be made.

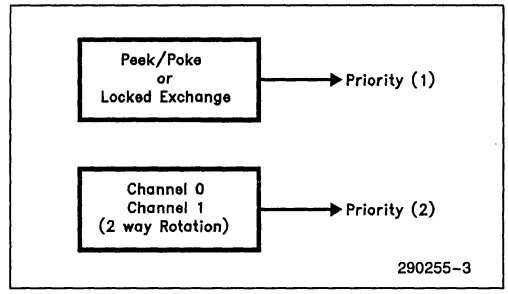
Whenever the BMIC is commanded to do an EISA POKE cycle, the BMIC will assert the MREQ# signal low normally, transfer up to four bytes of data, and release the bus by de-asserting MREQ# high. A potential problem exists, however, when the slave device extends the cycle by de-asserting EXRDY low. If the slave holds this signal low past the time that the BMIC is forced to release MREQ# high (it has been preempted while waiting for the slave to assert EXRDY high), then the BMIC will drive MREQ# back low again immediately after this cycle ends if there is another transfer pending (TBI, PEEK, POKE or LOCKED-EXCHANGE). Note that according to the EISA spec, MREQ* signal description "A bus master must wait at least two BCLKs after releasing the bus before reasserting its MREQx*". To adhere to EISA specifications, it is required that LOCKED-EXCHANGE cycles be used in lieu of POKE cycles.

Any consecutive Peek/Poke or Locked exchange transfers must be initiated only after the previous Peek/Poke or Locked exchange has been completed. This can be accomplished by making sure that bit 2 of the local status/control register is set to a zero before initiating the transfer.

4.4 Arbitration

4.4.1 EISA/BMIC ARBITRATION

The BMIC will begin master mode operation any time a transfer request is pending. If more than one transfer request is pending, the BMIC will service them in the following order. Peek/Poke cycles have the highest priority access to the EISA bus followed by the two data channels. Once the BMIC has gained control of the EISA bus, the BMIC will first perform any peek, poke, or locked exchange transfers that may be pending. If there are no peek, poke, or locked exchange transfers pending, the BMIC will run data transfers initiated by either of the two transfer channels. The two transfer channels have equal priority with respect to each other and are serviced in an alternating fashion. The priorities and assignments are as follows:



The BMIC will maintain ownership of the EISA bus until it has serviced all outstanding data transfer requests or it is preempted from the bus by the removal of the MAK# signal. The BMIC can be configured to relinquish the EISA bus immediately, 4 μs, or 8 μs after a preempt is received. If the BMIC has completed all outstanding data transfer requests prior to the time-out of the preempt timer, it will give up the bus. If the BMIC finishes one task prior to the time-out of the preempt timer, it will start on the next pending transfer request unless the request is a peek, poke, or locked exchange cycle. The BMIC will not start a set of peek, poke, or locked exchange cycles after the MAK# signal has been removed. If a transfer is cut-off due to a preempt timer time-out, the BMIC, upon regaining access to the EISA bus and following its internal arbitration priority scheme, will continue the transfer that was preempted at the point the transfer was cut-off.

When a channel is interrupted for any reason, 1K page break, FIFO stall, channel clear, channel suspend, or transfer complete, the BMIC may immediately relinquish the EISA bus depending on the state of the CFGFF bit in the Channel Configuration register set.

NOTE:
During a FIFO pause, the CFGFF bit in the associated Channel's Configuration register is ignored. The function of the CFGFF bit, as related to the above channel interruptions, is as follows:

If the CFGFF bit = 1, the BMIC will immediately relinquish control of the EISA bus upon the detection of any of the above interruptions. This will occur

regardless if there are additional data transfer requests pending. If there are additional data transfer requests pending, the BMIC will reassert MREQ# a minimum of two BCLKs later to reacquire the EISA bus. The BMIC will follow the arbitration priority scheme outlined above when servicing a data transfer request after a transfer interruption has occurred.

If the CFGFF bit = 0, the BMIC retains ownership of the EISA bus upon detection of a FIFO stall or 1K page break as long as a preempt timer timeout has not occurred. If there are additional data requests pending, the BMIC will immediately perform the pending transfer and then rearbitrate for the EISA bus to complete the interrupted transfer. If there are no additional data requests pending, the BMIC will relinquish ownership of the EISA bus only after the current transfer interruption has been serviced and completed.

4.4.2 BMIC PREEMPT TIMER

The BMIC can be preempted from the EISA bus by the 82357 (ISP). The 82357 negates MAK#, indicating to the BMIC that it must finish the current bus cycle and relinquish control of the EISA bus by negating MREQ# within 64 BCLK periods (8 μs).

The BMIC provides a programmable preempt timer which can be programmed to relinquish the bus within 3, 32, or 64 BCLKs. The preempt timer is programmable through the Global Configuration register.

The following diagrams illustrate the latest the BMIC will start a new transfer after MAK# has been negated.

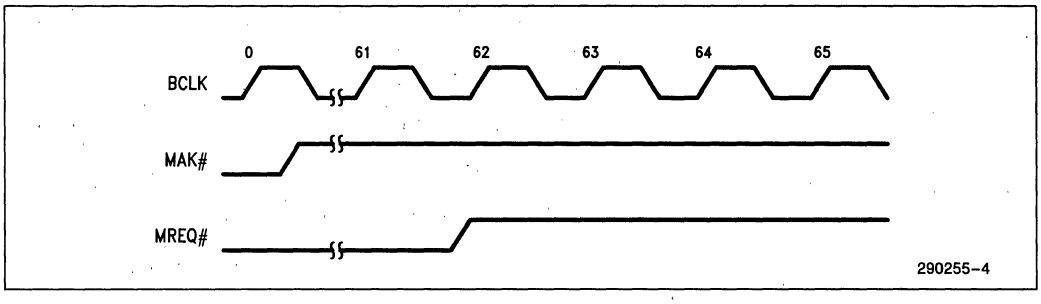
Depending on the type of transfer started, the BMIC will respond as follows:

Assumptions:

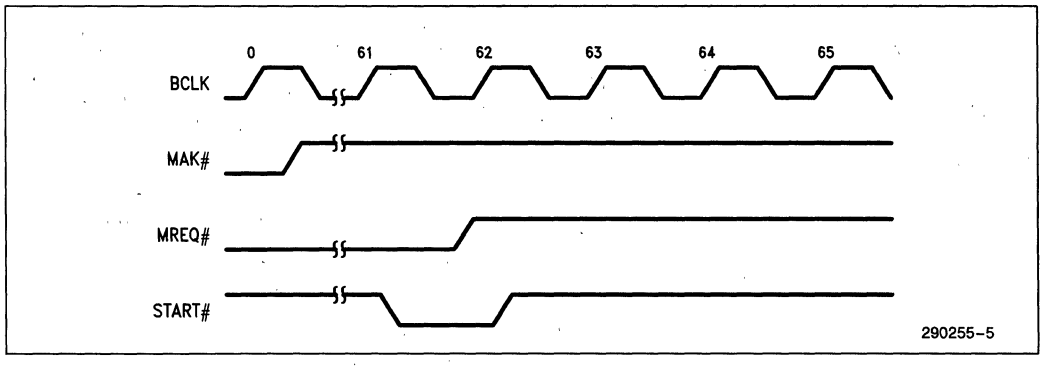
1. The 82357 has negated the MAK# signal at BCLK zero.
2. The preempt timer is programmed to relinquish the EISA bus within 64 BCLKs after the negation of MAK#.
3. Let X = programmed value of preempt delay (in BCLKs).

BMIC Response:

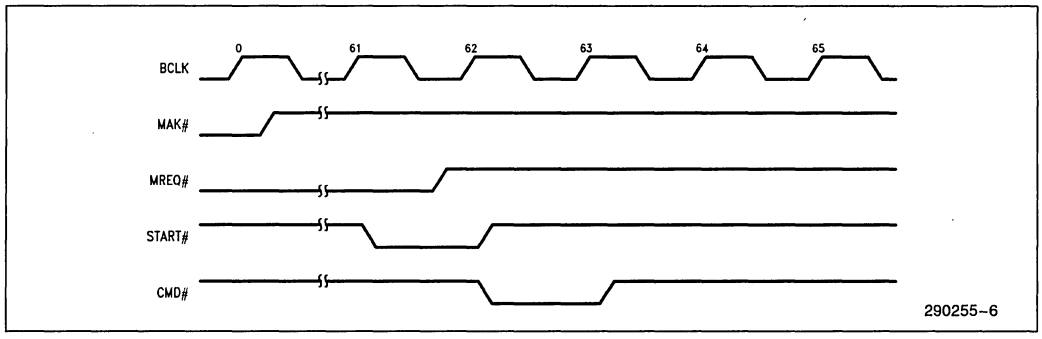
For all transfers, the BMIC will negate MREQ# within (X-2.5) BCLK periods following the MAK# transition to an inactive state (BCLK 61.5).



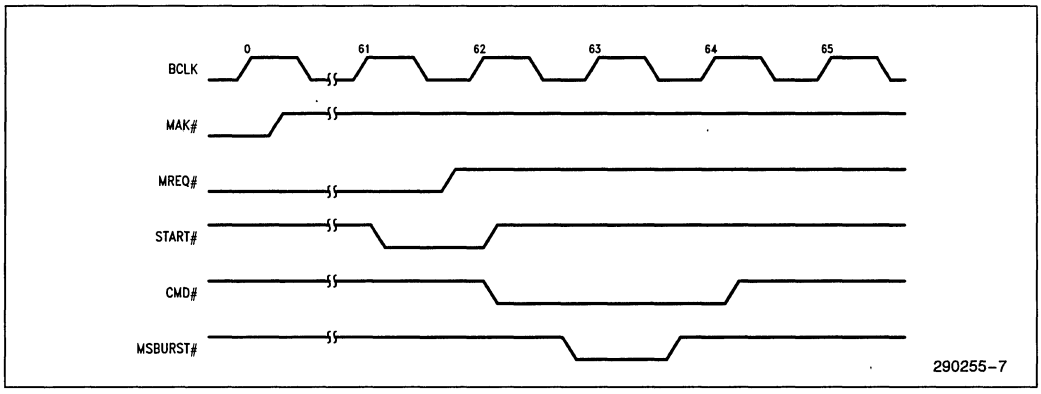
For all transfers, the BMIC may assert START# on any of the first X-3 rising edges of BCLK following the MAK# transition to an inactive state (BCLK 61).



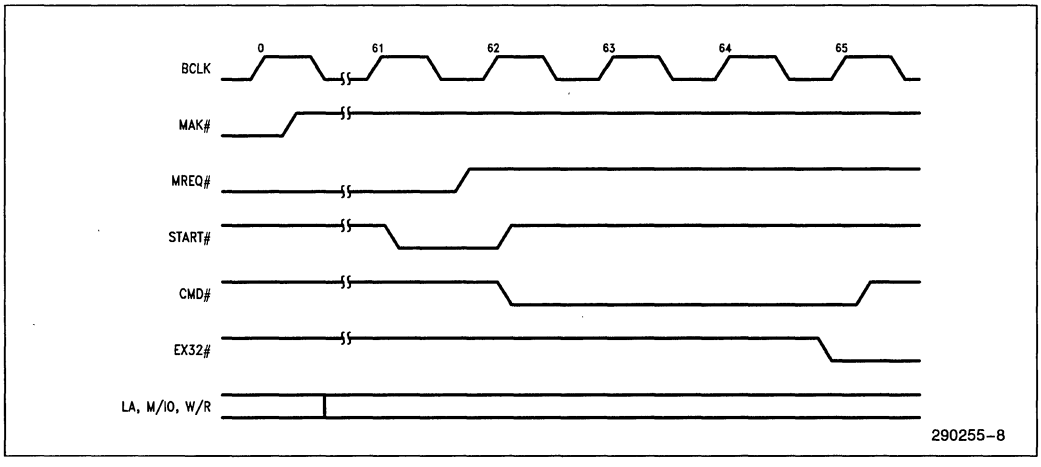
If the last cycle is a non-burst two BCLK cycle, CMD# will become inactive within (X-1) BCLK periods from the inactive transition of MAK# (BCLK 63), this is assuming that EXRDY is active.



If the last cycle is a burst EISA cycle, the BMIC will negate MSBURST# within (X-0.5) BCLK periods from the inactive transition of MAK# (BCLK 63.5). The last CMD# will go inactive within X BCLK periods from the deassertion of MAK# (BCLK 64). This is assuming EXRDY is active.



If the last cycle is mismatched, cycle completion will be controlled by the system. The BMIC will drive the LA address, M/IO, and W/R signals until the falling edge of BCLK after the last CMD# inactive transition.



4.5 EISA Address Incrementer

The Transfer Channel Current Address register set for each channel functions as an address incrementer and is used to generate and track the address of the data during transfers. The register set increments the address according to the number of bytes being transferred during that cycle. The transfer is automatically aligned on doubleword boundaries. The two least significant bits of the starting 32-bit address (A0 and A1) are used to determine the initial address increment value.

For 32-bit transfers, the BMIC provides an initial address increment of 1, 2, 3 or 4 depending on the value of address lines A<1:0>. After the initial increment, the BMIC increments the address by 4 until the last cycle is detected.

The following example illustrates the BMIC address incrementer during a 32-bit master mode transfer.

		EISA Address			
		A3	A2	A1	A0
Start Address	FFFFF001h	0	0	0	1
Initial Increment (Incremented by 3)	FFFFF004h	0	1	0	0
All Increments Following (Incremented by 4)	FFFFF008h	1	0	0	0
	FFFFF00Ch	1	1	0	0

The starting address A<1:0> is 01, this means that the initial increment must be 3 in order to align the next increments on doubleword boundaries. The subsequent increments will be by 4 until the last cycle is detected.

For 16-bit transfers, the BMIC provides an initial address increment of 1 or 2 depending on the status of address lines A<1:0>. After the initial increment, the BMIC increments the address by two until the last cycle is detected.

The following example illustrates the BMIC address incrementer during a 16-bit master mode transfer.

		EISA Address			
		A3	A2	A1	A0
Start Address	FFFFF001h	0	0	0	1
Initial Increment (Incremented by 1)	FFFFF002h	0	0	1	0
All Increments Following (Incremented by 2)	FFFFF004h	0	1	0	0
	FFFFF006h	0	1	1	0

The starting address A<1:0> is 01, this means that the initial increment must be 1 in order to align the next increments on singleword boundaries. The subsequent increments will be by 2 until the last cycle is detected.

NOTE:
The BMIC internally assembles 32-bit dwords. When a 16-bit burst transfer is preempted, the transfer will stop on a doubleword boundary.

4.6 EISA Byte Decrementer

The Transfer Channel Current Count register set for each channel contains the intermediate value of the byte count during the transfer and is used as the byte decrementer. The decrementer's function is partially based upon the address incrementer. In the above 32-bit incrementer example, the byte count would be decremented by 3 on the first cycle. After the initial decrement, the channel's Current Count register set is decremented by 4 until the last cycle is detected. In the above 16-bit incrementer example, the byte count would be decremented by 1 on the first cycle. After the initial decrement, the channel's Current Count register set is decremented by 2 until the last cycle is detected. Note that the Current Count register does not decrement entirely to zero. Instead, it retains the value of the number of bytes transferred during the last cycle.

4.7 EISA Address Incrementer/Byte Decrementer Illustration

The following table illustrates the various states of (A0, A1) vs the transfer byte-count and the initial address during a 32-bit transfer.

Byte Count	Starting Address	Next Address	Initial Increment	Number of Bytes Left	Last Cycle	Number of Cycles Left
1	XXX 0000	NA	NA	0	Yes	0
	XXX 0001	NA	NA	0	Yes	0
	XXX 0010	NA	NA	0	Yes	0
	XXX 0011	NA	NA	0	Yes	0
2	XXX 0000	NA	NA	0	Yes	0
	XXX 0001	NA	NA	0	Yes	0
	XXX 0010	NA	NA	0	Yes	0
	XXX 0011	XXX 0100	1	1	No	1
3	XXX 0000	NA	NA	0	Yes	0
	XXX 0001	NA	NA	0	Yes	0
	XXX 0010	XXX 0100	2	1	No	1
	XXX 0011	XXX 0100	1	2	No	1
4	XXX 0000	NA	NA	0	Yes	0
	XXX 0001	XXX 0100	3	1	No	1
	XXX 0010	XXX 0100	2	2	No	1
	XXX 0011	XXX 0100	1	3	No	1
5	XXX 0000	XXX 0100	4	1	No	1
	XXX 0001	XXX 0100	3	2	No	1
	XXX 0010	XXX 0100	2	3	No	1
	XXX 0011	XXX 0100	1	4	No	1
6	XXX 0000	XXX 0100	4	2	No	1
	XXX 0001	XXX 0100	3	3	No	1
	XXX 0010	XXX 0100	2	4	No	1
	XXX 0011	XXX 0100	1	5	No	2
7	XXX 0000	XXX 0100	4	3	No	1
	XXX 0001	XXX 0100	3	4	No	1
	XXX 0010	XXX 0100	2	5	No	2
	XXX 0011	XXX 0100	1	6	No	2
8	XXX 0000	XXX 0100	4	4	No	1
	XXX 0001	XXX 0100	3	5	No	2
	XXX 0010	XXX 0100	2	6	No	2
	XXX 0011	XXX 0100	1	7	No	2
9	XXX 0000	XXX 0100	4	5	No	2
	XXX 0001	XXX 0100	3	6	No	2
	XXX 0010	XXX 0100	2	7	No	2
	XXX 0011	XXX 0100	1	8	No	2
10	XXX 0000	XXX 0100	4	6	No	2
	XXX 0001	XXX 0100	3	7	No	2
	XXX 0010	XXX 0100	2	8	No	2
	XXX 0011	XXX 0100	1	9	No	3

NOTES:
1. "X" = Don't Care
2. If the "byte count" is less than or equal to the "initial increment", then the current cycle = the first cycle = the last cycle.
3. If the number of bytes left is less than or equal to 4, then the next cycle = the last cycle.
4. For information regarding byte alignment, refer to Section 6.3.1.

4.8 I/O Address Range Decode Support

The BMIC provides on-board decoder logic, two I/O select pins (IOSEL<1:0>#), and a set of 8-bit I/O Decode Range registers to support both general I/O decode and expansion board slot specific I/O decode. The BMIC also uses the AEN signal when decoding I/O locations.

The set of I/O Decode registers include two I/O Decode Range Base Address registers and two I/O Decode Range Control registers (refer to Section 8.2.6). The I/O Decode registers are used to define the address ranges of interest to the bus master expansion board. Each IOSEL#<1:0> pin has an associated Control and Base register along with an associated address range as defined by the I/O Decode register set.

Through the I/O Decode Range Control register set, the BMIC can be programmed to respond to a select I/O address range as either an 8-bit or 32-bit EISA device. The only control signal provided by the BMIC to the EISA bus during an I/O decode is the EX32# signal. The output state of the EX32# pin on the BMIC will indicate the elected response (low = 32-bit EISA, high = 8-bit EISA). The Control register set controls the size of the I/O decode range, the I/O decode type (slot specific or general I/O), and the I/O decode address latching. The I/O address can be latched by the CMD# signal (de-pipelined) or merely decoded. By latching the I/O address, the associated IOSEL# line will remain active a minimum of 5 ns from the rising edge of CMD#.

The IDOEs do not go active during an IOSEL cycle outside the shared register access space.

The I/O decode range size depends on the value of bits <4:0> in the Control register. Each of these bits masks a corresponding address comparison bit in the Base register. If no bits are masked in the Control register, the BMIC will decode a doubleword address. The bits are masked as follows:

I/O Control Register	I/O Base Register Bit Masked	EISA Address Bit Masked
Bit 0	Bit 0	IADS2
Bit 1	Bit 1	IADS3
Bit 2	Bit 2	IADS4
Bit 3	Bit 3	IADS5
Bit 4	Bit 4, 5	IADS<7:6>

The I/O Decode Range Base Address register contains the address range that is used during the I/O decode address comparison. The following table gives the bits in the I/O Base Address Register and the EISA Address that are used during the comparison:

I/O Base Address Register	(EISA Address Bits)	
	Slot Specific	General I/O
Bit 0	IADS2	IADS2
Bit 1	IADS3	IADS3
Bit 2	IADS4	IADS4
Bit 3	IADS5	IADS5
Bit 4	IADS6	IADS6
Bit 5	IADS7	IADS7
Bit 6	IADS10	IADS8
Bit 7	IADS11	IADS9

If bit 6 in the I/O Decode Range Control register is programmed for General I/O decode, and the two most significant bits in the I/O Decode Range Base Address register are programmed to 0 (IADS<9:8>), I/O decoding for that range will be disabled. This is done to ensure that the I/O address does not conflict with the slot specific address range or the EISA system board address range. The following table summarizes the EISA system I/O address mapping:

I/O Address Range (HEX)	I/O Range Reserved for
0000–00FF	EISA/ISA System Board
0100–03FF	General I/O (ISA Expansion Board)
0400–04FF	ISP (82357)
0500–07FF	General I/O (Alias of 0100h–03FFh)
0800–08FF	EISA System Board
0900–0BFF	General I/O (Alias of 0100h–03FFh)
0C00–0CFF	EISA System Board
0D00–0FFF	General I/O (Alias of 0100h–03FFh)
Slot Specific Range where X = Slot Number	
X000–X0FF	Slot (X)
X100–X3FF	General I/O (Alias of 0100h–03FFh)
X400–X4FF	Slot (X)
X500–X7FF	General I/O (Alias of 0100h–03FFh)
X800–X8FF	Slot (X)
X900–XBFF	General I/O (Alias of 0100h–03FFh)
XC00–XCFF	Slot (X) (BMIC Registers 0C80h–0CAFh)
XD00–XFFF	General I/O (Alias of 0100h–03FFh)

The following is an example of the BMIC programmed for slot specific decode:

I/O Decode Range 0 Control register programmed for (EFh)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
(EFh)	1	1	1	0	1	1	1	1

— Bit 7—Respond as a 32-bit EISA slave
— Bit 6—Slot specific decode enabled
— Bit 5—Slot specific address latched by CMD#
— Bit 4—Compare I/O Decode Range 0 Base Address Bits (5) and (4) with EISA address signals IADS7 and IADS6 respectively
— Bit 3—Mask I/O Decode Range 0 Base Address Bit (3)
— Bit 2—Mask I/O Decode Range 0 Base Address Bit (2)
— Bit 1—Mask I/O Decode Range 0 Base Address Bit (1)
— Bit 0—Mask I/O Decode Range 0 Base Address Bit (0)

I/O Decode Range 0 Base Address register programmed for (2-h)

	IADS11	IADS10	IADS7	IADS6	IADS5	IADS4	IADS3	IADS2
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
(2-h)	0	0	1	0	—	—	—	—

EISA slot specific address range decoded—X080h through X0BFh where X represents the expansion board slot number

```
      (X080h)                              (X0BFh)
xxxx000010(00)(00)00throughxxxx000010(11)(11)11
          ↑  ↑                              ↑  ↑
                EISA Address
                 Bits Masked
                 (IADS<5:2>)
```

Byte enables IBE<3:0># and EISA address lines IADS<1:0> are not used during either slot specific or general I/O decode. During slot specific I/O decode, EISA address lines IADS<9:8> must be 0 to ensure that the I/O address does not conflict with the ISA general I/O address range (0100h–03FFh).

IOSEL0# and EX32# will be driven low by the BMIC if addresses X080h through X0BFh are present on the EISA bus.

AEN is used as part of the decode and must be negated low when a response from the BMIC is required.

5.0 TRANSFER BUFFER INTERFACE

5.1 Transfer Buffer Interface Signals

The Transfer Buffer Interface portion of the BMIC provides the signals essential for interfacing to the expansion board as required for EISA-to-expansion board and expansion board-to-EISA burst data transfers. The Transfer Buffer Interface is designed to interface to a high speed transfer buffer and simple logic similar to that used in traditional DMA designs. This interface includes a 16-bit data bus, one clock input, and seven control signals.

The 16-bit data lines TDAT<15:0> are used by the BMIC to transfer the data to and from the transfer buffer logic on the expansion board during transfers. The data is word aligned. The BMIC automatically assembles the words received from the expansion board into 32-bit dwords for 32-bit transfers over the EISA bus. The data lines are also used by the BMIC to transport internally generated transfer start and real-time addresses to the external logic for use during data transfers (refer to Section 5.3).

The clock input (TCLK) controls the transfer rate between the BMIC and the external transfer buffer logic. The TCLK can be asynchronous to the BCLK.

The seven control signals include:

— Transfer Request (TRQ#): an output to request data transfers over the Transfer Buffer interface.
— Transfer Acknowledge (TACK#): An input to acknowledge data transfers. The TACK# signal may be used by the transfer buffer logic to add wait states to the data cycle.
— Data Transfer Direction (TDIR): An output to inform the external transfer buffer logic as to the direction of the current transfer (EISA read or EISA write).
— Transfer Channel Select (TCHAN): An output to indicate which of the two channels is currently active.
— Transfer Address Counter Load (TLD#): An output to load the current transfer start address to an external address counter, depending on the expansion board application.
— Transfer Data Output Enable (TDOE#): An input that unconditionally disables the BMIC from driving the TDAT<15:0> lines. With this signal, the BMIC can be prevented from driving the TDAT<15:0> lines while the local processor accesses the transfer buffer logic on the expansion board. No handshaking is required, so throughput is increased.
— Transfer End-of-Process (TEOP#): TEOP# is a status output pin that signals the end of a data transfer to the external transfer buffer logic.

NOTE:
Refer to Section 9.4 for additional information regarding the above signals.

5.2 External Transfer Buffer Logic

The Transfer Buffer interface is designed for high speed devices, such as SRAM based designs, or FIFOs. The Transfer Buffer interface data path is 16 bits wide. This requires the transfer clock (TCLK) to run at a speed of 16 MHz to 20 MHz to maintain the EISA maximum data rate of 33 Mbytes/sec. The fast cycle times required on the data Transfer Buffer interface can be implemented in the controlled environment found locally on the expansion board. If two BCLK transfers are used on the EISA side (16 Mbytes/sec), the timing requirements for the transfer buffer can be relaxed, and lower cost implementations can be utilized.

If the transfer buffer controller does dynamic arbitration for the transfer buffer between the BMIC and the peripheral device(s) on the expansion board, the peripheral device accesses should be short enough so that the BMIC's data FIFO can handle the interruption to its data flow without stalling the EISA transfer.

Examples of transfer buffer architecture implementations that could be interfaced to the BMIC include:

— A FIFO implementation which is large enough to buffer the difference in throughput rates between the peripheral device on the expansion board and the EISA Bus. See Section 5.2.1.

— A small high-speed DMA like device that generates addressing for a SRAM based transfer buffer.

— A controller implementation for dual-ported SRAM for high transfer buffer bandwidth.

— A page or nibble-mode dynamic-RAM controller implementation for large, low cost transfer buffers.

— For graphics systems, the frame buffer itself can be used for the transfer buffer with a non-linear address generator for transferring windows in the screen image.

5.2.1 FIFO IMPLEMENTATION

During EISA writes, the BMIC will overread the transfer buffer (read data beyond the number of bytes to be transferred) by a maximum of 28 bytes. These overread bytes may contain valid data (back to back transfers) which will be lost. The data loss can be avoided through software or hardware. The software solution avoids back to back transfers. This implies that there is data for only one transfer in the FIFO at any given time.

The hardware solution requires an external 22-bit Byte Counter and a Flip-Flop. The terminal count of the Byte Counter is used to SET the Flip-Flop which disables BMIC reads to the FIFO. The BMIC will continue to read (overread) "stale" data. The BMIC TEOP# output signal is used to RESET the Flip-Flop enabling BMIC reads to the external FIFO.

5.3 Transfer Interface Start Address Generation

The BMIC provides four 8-bit Transfer Buffer Interface (TBI) registers, two Base and two Current registers, which can be programmed with 16-bit transfer start addresses. Each transfer channel has an associated Base and Current register pair. The Base registers contain the start address and the Current registers provide the real-time address used to track the current transfer. The Current registers will increment by one each time a 16-bit word is transferred across the Transfer Buffer interface.

The 16-bit start address is transferred across the TDAT<15:0> lines to the transfer buffer logic at the beginning of all new data transfers (i.e., each time the TBI Base register set contents are transferred to the TBI Current register set). The contents of the TBI Base registers are transferred to the TBI Current registers after a write to the associated channel's Transfer Strobe register is completed (refer to Section 4.2). The BMIC provides a load signal (TLD#) which can be used to latch the start address into an external address counter for use by the transfer buffer logic.

The BMIC can also be programmed to generate the transfer address each time the associated channel regains the bus, in which case, the address will be the real-time address. By programming the CFGEA bit in the Channel Configuration register to a "1", the start address will be transferred to the transfer buffer logic at the beginning of all new transfers and the real-time address will be transferred each time the associated channel regains the bus. If the CFGEA bit is set to a "0", the transfer start address will be transferred at the beginning of all new transfers and the real-time address will not be transferred.

NOTE:
The TBI Current register set is readable by the local processor. However, there are possible coherency problems involved with reading multiple bytes while the current registers are being updated during a transfer. To avoid these problems, the channel's transfer should be temporarily suspended (using the channel's Configuration Register) before trying to read the channel's TBI Current register set.

5.4 Transfer Buffer Interface Timing Example

Figures 5-1 and 5-2 illustrate the start up and conclusion of a transfer cycle across the Transfer Buffer interface and should be used as a reference when reading the following text.

1. At the start of a data transfer TCHAN and TDIR change to their new values prior to the falling edge of TLD# to set up the cycle. TCHAN and TDIR will not change states as long as TRQ# is asserted.
2. TLD# is asserted until acknowledged by TACK#. The transfer address is transferred to the external logic each time the TBI Base register contents are transferred to the TBI Current register set (new transfer) and, if programmed, each time the current channel regains the bus.
3. The new address is loaded using the TDAT bus during TLD# at point (A). The TDAT bus should be turned on by asserting TDOE# during TLD# if the internal start address is required. Once the external channel address and direction are set up, the data transfer can begin.
4. Data transfer requests are signaled by TRQ# being asserted (low). TRQ# will remain active until the data transfer is completed or a transfer interruption occurs (refer to Section 1.0) followed by TACK# active. During an EISA write, there will be a one TCLK delay between TLD# deasserting and TRQ# asserting as denoted by point (D) in Figure 5-2. This is to allow time for the external buffers to change direction after the TLD# has been completed.
5. Each word transfer to or from the BMIC is acknowledged by the TACK# signal. If TACK# is active at the rising edge of TCLK, one word will be transferred. If TACK# is not active at the rising edge of TCLK, the word that is currently being transferred will be inhibited and a wait state will

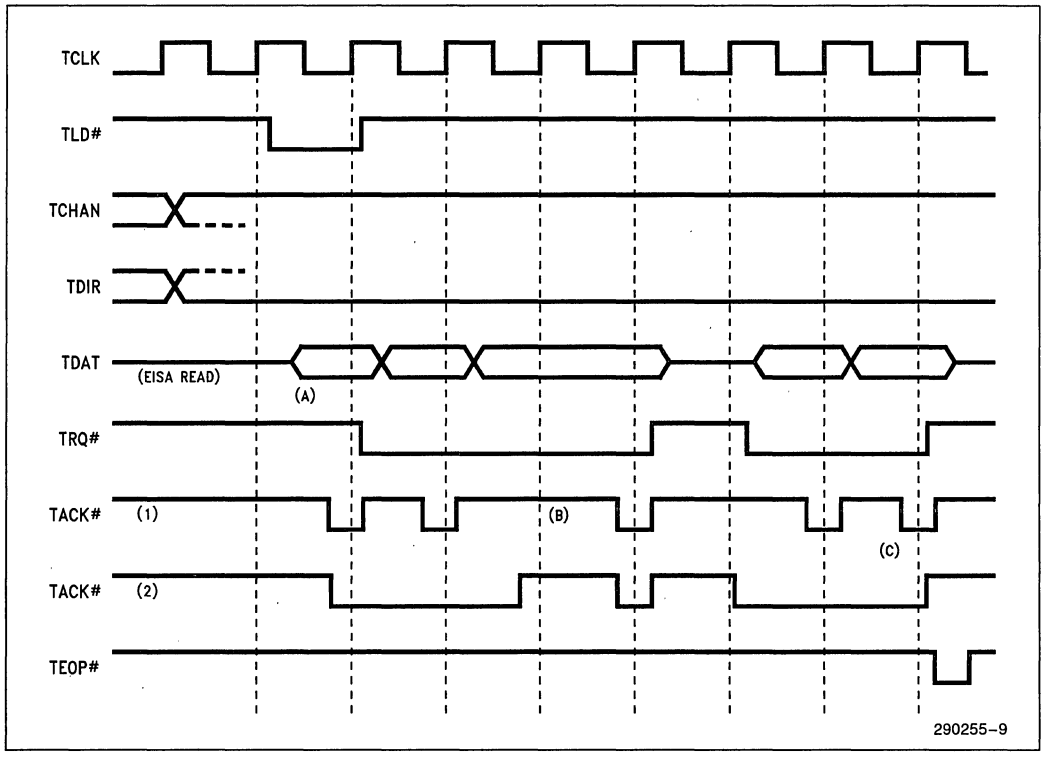


Figure 5-1. Transfer Buffer Interface Timing (EISA READ)

be inserted. This is shown at point (B) in Figure 5-1. Such a wait may be needed when the external transfer buffer logic is arbitrating between the BMIC and the I/O subsystem on the expansion board. Wait states may also be inserted by stretching TCLK at point (C) in both of the figures. Clock stretching is possible as long as the one to one ratio of TCLK to BCLK is not violated.

NOTE:

A long TCLK stretch time will hang the Transfer Buffer interface. Also, TCLK must be running during the time TRQ# is inactive in order for the Transfer Buffer interface to function properly.

As indicated above, TACK# must be stable at the rising edge of TCLK. However, TACK# can assume any convenient pattern at other times. As shown by the first pattern, TACK# (1) pulses low at the TCLK edge that data is transferred. This pattern is particularly useful when TCLK wait-states are desired as indicated at point (B) in Figure 5-1. The alternate pattern (TACK#2) is useful during TCLK stretching since TACK# is always low during TRQ# as shown at point (C). This is effective since the transfer clock edge timing is controlled by the amount TCLK is stretched.

6. TEOP# is asserted at the end of a transfer by the BMIC.

   The BMIC will indicate end-of-process by asserting TEOP# shortly after the negation of the last CMD# in the transfer. During an EISA write transfer, the BMIC will assert TEOP# a maximum of two TCLKs after CMD# is negated. During an EISA read transfer, the BMIC will assert TEOP# typically eight TCLKs after the negation of CMD#. In either case (EISA read or EISA write), the TEOP# signal is delayed from the rising edge of TCLK.

NOTE:

The BMIC will assert the expansion board interrupt signal (LINT) at the end of a transfer, if so programmed in the Transfer Channel Configuration register.

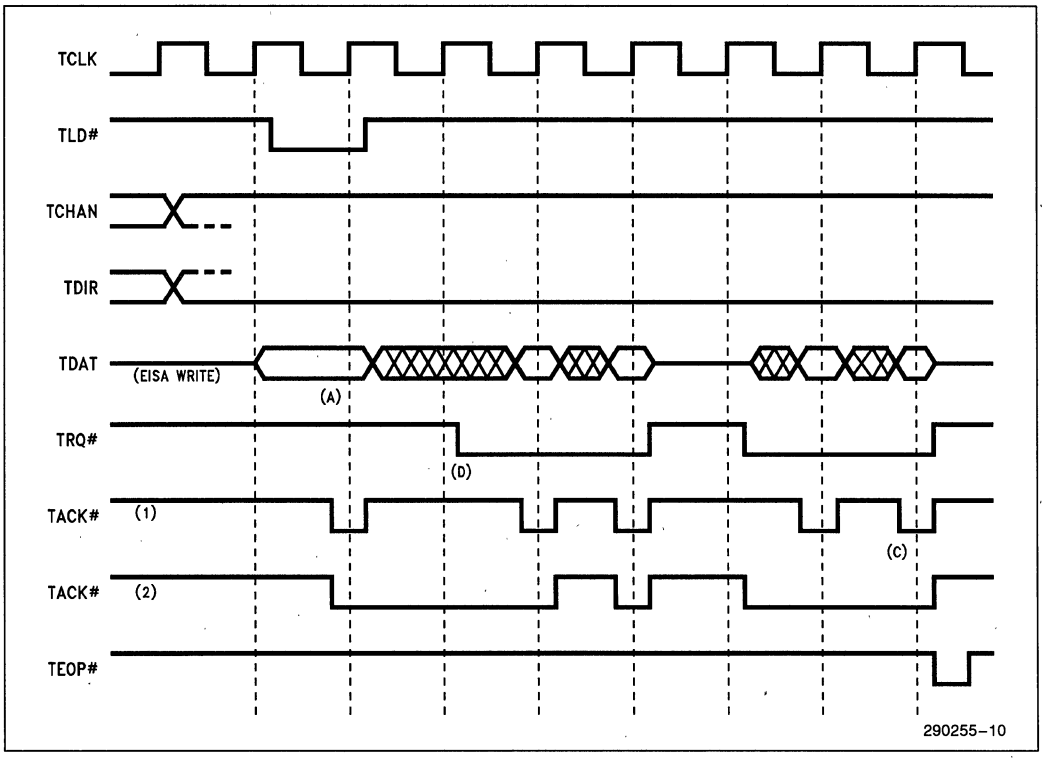


Figure 5-2. Transfer Buffer Interface Timing (EISA WRITE)

6.0 FIFO/DATA ALIGNER

6.1 FIFO/Data Aligner

The BMIC uses two identical FIFOs, one per transfer channel, and a common data aligner for data transfers between system memory and the bus master expansion board. The primary function of the FIFO/Data Aligner Unit is to help isolate and simplify the timing relationships between the EISA bus and the devices on the expansion board.

The FIFO allows the timing on the expansion Board side of the BMIC to be based on a locally generated clock signal. This transfer clock (TCLK) can be independent of the EISA BCLK signal that governs EISA bus timing. The FIFO also provides latency protection for wait states generated on either the EISA bus or expansion board.

The Data Aligner arranges the 16-bit data from the external transfer buffer to any arbitrary byte alignment in system memory. The data aligner also performs the assembly and disassembly of the EISA data during the transfer. The TDAT data assembly and disassembly is done by the Transfer Buffer interface portion of the BMIC.

6.2 FIFOs

Each FIFO on-board the BMIC is 24 bytes in size. The transfer data is written into the FIFOs from either the expansion board or the EISA bus side, depending on the direction of transfer. The data is written into the FIFO as doublewords during the transfer. However, if the data is not doubleword aligned, partial FIFO loads will be done at the beginning or end of a transfer depending on the byte count, address programmed and the direction of the transfer.

The condition of the FIFOs can be determined by a read to the Transfer Channel Status register set. A read to this register will indicate whether the FIFOs are stalled or active. A FIFO stall is defined as a FIFO that is full during an EISA read or empty during an EISA write. In either case, the transfer buffer logic is unable to keep up with the EISA device. If a FIFO stall occurs, the transfer will be stopped and the BMIC will either service the transfer request with the highest priority or relinquish the EISA bus to the system. The BMIC will relinquish the bus to the system if the CFGFF bit in the channel's corresponding Configuration register is set to a 1.

6.3 Data Aligner

6.3.1 EISA BYTE ALIGNMENT

The BMIC automatically handles the byte alignment for the EISA bus in the case of misaligned doubleword boundaries and assumes no performance penalty. The BMIC will do any partial doubleword transfers as required at the beginning and the end of all transfers. The two least significant bits of the 32-bit transfer start address (A1 and A0) are used to provide the byte alignment for both EISA read and EISA write transfers. The following tables illustate the BMIC's byte alignment approach during 32- and 16-bit transfers:

In the following tables "—" represents no data transferred and the digits represent the data items being transferred. The byte alignment for an EISA read is identical to that of an EISA write.

EISA Write (32-bit/12-byte Transfer) and (16-bit/6-byte Transfer)

		(32-Bit)				(16-Bit)			
A1	**A0**	**Output Data to EISA Bus**				**Output Data to EISA Bus**			
Byte Lane	**→**	**3**	**2**	**1**	**0**	**3**	**2**	**1**	**0**
0	0	03	02	01	00	—	—	01	00
		07	06	05	04	—	—	03	02
		11	10	09	08	—	—	05	04
0	1	02	01	00	—	—	—	00	—
		06	05	04	03	—	—	02	01
		10	09	08	07	—	—	04	03
		—	—	—	11	—	—	—	05
1	0	01	00	—	—	—	—	01	00
		05	04	03	02	—	—	03	02
		09	08	07	06	—	—	05	04
		—	—	11	10	—	—	—	—
1	1	00	—	—	—	—	—	00	—
		04	03	02	01	—	—	02	01
		08	07	06	05	—	—	04	03
		—	11	10	09	—	—	—	05

6.3.2 DATA ASSEMBLY/DISASSEMBLY

Before being placed on either the TDAT or IDAT data buses during an EISA read or EISA write, the data will be assembled or disassembled as required. The IDAT data is assembled and disassembled by the FIFO/data aligner portion of the BMIC and the TDAT data is assembled and disassembled by the Transfer Buffer interface portion of the BMIC. The following paragraphs illustrate the BMIC's assembly and disassembly approach during 32- and 16-bit transfers. The illustration assumes that byte alignment is not required.

During 32-bit EISA read transfers, the 32-bit doublewords are removed from the EISA bus and placed into the FIFO. After flowing through the FIFO, the 32-bit doublewords are copied-down to 16-bit words and then placed on the TDAT bus.

During 32-bit EISA write transfers, the 16-bit words are removed from the TDAT lines, assembled into 32-bit doublewords, and then placed into the FIFO. After flowing through the FIFO, the 32-bit data is placed on the EISA bus. No further assembly or disassembly is required after the FIFO as the data is already in 32-bit doubleword form.

During 16-bit EISA read burst transfers, the 16-bit words are removed from the EISA bus, assembled into 32-bit doublewords, and then placed into the FIFO. After flowing through the FIFO, the 32-bit data is copied-down to 16-bit words and then placed on the TDAT bus.

During 16-bit EISA write burst transfers, the 16-bit words are removed from the TDAT bus, assembled into 32-bit doublewords, and then placed into the FIFO. After flowing through the FIFO, the 32-bit data is copied-down to 16-bit words and then placed on the EISA bus.

7.0 LOCAL PROCESSOR INTERFACE

The BMIC's Local Processor interface is based on an asynchronous, 8-bit interface. All of the slave signals required for a local processor to program the BMIC are provided through this interface. These signals include (LCS#, LRD#, LWR#); two address lines (LADS0 and LADS1) for addressing internal registers; an 8-bit data path (LDAT); an interrupt signal (LINT); and a ready signal (LRDY). LINT allows the BMIC to interrupt the local processor and the ready signal (LRDY) indicates when valid data is available on the LDAT lines (shared register accesses only, see below). If a local processor is not used, the Local Processor interface can be connected to the 8-bit ISA bus (refer to Section 7.3). The choice of the local microprocessor or microcontroller used depends upon the specific application and the degree of performance and data processing needed (refer to Section 7.2).

The Local Processor interface portion of the BMIC contains two 8-bit registers which are used by the local processor to access all of the BMIC's internal registers. These registers are mapped into the Local Processor interface and include a Local Data register and a Local Index register. These registers are selected using the Local Processor interface's two address lines. The Local Status/Control register is also directly mapped into the Local Processor interface and is used to provide the local processor with the interrupt, peek/poke, and Base register status.

The BMIC allows the local processor and the EISA bus to communicate with each other through a set of Command/Status registers. The Command/Status registers are referred to as shared registers and include a set of Mailbox registers, Semaphore ports, and doorbell registers. The mailbox registers are used to pass messages to and from the local processor and the EISA bus and are controlled by the Semaphore ports. The Doorbell register set is used to inform the respective processor of new messages. Also part of the shared register set are the ID registers, which are used to support the EISA expansion board ID function.

The BMIC allows the local processor access to individual locations in system memory or I/O space using the Peek/Poke feature. The local processor can also initiate BMIC burst and non-burst (two BCLK) data transfers to and from system memory.

7.1 Shared Registers—Status/ Command Support

As data transfer rates increase, it is critical that an efficient command and status passing mechanism be implemented so that command and status exchange does not become a new bottleneck to system performance. The BMIC utilizes a high-performance command/status interface between the main system and the local processor to minimize command/status overhead.

The Shared registers are a group of registers accessable by the system CPU or EISA bus master and the local processor for general-purpose command and status interactions and EISA expansion board ID function support. The features of the BMIC command/status support include a pair of semaphore ports, a set of interrupt ports ("doorbell registers"), and a set of mailbox registers. With these functions, many different types of high-performance communication protocols can be defined between the system and the expansion board. The Global

Configuration register, the System Interrupt Enable/Control register, and the ID registers are also part of the shared register set.

7.1.1 SEMAPHORE PORTS

The two semaphore ports are specifically designed to allow set-and-test functions in I/O space. Specifically, the ports are used to lock access to the mailbox registers and to lock access to links in main memory. Each of the semaphore ports consists of two parts: the semaphore flag bit and the semaphore test bit.

When a write occurs to the semaphore flag bit through either the EISA interface or the Local Processor interface, the old value of the semaphore flag bit is copied to the appropriate semaphore test bit. The old value of the semaphore flag bit is then available in the test bit to be read back by the processor. If the value read back from the semaphore test bit is a "1", the requested resource is unavailable for use. If the value read back is a "0", the requested resource is available for use and is now locked by the requesting processor or bus master. In this manner, set-and-test algorithms can be implemented without using the EISA bus lock function. The processor or EISA bus master unlocks the semaphore by simply writing a "0" to the associated semaphore flag bit.

NOTE:
The Semaphore ports and resources are locking only in a software sense, as in any semaphore in main memory. The Semaphore ports are identical and are not associated with either interface (EISA or Local). The protocol for the semaphores and the effect they have on other shared registers, like the Mailbox registers, is strictly a matter of how the system software chooses to use them.

Implementing the semaphore in the BMIC instead of main memory eliminates the need for the BMIC to arbitrate for the EISA bus every time it wishes to update or test the semaphore. Note that the semaphore scheme described here is functional only when a single device on the EISA is communicating with the BMIC; the semaphore coordinates "locks" between the single device and the local processor. In the case that multiple masters attempt to lock access to the BMIC, the masters must first agree amongst themselves which one has the privilege to use the BMIC semaphore port(s).

7.1.2 MAILBOX REGISTERS

A set of 16 8-bit general-purpose mailbox registers are used to pass information between the bus master expansion board and the EISA system. The 16 registers are mapped contiguously in EISA slot-specific I/O space, so they can be accessed as bytes, words, or doublewords. These registers can be used to directly pass command and status information, or they can be used as pointers to larger command blocks in memory.

The mailbox registers can be read or written at any time from either the EISA bus or the Local Processor interface. An internal arbitration is implemented in such a way that if there is a simultaneous read and write from both sides of a mailbox register, then the read operation will not contain indeterminate bits. In other words, when a read operation is done on a mailbox register at the same time as a write operation to that register, the bit pattern that is read will be either the old bit pattern in the mailbox, or the new bit pattern being written, but never some transitory, invalid bit pattern.

7.1.3 DOORBELL REGISTERS

There are two 8-bit doorbell Interrupt/Status registers in the BMIC, one assigned to the EISA side and one assigned to the expansion board side. The EISA System Doorbell register is used by the local processor to request service from the EISA side and the Local Doorbell register is used by the device on the EISA side to send an interrupt request to the local processor on the bus master expansion board. The doorbell Interrupt/Status registers are implemented with "sticky" bits, so that individual bits in the register can be set by the interrupting device or reset by the servicing device without knowledge of the states of the other bits in the register. The eight bits in each doorbell register allow up to eight separate devices or events in each direction to have interrupt requests pending simultaneously. The interrupt requests pending in either of the two Doorbell registers are ORed with the other interrupt sources from within the BMIC, and the result is sent out over one of the two interrupt pins: LINT or EINT.

Each doorbell register has an associated 8-bit Interrupt Enable register used to enable or disable the interrupts on an individual basis. The BMIC also includes a System Interrupt. Enable/Control register and a Local Status/Control register used to disable the system (EINT) and local (LINT) interrupts and to verify the status of the system and local interrupts on a global basis (refer to Sections 8.1.1.3.3 and 8.2.2).

The following paragraphs describe the operation of the Local Doorbell Interrupt/Status register. The EISA System Doorbell Interrupt/Status register is similar, but operates in the opposite direction.

Each device or event that can interrupt the bus master expansion board can be assigned a bit position within the BMIC's Local Interrupt/Status Doorbell

register. When the device on the EISA bus wants to send an interrupt request to the bus master expansion board, it writes to the Local Interrupt/Status Doorbell register (from the EISA side) with that device's assigned bit position set active. This will set that bit in the Local Interrupt/Status Doorbell register, but leave the other bits in the register unaffected. If that bit position is not disabled, then the interrupt signal to the local processor will be asserted.

When the local processor services the interrupt, it checks the Local Status/Control Register to determine the source of the interrupt. If the control register indicates that the Local Doorbell register is one of the active interrupt sources, then the local processor can read the Local Doorbell register to determine which bits are active and requesting interrupts. If the local processor decides to service one of the requests from the Local Doorbell register, it can write to the Local Doorbell register with that bit's position set. This action will cause that bit in the Local Doorbell register to reset, but the other bits will remain unaffected. Thus, each bit in the Local Doorbell register is like a set-reset flip-flop, with the EISA bus controlling the "set" input, and the Local Processor interface controlling the "reset" input.

7.2 Local Processor Recommendations

The Local Processor interface to the BMIC will support numerous processors, from the 8088 microprocessor to the 376 embedded processor.

The 80186, 80C186, 80188, and 80C188 family of processors provides a clean interface to the BMIC's Local Processor interface and eliminates the need for additional logic. An on-board programmable wait-state generator eliminates the need for external wait-state generation logic between the processor and the BMIC during non-shared register accesses.

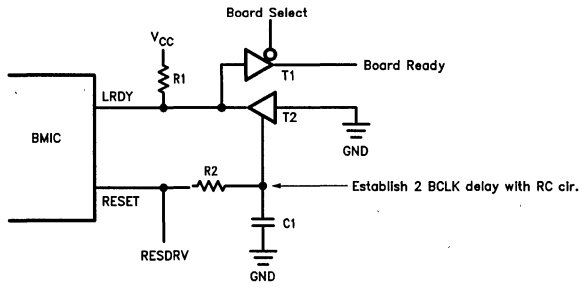
7.3 Requirements for No Local Processor

The BMIC allows for expansion board designs that do not require a local processor. To support the programming of the BMIC in a no local processor board design, the Local Processor interface must be connected to the ISA bus. However, when the ISA bus is used, the BMIC must be informed that there is no local processor and that it must change its function slightly (refer to next section). To inform the BMIC that no local processor is present, LRDY must be driven low during RESET and remain low a minimum of two BCLKs after RESET is negated.

The following circuit can be used to establish the proper LRDY/RESET timing as required for a no local processor design (see Figure 7-1).

7.4 EISA ID Function Support/ Registers

The BMIC provides support for the EISA expansion board ID function. The primary ID implementation takes advantage of the local processor. Upon reset, the local processor executes a routine from its ROM that writes the product identifier for the expansion board to the four 8-bit ID registers in the BMIC. The registers are accessed through the Local Processor interface and are located at local index addresses 00h–03h. On the EISA side, these registers are mapped into the EISA slot specific ID address range XC80h–XC83h.



R1 = 8.2K
R2 = Depends on frequency of BCLK and application implemented.
C1 = Depends on frequency of BCLK and application implemented.
T1 = Active low tri-state buffer
T2 = Active high tri-state buffer

Figure 7-1. LRDY/RESET Circuit with No Local Processor

If the host CPU accesses the ID registers in the BMIC before the local processor has programmed them, the BMIC will return the setup delay ID code 0111XXXXh in the byte 0 ID register located at EISA slot specific I/O address XC80h. The byte 0 ID register should be programmed last by the local processor.

If a local processor is not used, external registers will have to be implemented on the expansion board to hold the expansion board ID value. The BMIC will automatically set its I/O Decode Range 0 Control register to decode 8-bit EISA ID addresses. The IOSEL0# output signal can then be used to trigger external logic on the expansion board to enable ID data onto the IDAT<7:0> data lines. The ID register must be connected as shown in Figure 7-2. The external logic should monitor SA1 and SA0 on the ISA bus to determine which data byte to drive.

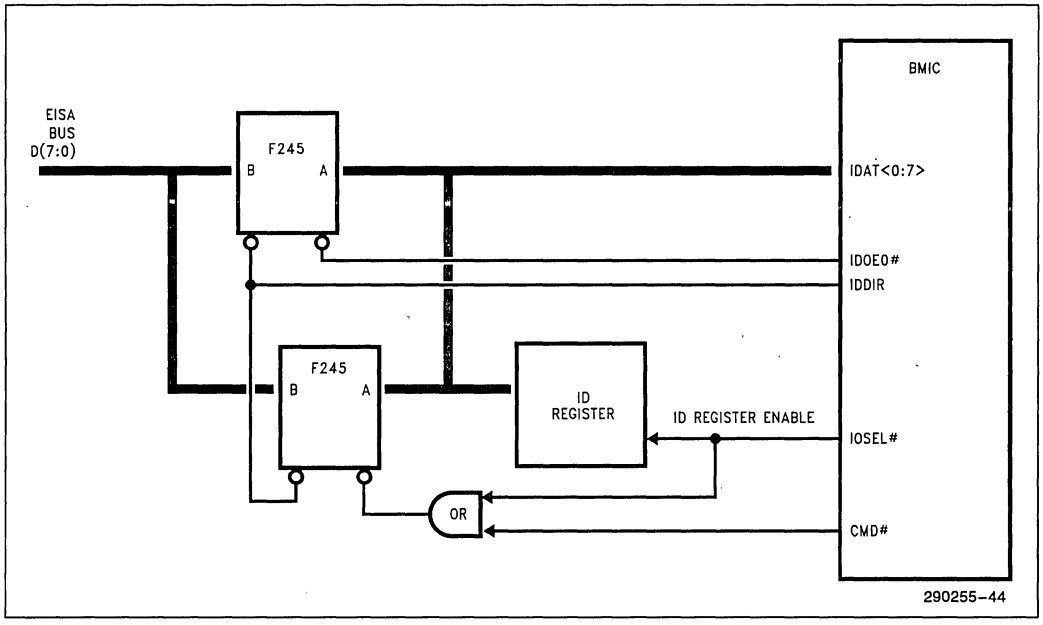


Figure 7-2. IDOE0# Connection during ID Register Access

8.0 REGISTER DESCRIPTION

8.1 Shared Register Description

The following is a table of the Shared register group listing the number of registers, register type (read/write) as related to the local and EISA side, register name, and register size:

Number	EISA	Local Type	Register Name Type	Active Bits per Register
2	R/W	R/W	Semaphore Register	2 Bits
16	R/W	R/W	Mailbox Register	8 Bits
1	R/W	R/W	Local Doorbell Interrupt/Status Register	8 Bits
1	R	R/W	Local Doorbell Enable Register	8 Bits
1	R/W	R/W	EISA System Doorbell Interrupt/Status Register	8 Bits
1	R/W	R	EISA System Doorbell Enable Register	8 Bits
1	R/W	R	System Interrupt Enable/Control Register	8 Bits
1	R	R/W	Global Configuration Register	8 Bits
4	R	R/W	ID Register	8 Bits

8.1.1 COMMAND/STATUS SUPPORT REGISTERS

8.1.1.1 Semaphore Ports (Read/Write)

The BMIC contains two Semaphore ports which can be used to software lock resources between the EISA bus and the local processor. Each semaphore port controls a 1-bit semaphore flag. Upon reset, the Semaphore ports are reset to 0.

Semaphore Port 0 EISA Address—XC8Ah
Semaphore Port 0 Local Index Address—0Ah

Semaphore Port 1 EISA Address—XC8Bh
Semaphore Port 1 Local Index Address—0Bh

—	—	—	—	—	—	Bit 1	Bit 0

Bit 7–2 —Reserved, set to 0

Bit 1 —Semaphore Test bit (Read Only)

Bit 0 —Semaphore Flag bit (Read/Write)

Bit (0) reflects the actual value of the semaphore at any given instant. Whenever a write is done to the Semaphore Flag bit (0), its previous value is simultaneously copied to the Semaphore test bit (1). Internal to the BMIC, there are two test bits for each semaphore port: one for the EISA interface and one for the Local Processor interface. To do a test-and-set function, write to the semaphore port with the desired semaphore value in the flag bit. After a write has been completed, read the semaphore port and check the test bit to verify that a collision did not occur.

8.1.1.2 Mailbox Registers (Read/Write)

The mailbox registers are sixteen 8-bit, general purpose registers. The format of the contents of the mailbox registers is user-defined. The Mailbox register set is not initialized to a fixed value upon reset.

EISA Address—XC90h through XC9Fh
Local Index Address—10h through 1Fh

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

8.1.1.3 Doorbell Registers

8.1.1.3.1 Local Doorbell Interrupt/Status Register (Read/Write)

This register is implemented with "sticky" bits (refer to Section 7.1.3). The Local Doorbell Interrupt/Status register is used by the EISA bus to send an interrupt request to the expansion board. When read from, this register indicates the status of pending interrupt events. Upon reset, the Doorbell Interrupt/Status register is reset to 0.

EISA Address—XC8Dh
Local Index Address—0Dh

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bit 7–0 1 = Doorbell interrupt pending (local CPU read)

Set Doorbell bit (EISA write)

Reset Doorbell bit (Local CPU write)

0 = No doorbell interrupt pending (Local CPU read)

No action (EISA or local CPU write)

Bits 0–7 allow up to eight events or devices on the EISA side to interrupt the local side of the BMIC. The above bits can only be reset by the servicing processor on the local side.

8.1.1.3.2 Local Doorbell Enable Register (Read/Write)

The Local Doorbell Enable register is used by the local processor to enable or disable interrupt requests to the local expansion board. This register is read only from the EISA side. Upon reset, the Doorbell Enable register is set to 0.

EISA Address—XC8Ch
Local Index Address—0Ch

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bit 7–0 1 = Enable doorbell interrupt for corresponding bit position

0 = Disable doorbell interrupt for corresponding bit position

No action (local CPU write)

Bits 0 through 7 act as interrupt enables for bits 0 through 7 in the Local Doorbell Interrupt/Status register respectively.

8.1.1.3.3 EISA System Doorbell Interrupt/Status Register (Read/Write)

This register is implemented with "sticky" bits (refer to Section 7.1.3). The EISA System Doorbell Interrupt/Status register is used by the expansion board to send an interrupt request to the EISA bus. When read from, this register indicates the status of pending interrupt events. Upon reset, the EISA System Doorbell Interrupt/Status register is reset to 0.

EISA Address—XC8Fh
Local Index Address—0Fh

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bit 7–0 1 = Doorbell interrupt pending (EISA read)
Set Doorbell bit (Local CPU write)
Reset Doorbell bit (EISA write)
0 = No doorbell interrupt pending

Bits 7–0 allow up to eight events or devices on the expansion board to send interrupts to the EISA bus. The above bits can only be reset by the servicing processor on the EISA side.

8.1.1.3.4 EISA System Doorbell Enable Register (Read/Write)

The EISA System Doorbell Enable register is used by the EISA processor to enable or disable interrupt requests to the EISA side. This register is read only from the local side. Upon reset, the EISA System Doorbell Enable register is reset to 0.

EISA Address—XC8Eh
Local Index Address—0Eh

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bit 7–0 1 = Enable doorbell interrupt for corresponding bit position
0 = Disable doorbell interrupt for corresponding bit position

Bits 0 through 7 act as interrupt enables for bits 0 through 7 in the EISA System Doorbell Interrupt/Status register respectively.

8.1.1.3.5 System Interrupt Enable/Control Register (Read/Write)

This register is used by the processor on the EISA side to disable the EINT signal. The EISA processor also can read this register to determine whether there are any pending interrupt requests in the EISA System Doorbell Interrupt/Status register. This register is read only from the local side. Upon reset, this register is reset to 0.

EISA Address—XC89h
Local Index Address—09h

—	—	—	—	—	—	Bit 1	Bit 0

Bit 7–2 — Reserved, set to 0
Bit 1 — (read-only bit)
1 = Enabled interrupts are pending in EISA System Doorbell Interrupt/Status register
0 = No enabled interrupts are pending in EISA System Doorbell Interrupt/Status register
Bit 0 —
1 = Enable interrupts from System Doorbell register (EISA write)
0 = Disable interrupts from System Doorbell register (EISA write)

8.1.2 GLOBAL CONFIGURATION REGISTER (READ/WRITE)

This register is used to program the type of protocol, edge or level-triggered, that will be used with the EINT and LINT interrupt signals. The Global Configuration register is also used to program the preempt timer and provide four bits for a BMIC hardware revision number. This register is read only from the EISA side. Upon reset, bits 0–3 are reset to 0.

EISA Address—XC88h
Local Index Address—08h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bits 7–4 (read-only)
Hardware revision number of the BMIC

Bit 3
1 = System interrupt pin (EINT) uses edge-triggered protocol (Active high)
0 = System interrupt pin (EINT) uses level-triggered protocol (Active low open collector)

Bit 2
1 = Local interrupt pin (LINT) is set for active high operation
0 = Local interrupt pin (LINT) is set for active low operation

Bits 1, 0 Delay to give up bus after preempt
00 = 3 BCLKs
01 = 32 BCLKs
10 = 64 BCLKs
11 = reserved

EISA Address—XC80h through XC83h (bytes 0–3)
Local Index Address—0h through 3h (bytes 0–3)

8.1.3 ID REGISTERS

The ID register set consists of four 8-bit registers. These registers are programmed at initialization time with the product identifier for the expansion board which contains the BMIC. The registers are mapped as read-only into the EISA ID I/O address range. Upon reset, the ID byte 0 register will contain the value 0111XXXX, which is the EISA ID delay value. The local processor should program byte 0 last. If the external ID support scheme is selected, then these registers are disabled. The bit definitions defined below have significance for the EISA ID protocol but not for any BMIC hardware functionality. Upon reset, ID bytes 1–3 are not initialized to a fixed value.

ID Register Bytes 0–3:

	7	6	5	4	3	2	1	0
Byte 0	—	MCC14	MCC13	MCC12	MCC11	MCC10	MCC24	MCC23
Byte 1	MCC22	MCC21	MCC20	MCC34	MCC33	MCC32	MCC31	MCC30
Byte 2	MCC43	MCC42	MCC41	MCC40	MCC53	MCC52	MCC51	MCC50
Byte 3	MCC63	MCC62	MCC61	MCC60	MCC73	MCC72	MCC71	MCC70

ID Register Byte 0:

Bit 7	—	Reserved
Bits 6–2	MCC1<4:0>	First character of manufacturer's code
Bits 1, 0	MCC2<4:3>	First portion of second character of manufacturer's code

ID Register Byte 1:

Bits 7–5	MCC2<2:0>	Second portion of second character of manufacturer's code
Bits 4–0	MCC3<4:0>	Third character of manufacturer's code

ID Register Byte 2:

Bits 7–4	MCC4<3:0>	First hex digit of product number
Bits 3–0	MCC5<3:0>	Second hex digit of product number

ID Register Byte 3:

Bits 7–4	MCC6<3:0>	Third hex digit of product number
Bits 4–0	MCC7<3:0>	Hexadecimal digit of product revision

8.2 Local Processor Only Registers

The following is a table of the Local Processor Only register group listing the number of registers, register type (read/write) as related to the local side, register name, and register size:

Number	Local Type	Register Name	Active Bits per Register
INDEX REGISTERS			
1	R/W	Local Index Register	8 Bits
1	R/W	Local Data Register	8 Bits
1	R/W	Local Status/Control Register	8 Bits
DATA CHANNEL TRANSFER REGISTERS			
4	R/W	Data Transfer Channel 0 Base Address Register	8 Bits
4	R/W	Data Transfer Channel 1 Base Address Register	8 Bits
4	R	Data Transfer Channel 0 Current Address Register	8 Bits
4	R	Data Transfer Channel 1 Current Address Register	8 Bits
3	R/W	Data Transfer Channel 0 Base Count Register	8 Bits
3	R/W	Data Transfer Channel 1 Base Count Register	8 Bits
3	R	Data Transfer Channel 0 Current Count Register	8 Bits
3	R	Data Transfer Channel 1 Current Count Register	8 Bits
DATA TRANSFER CONTROL/STATUS REGISTERS			
1	W	Channel 0 Transfer Strobe Register	0
1	W	Channel 1 Transfer Strobe Register	0
1	R/W	Channel 0 Configuration Register	8 Bits
1	R/W	Channel 1 Configuration Register	8 Bits
1	R/W	Channel 0 Status Register	6 Bits
1	R/W	Channel 1 Status Register	6 Bits
PEEK/POKE REGISTER			
4	R/W	Peek/Poke Address Register	8 Bits
4	R/W	Peek/Poke Data Register	8 Bits
1	R/W	Peek/Poke Control Register	8 Bits
I/O DECODE REGISTERS			
1	R/W	I/O Decode Range 0 Base Address Register	8 Bits
1	R/W	I/O Decode Range 1 Base Address Register	8 Bits
1	R/W	I/O Decode Range 0 Control Register	8 Bits
1	R/W	I/O Decode Range 1 Control Register	8 Bits
TRANSFER BUFFER INTERFACE (TBI) REGISTERS			
2	R/W	TBI Channel 0 Base Address Register	8 Bits
2	R/W	TBI Channel 1 Base Address Register	8 Bits
2	R	TBI Channel 0 Current Address Register	8 Bits
2	R	TBI Channel 1 Current Address Register	8 Bits

8.2.1 INDEX REGISTERS

The BMIC's register set is accessed using the local Index and Local Data register set (refer to Section 3.2.6.1). The Local Index and Local Data registers are mapped directly into the Local Processor interface of the BMIC.

8.2.1.1 Local Index Register (Read/Write)

The Local Index register contains the address of the BMIC register that is currently being accessed. An optional auto-increment mode is supported through this register, which automatically increments the index register after each Local Data register read or write. Upon reset, the Local Index register is set to 0.

Local Address—1h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bit 7 — 1 = Autoincrement local index register after access to local data register

0 = Do not autoincrement

Bits 6–0 — Local index address

8.2.1.2 Local Data Register (Read/Write)

During a BMIC local register access, the value of the register being accessed is passed through this register.

Local Address—0h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

8.2.2 LOCAL STATUS/CONTROL REGISTER (READ/WRITE)

The Local Status/Control register is directly mapped into the Local Processor interface and is accessible using the two address lines (LADS<1:0>). This register provides current local doorbell interrupt status, current Channel 0 and Channel 1 interrupt and Base register status, and current peek/poke cycle status. This register is also used by the local processor on the expansion board to disable and provide the current status of the LINT signal (active or inactive). Bit 4 in this register is read/write and the remaining bits are read only. Upon reset, the Local Status/Control register is reset to 0.

Local Address—2h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bit 7 — R
- 1 = Enabled interrupts are pending in Local Doorbell register
- 0 = No enabled interrupts are pending in Local Doorbell register

Bit 6 — R
- 1 = Enabled interrupts are pending from channel 1 events
- 0 = No enabled interrupts are pending from channel 1 events

Bit 5 — R
- 1 = Enabled interrupts are pending from channel 0 events
- 0 = No enabled interrupts are pending from channel 0 events

Bit 4 — R/W
- 1 = Local interrupts enabled
- 0 = All local interrupts disabled

Bit 3 — R
- 1 = Local interrupt signal (LINT) is currently active
- 0 = LINT signal is currently inactive

Bit 2 — R
- 1 = Most recent peek/poke command is still pending
- 0 = Most recent peek/poke command is complete

Bit 1 — R
- 1 = Base register set for channel 1 is busy
- 0 = Base register set channel 1 is available

Bit 0 — R
- 1 = Base register set for channel 0 is busy
- 0 = Base register set for channel 0 is available

8.2.3 DATA CHANNEL TRANSFER REGISTERS

The Data Channel Transfer register set is used to control burst and standard EISA data transfers. Each transfer channel has a set of Base and Current registers, and also a Transfer Strobe, Configuration, and Status register.

NOTE:
The Base register set and the Transfer Strobe register must be initialized before a transfer can take place. They are not initialized to a fixed value upon reset.

8.2.3.1 Channel 0 and 1 Transfer Base Address Registers (Read/Write)

Each Channel has an associated Base Address register set. The Transfer Base Address registers are programmed with the 32-bit starting address to be used during the data transfer. After the Base registers have been programmed, they should not be programmed again until the contents of the Base registers have been transferred to the Current registers. The Base Address registers are not initialized to a fixed value upon reset.

Channel 0 Local Index Address—43h through 46h (bytes 0 through 3)
Channel 1 Local Index Address—63h through 66h (bytes 0 through 3)

BYTE 3	BYTE 2	BYTE 1	BYTE 0

8.2.3.2 Channel 0 and 1 Transfer Current Address Registers (Read Only)

Each Channel has an associated Current Address register set. The Transfer Current Address registers contain the real-time status of the 32-bit transfer address. The Current Address registers are not initialized to a fixed value upon reset.

NOTE:
The current register set is readable by the local processor. However, there are possible coherency problems involved with reading multiple bytes while the current registers are being updated during a transfer. To avoid these problems, a channel's transfer should be temporarily suspended (using the channel's Configuration Register) before trying to read the channel's current register set.

Channel 0 Local Index Address—53h through 56h (bytes 0 through 3)
Channel 1 Local Index Address—73h through 76h (bytes 0 through 3)

BYTE 3	BYTE 2	BYTE 1	BYTE 0

8.2.3.3 Channel 0 and 1 Transfer Base Count Registers (Read/Write)

Each Channel has an associated Base Count register set. The Transfer Base Count registers are programmed with the number of bytes to be transferred. Each Channel has 22 bits of counter space for a maximum transfer block size of 4 Mbytes. Bits 22 and 23 are used for channel control. The Base Count registers are not initialized to a fixed value upon reset.

Channel 0 Local Address—40h through 42h (bytes 0 through 2)
Channel 1 Local Address—60h through 62h (bytes 0 through 2)

BYTE 2			BYTE 1	BYTE 0
Bit 23	Bit 22	Bit 16–21	Bits 8–15	Bits 0–7

Bit 23 — R/W
- 1 = Start transfer as soon as base register set is copied to current register set
- 0 = Hold transfer after current register set is loaded. Wait for transfer suspend bit 0 to be reset

Bit 22 — W
- 1 = Transfer from bus master expansion board to EISA bus (EISA write)
- 0 = Transfer from EISA bus to bus master expansion board (EISA read). This is applicable only to channel 1 and not for channel 0, as channel 0 can perform EISA WRITE transfers only.

Bits 0–21 — R/W — Transfer byte count

If bit 23 in the Base Count register is not set to a 1, the channel suspend bit (CFGSU) in that channel's corresponding configuration register is automatically set to a 1. The bit will be set during the Base register to Current register transfer. This ensures that a channel request for that channel is not generated. When the local processor resets the channel suspend bit to 0 in the corresponding Configuration register, a transfer request will be generated.

NOTE:
If the initial byte count is programmed to be "0", no transfer request will be generated and no transfer will occur.

8.2.3.4 Channel 0 and 1 Transfer Current Count Registers (Read Only)

Each Channel has an associated Current Count register set. The Transfer Current Count registers contain the 22-bit value representing the number of bytes remaining to be transferred on the channel. This value can be from one byte to four Mbytes. Bit 23 is reserved. Bit 22 is used to indicate the direction of the transfer. Upon reset, the Current Count registers are not initialized. At the end of a transfer, this register contains the value of the number of bytes transferred during the last cycle.

Channel 0 Local Index Address—50h through 52h (bytes 0 through 2)
Channel 1 Local Index Address—70h through 72h (bytes 0 through 2)

BYTE 2			BYTE 1	BYTE 0
Bit 23	Bit 22	Bit 16–21	Bits 8–15	Bits 0–7

Bit 23 — Reserved

Bit 22 — 1 = Current transfer is from bus master expansion board to EISA bus
0 = Current transfer is from EISA bus to bus master expansion board

Bits 0–21 — Current transfer byte count

8.2.4 DATA TRANSFER STATUS/CONTROL REGISTERS

8.2.4.1 Channel 0 and 1 Transfer Strobe Registers (Write Only)

Each channel has an associated Transfer Channel Strobe register. The Strobe register is used to initiate the transfer of information from the Base register set to the Current register set. The act of writing to this register will initiate the Base to Current transfer. There are no bits to this register, the data written to this register is ignored and the register cannot be read.

If bit 23 in the Transfer Base Count Register is set to a 1, the data transfer will be requested immediately. Otherwise, the transfer will wait until the transfer suspend bit CFGSU for the corresponding channel is reset. The transfer suspend bit is located in the Configuration register.

Channel 0 Local Index Address—49h
Channel 1 Local Index Address—69h

8.2.4.2 Channel 0 and 1 Transfer Channel Configuration Registers (Read/Write)

Each channel has an associated Transfer Configuration register. Upon reset, the Configuration registers are reset to 0. The Configuration register set is used to configure the channels as follows:

Channel 0 Local Index Address—48h
Channel 1 Local Index Address—68h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CFGEA	CFGIE	CFGIT	CFGFF	CFGBR	CFGCL	CFGEI	CFGSU

Bit 7 — CFGEA 1 = Enable real-time address transfer to transfer buffer logic
0 = Disable real-time address transfer to transfer buffer logic

Bit 6 — CFGIE Reserved. This bit must always be written with 0.

Bit 5 — CFGIT 1 = Enable interrupt on transfer complete
0 = Disable interrupt on transfer complete

Bit 4 — CFGFF 1 = Give up ownership of EISA bus if a transfer interruption occurs on this channel
0 = Retain ownership of EISA bus if a transfer interruption occurs on this channel

Bit 3 — CFGBR 1 = Enable EISA burst transfer
0 = Disable burst transfers (channel uses non-burst (2 BCLK) cycle transfers)

Bit 2 — CFGCL 1 = Clear channel
Stop any transfers and flush the data FIFO
0 = No operation
Always returns a 0 when read

Bit 1 — CFGEI Reserved. This bit must always be written with 0.

Bit 0 — CFGSU 1 = Temporarily suspend transfer
0 = Allow transfer to proceed

The CFGEA Bit enables the real-time address transfer to the transfer buffer logic. If the CFGEA bit is set to a 1, the transfer buffer real-time address for the active channel is transferred to the transfer buffer logic each time that channel regains the bus and the start address is transferred each time the Base register contents are loaded into the corresponding Current registers. If the CFGEA bit is set to 0, the address load signal (TLD#) is activated only when the Base is loaded into the Current register (refer to Section 5.3).

The CFGIE Bit is a reserved bit. Zero (0) must always be written at this bit location. This bit can be ignored during register reads.

The CFGIT Bit enables an interrupt on transfer complete (EOP).

The CFGFF Bit controls whether EISA bus ownership is relinquished or maintained after a transfer interruption. When a channel is interrupted for any reason, (1K page break, FIFO stall, channel clear, transfer suspend, or transfer complete), the BMIC may relinquish the EISA bus depending on the state of the CFGFF bit in the above register. The function of the CFGFF bit, as related to the above channel interruptions, is as follows:

If the CFGFF bit = 1, the BMIC will relinquish control of the EISA bus upon the detection of any of the above interruptions. This will occur regardless if there are additional data transfer requests pending. If there are additional data transfer requests pending, the BMIC will reassert MREQ# a minimum of two BCLK's later to reacquire the EISA bus.

If the CFGFF bit = 0, the BMIC retains ownership of the EISA bus upon detection of a FIFO stall or 1K page break as long as a preempt timer timeout has not occurred. If there are additional data requests pending, the BMIC will immediately perform the pending transfer and then rearbitrate for the EISA bus to complete the interrupted transfer. If there are no additional data requests pending, the BMIC will relinquish ownership of the EISA bus only after the current transfer interruption has been serviced and completed.

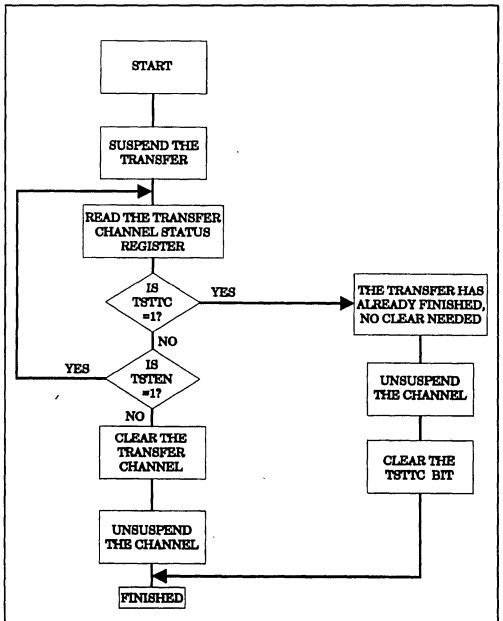
NOTE:
During a FIFO pause, CFGFF is ignored.

The CFGBR Bit defines the type of transfer cycles (burst or non-burst) that can be requested on the transfer channel. If burst cycles have been selected and system memory is unable to run burst cycles, the BMIC will default to non-burst (two BCLK) or mismatched cycles.

The CFGCL Bit is used to generate a channel clear. A channel clear terminates the current transfer and flushes the associated FIFO. The FIFO is reset during the next Base to Current register copy.

Before a channel is issued a clear command, the channel must first be suspended by writing a "1" into Bit 0 (CFGSU) of the Transfer Channel Configuration Register. Next the Transfer Channel Status Register must be read. If the TSTTC (Bit 0) is set to a "1", then the channel has already completed the transfer. The channel is then unsuspended (write a "0" into Bit 0 [CFGSU] of the Transfer Channel Configuration Register), and the TSTTC bit is then cleared.

If the TSTTC bit is a "0", then the TSTEN bit is checked. If this bit is a "1", then the channel has not returned to idle yet, and the Transfer Channel Status Register is re-polled. If the TSTEN bit is a "0", then the channel has successfully returned to idle and can now be cleared with no errors. This is done by setting Bit 2 (CFGCL bit) to a "1" in the Transfer Channel Configuration Register. A flowchart for this operation is shown in the following figure.



NOTE:
TSTTC must ALWAYS be cleared when a Transfer is completed.

Figure 8-1. Channel Clear Flowchart

If a channel is enabled for a transfer during a Channel clear, the BMIC will generate an end of process by asserting TEOP#. If the channel is not enabled for a transfer or the channel clear is preceded by a channel suspend, a TEOP# will not be generated. The channel clear will be active for at least two complete BCLK cycles.

The CFGEI Bit is a reserved bit. Zero (0) must always be written at this bit location. This bit can be ignored during register reads.

The CFGSU Bit is used to temporarily suspend the data transfer.

8.2.4.3 Channel 0 and 1 Transfer Channel Status Registers (Read/Write)

Each channel has an associated Transfer Channel Status register. Bits 2 through 4 are read only and bits 5 through 7 are reserved. Upon reset, the Channel Status register set is reset to "0".

Channel 0 Local Index Address—4Ah
Channel 1 Local Index Address—6Ah

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	TST1K	RSVD	TSTEN	TSTIP	TSTET	TSTTC

Bit 7, 6, 5 —W — Reserved. 0 should be written into these bits during writes. Ignore any data on these bits during register reads

Bit 4 — R — Reserved

Bit 3 — R — 1 = The transfer channel is enabled for transfer (transfer in progress)
0 = The transfer channel is not enabled for transfer (transfer not in progress)

Bit 2 — R — 1 = A transfer request is active on this channel.
0 = No transfer request is active on this channel.

Bit 1 — Reserved. Ignore data at this bit location.

Bit 0 — R/W — 1 = Transfer completed on this channel (read)
Reset this bit (write)
0 = No transfer completion on this channel (read)
No action (write)

Bits (7), (6), TST1K, and (4) are reserved. Any data read from these bits should be ignored.

The TSTIP and TSTEN Bits are read only and indicate whether the corresponding channel is requesting a transfer or whether the channel's transfer is currently in progress.

The TSTET Bit is a reserved bit and should be ignored during all register reads. Zero (0) should always be written at this bit location.

The TSTTC Bit is read/write and is used to indicate the current end-of-process status of the transfer. If an EOP occurs, the BMIC will set bit (0) to a "1" and generate an interrupt to the local processor. The BMIC will not generate the interrupt if the CFGIT bit in the channel's corresponding Transfer Configuration register is set to a "0".

NOTE:
The TSTTC bit is implemented as a sticky bit. This bit can be reset by the local processor without affecting the status of the other bits in the register.

8.2.5 PEEK/POKE REGISTERS

The Peek/Poke register set consists of four 8-bit Address registers, four 8-bit Data registers and one Peek/Poke control register. The Address and Data registers are used to define the 32-bit address and data that will be used during the peek/poke cycles, and the Control register is used to request and define the type of cycle that will be generated (peek, poke, or locked exchange). The peek/poke or locked exchange cycle is initiated by writing to the Peek/Poke control register. During Reset, the Peek/Poke Address registers and the Control register are reset to "0".

8.2.5.1 Peek/Poke Address Registers (Read/Write)

The four 8-bit Peek/Poke Address registers contain the 32-bit peek/poke address. Only the lower 16 bits are used for I/O cycles. Address bits 0 and 1 are ignored. Upon reset, this register is reset to "0".

Local Index Address—34h through 37h (bytes 0 through 3)

BYTE 3	BYTE 2	BYTE 1	BYTE 0
Bits 24–31	Bits 16–23	Bits 8–15	Bits 2–7

8.2.5.2 Peek/Poke Data Registers (Read/Write)

The four 8-bit peek/poke data registers hold the data for the peek/poke cycle. Each peek/poke data register is associated with one byte lane. During peek transfers, only those peek/poke data registers whose corresponding byte enable bit is set in the peek/poke control register contain valid data. During poke transfers, the data must be placed in the appropriate register as determined by the corresponding byte enable bit.

Local Index Address—30h through 33h (bytes 0 through 3)

BYTE 3	BYTE 2	BYTE 1	BYTE 0
Bits 24–31	Bits 16–23	Bits 8–15	Bits 0–7

The Shared register timings (t85, t93, t96–t98) are used when accessing the Peek/Poke Data registers.

8.2.5.3 Peek/Poke Control Register (Read/Write)

The Peek/Poke Control register is written to by the local processor when a peek/poke transfer is desired over the EISA bus. Upon reset, this register is reset to "0".

Local Index Address—38h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bit 7 — Reserved. Set to 0

Bits 6, 5 — 10 = Do read cycle (peek)

Bits 6, 5 — 01 = Do write cycle (poke)
11 = Do locked exchange cycle (peek/poke)
00 = Do Nothing (Nop)

Bit 4 — 1 = Do memory cycle
0 = Do I/O cycle

Bit 3
Bit 2
Bit 1
Bit 0
1 = Byte enable for given byte lane
0 = Byte disable for given byte lane

Bits (6) and (5) are used to define the type of cycle requested (peek, poke or locked exchange).

Bit (4) defines whether the cycle is memory or I/O.

Bits (3–0) are used to define the byte enables for the doubleword data written to or read from the Peek/Poke data register. Peek/Poke cycles will not be generated for illegal combinations of byte enables.

ILLEGAL COMBINATIONS OF BYTE ENABLES:

Bits 3–0	IBE# <3:0>
0000	1111
0101	1010
1001	0110
1010	0101
1011	0100
1101	0010

NOTE:
Bits 3–0 in the above register are active high whereas the EISA byte enables (IBE# <3:0>) are active low.

8.2.6 I/O RANGE DECODE REGISTERS

The I/O Decode Range register set consists of two I/O Decode Range Base Address registers and two I/O Decode Range Control Registers. The Address registers are used to define the address range of interest to the expansion board and the Control registers are used to define the decode range size, type of decode (slot specific or general), and the response of the local I/O (32-bit EISA or 8-bit EISA). The I/O decode register set controls the two IOSEL# pins on the BMIC. Each pin has an associated Address and Control register.

Upon reset, the I/O Decode Range registers are initialized according to the following table:

Local Processor	*Local Processor Not Present		*Local Processor Present	
Range	**Rng 0**	**Rng 1**	**Rng 0**	**Rng 1**
Control Registers	60h	60h	20h	20h
Address Registers	E0h	00h	00h	00h

*Refer to Section 7.3 for information regarding "local processor present" or "local processor not present".

8.2.6.1 Range 0 and 1 I/O Decode Base Address Registers (Read/Write)

Each Decode range and IOSEL# pin has an associated I/O Decode Range Base Address register.

Range 0 Local Index Address—39h
Range 1 Local Index Address—3Bh

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

During general I/O decode, bits 7–0 are used to compare against EISA address lines LA<9:2>. During slot specific decode, bits 7 and 6 are compared against EISA address lines LA<11:10> and bits 5–0 are compared against EISA address lines LA<7:2> (refer to Section 4.8).

8.2.6.2 Range 0 and 1 I/O Decode Control Registers (Read/Write)

Each Decode range and IOSEL# pin has an associated I/O Decode Range Control register.

Range 0 Local Index Address—3Ah
Range 1 Local Index Address—3Ch

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bit 7 — 1 = Respond as a 32-bit EISA I/O device
0 = Respond as an 8-bit EISA I/O device

Bit 6 — 1 = Slot Specific I/O Decode
0 = General I/O Decode

Bit 5 — 1 = IOSEL# held during CMD# active
0 = IOSEL# follows I/O address changes

Bit 4, Bit 3 — 1 = Do not compare I/O Range Base Address Register and corresponding EISA address bit (Mask)

Bit 2, Bit 1, Bit 0 — 0 = Compare I/O Range Base Address Register with corresponding EISA address bit

Refer to Section 4.8 for a complete description of the I/O Decode Range Control registers and the BMIC decode function in general.

8.2.7 TRANSFER BUFFER INTERFACE (TBI) REGISTERS (READ/WRITE)

The TBI registers are programmed to provide the 16-bit start address of the data transfer for use by the transfer buffer logic (refer to Section 5.3). Each transfer channel has a corresponding TBI Base and Current Address register set. The contents of the TBI Base Address registers are transferred to the TBI Current Address registers during a write to the channel's corresponding Transfer Channel Strobe Register.

8.2.7.1 Channel 0 and 1 TBI Base Address Registers (Read/Write)

The BMIC provides two 8-bit TBI Base Address registers per channel. The registers are programmed with the 16-bit start address of the data in the Transfer Buffer memory space. The TBI Base Address register set is not initialized to a fixed value upon reset.

Channel 0 Local Index Address—4Bh and 4Ch (byte 0 and 1)
Channel 1 Local Index Address—6Bh and 6Ch (byte 0 and 1)

Byte 1	Byte 0

8.2.7.2 Channel 0 and 1 TBI Current Address Registers (Read Only)

The BMIC provides two 8-bit TBI Current Address registers per channel. The TBI Current Address registers contain the 16-bit real-time address of the data transfer. The contents of the Current register set are transferred to the external buffer logic at the beginning of every new data block transfer. The BMIC may also be programmed to transfer the contents of the Current Address register each time the corresponding channel regains control of the bus (refer to Section 5.3). The TBI Current Address register set is reset to "0" upon device reset.

Channel 0 Local Index Address—58h and 59h (byte 0 and 1)
Channel 1 Local Index Address—78h and 79h (byte 0 and 1)

Byte 1	Byte 0

NOTE:
The TBI current registers contain real-time status and may change at anytime. If a stable value is needed while reading a set of these registers, the channel should be temporarily suspended by setting the CFGSU bit in the Channel Configuration register to a "1" before these registers are read.

9.0 DETAILED PIN DESCRIPTION

9.1 EISA Interface Signals

Pin Name	Description
START#	I/O TRI-STATED (EISA CYCLE START STROBE) The START# signal provides timing control at the start of a cycle. During EISA master mode, the BMIC drives this signal low after LA<31:2> and M/IO become valid and negates START# on the rising edge of BCLK after one BCLK cycle time. During EISA slave mode, the BMIC uses this signal to indicate the start of a slave bus cycle. It is sampled on the rising edge of BCLK. Upon reset, this pin is tri-stated and placed in input mode.
CMD#	INPUT (EISA COMMAND STROBE) The CMD# provides timing control within the cycle. The 82358 Bus Controller asserts CMD# on the rising edge of BCLK, simultaneously with the negation of START#. CMD# is held low until the end of the cycle. The BMIC uses CMD# in EISA slave mode for timing control during internal Shared register read/write accesses.
M/IO	I/O TRI-STATED (EISA MEMORY/IO CYCLE STATUS PIN) M/IO is used to indicate that the type of cycle in progress is a memory cycle (high) or I/O cycle (low). M/IO is pipelined from one cycle to the next and must be latched by the addressed memory slave if needed for the whole cycle. During EISA master mode, the BMIC drives this signal. The BMIC will drive this pin high during burst and non-burst (two BCLK) cycles. The value of M/IO in a Peek/Poke or locked exchange cycle depends on the programmed value of bit 4 in the Peek/Poke Control register. During EISA slave mode, the M/IO pin is an input. As a slave, the BMIC will respond only as an I/O device. Upon reset, this pin is tri-stated and placed in input mode.
W/R	I/O TRI-STATED (EISA WRITE/READ CYCLE STATUS PIN) The W/R status signal identifies the cycle as a write (high) or read (low). W/R is pipelined from one cycle to the next and must be latched by the addressed memory slave if needed for the whole cycle. During EISA master mode, the BMIC drives this signal. During EISA slave mode, this pin is an input. Upon reset, W/R is tri-stated and placed in input mode.
EXRDY	I/O OPEN COLLECTOR (EISA READY SIGNAL) EXRDY is used by EISA I/O and memory slaves to request wait states during a cycle. Each wait state is one BCLK period. During EISA master mode, the BMIC first samples this signal on the falling edge of BCLK after CMD# is asserted. If it is low, the BMIC will insert a wait state, and continue inserting wait states as long as EXRDY is low at each successive falling edge of BCLK. During EISA slave mode, the BMIC drives EXRDY inactive until it is ready to complete cycles addressed to it. The EXRDY pin is an open collector output.
EX32#	I/O OPEN COLLECTOR (EISA 32-BIT SLAVE RESPONSE PIN) EX32# is an open collector and is used by memory or I/O slaves to indicate their support of 32-bit transfers. During EISA master mode, the BMIC samples EX32# on the same rising edge of BCLK that START# is deasserted. The BMIC uses this pin to determine if the addressed slave is capable of 32-bit transfers. During peek/poke and non-burst EISA data transfers, the BMIC is a 32-bit master only and will allow the 82358 Bus Controller to do all necessary bus conversions. During EISA slave mode, the BMIC drives EX32# low if it has 32-bit data to send to the EISA bus, otherwise this signal is inactive.

9.1 EISA Interface Signals (Continued)

Pin Name	Description
MASTER16#	OUTPUT OPEN COLLECTOR (EISA 16-BIT MASTER CONTROL) In master mode, the BMIC will assert MASTER16# (at the same time as START#) for one BCLK period when it is capable of downshifting from a 32-bit master to a 16-bit master. The BMIC will downshift if necessary during memory burst transfers only. The BMIC will automatically downshift from a 32- to 16-bit master if the EX32# signal is sampled inactive and the SLBURST# signal is sampled active. MASTER16# has no function in slave mode.
AEN	INPUT (EISA ADDRESS ENABLE SIGNAL) The BMIC uses AEN when in EISA slave mode to qualify I/O addresses. When negated (low), the BMIC uses AEN to decode possible accesses to its general and slot specific I/O space. When asserted (high), the address on the EISA bus will be ignored by the BMIC. AEN is sampled on the falling edge of CMD#. This signal is not used in master mode.
MSBURST#	OUTPUT TRI-STATED (EISA MASTER BURST SIGNAL) The BMIC asserts MSBURST# to indicate to the addressed memory slave that the BMIC will provide burst cycles. If the BMIC samples SLBURST# active on the rising edge of BCLK after START# is asserted, the BMIC will activate MSBURST# on the next BCLK falling edge and will proceed with burst cycles. If the BMIC samples SLBURST negated, MSBURST# will not be activated and the BMIC will proceed with either non-burst (two BCLK) or mismatched cycles, depending on the size of the slave device addressed. This signal is not used in slave mode. Upon reset, this pin is tri-stated.
SLBURST#	INPUT (EISA SLAVE BURST SIGNAL) The BMIC uses this signal in master mode to determine if the addresed slave memory is capable of supporting burst transfers. If the BMIC samples SLBURST# active on the rising edge of BCLK after START# is asserted, the BMIC will proceed with burst cycles. If the BMIC samples SLBURST# negated, either non-burst (two BCLK) or mismatched cycles will be generated.
LOCK#	OUTPUT TRI-STATED (EISA RESOURCE LOCK SIGNAL) The BMIC asserts this signal to guarantee exclusive memory and I/O access during locked peek/poke exchange. Upon reset, this pin is tri-stated.
MREQ#	OUTPUT (EISA MASTER BUS REQUEST SIGNAL) MREQ# is asserted by the BMIC to request EISA bus access. The BMIC will begin driving the bus with the address and control signals on the falling edge of BCLK, two BCLKs after MAK# is sampled active. During an EISA write transfer, MREQ# will not be asserted until the FIFO on the selected channel is full. During an EISA read transfer, MREQ# will be asserted immediately after receiving a transfer request, assuming that a slave cycle is not currently in progress. Upon reset, this pin is driven inactive high.
MAK#	INPUT (EISA MASTER BUS ACKNOWLEDGE SIGNAL) The MAK# signal is asserted by the 82357 (ISP) to grant EISA bus access to the BMIC. The BMIC samples MAK# on the falling edge of BCLK and will begin driving the bus with the address and control signals on the falling edge of BCLK, two BCLKs after MAK# is sampled active. The MAK# signal may be negated by the ISP to indicate to the BMIC that another device requires EISA bus access. The BMIC will negate MREQ# to release the bus within 64 BCLKs (8 μs) of sampling MAK# negated.
EINT	OUTPUT OPEN COLLECTOR (EISA INTERRUPT REQUEST SIGNAL) The EINT line is used by the BMIC to interrupt the system CPU or EISA bus master to request service. EINT can be programmed for either edge or level-triggered operations and is an open collector output in level-triggered mode. Upon reset, EINT is placed in level-triggered mode and floating.
BCLK	INPUT (EISA BUS CLOCK) This clock signal is used by the BMIC to synchronize the EISA control signals and data transfers to the system clock. BCLK typically runs at a frequency of 8.33 MHz with a normal duty cycle of 50%. The BCLK period is sometimes extended by the 82358 (EBC) by up to one BCLK period for synchronization purposes.

9.1 EISA Interface Signals (Continued)

Pin Name	Description
RESET	INPUT (EISA RESET SIGNAL) This signal is used by the BMIC to initialize all of its internal registers and state machines to a known state. This signal is asynchronous with respect to BCLK. To reset the BMIC properly, the RESET signal must be active for eight BCLK periods.
IDAT<31:0>	I/O TRI-STATED (EISA DATA LINES/UPPER 22 ADDRESS LINES) These data signals interface to the EISA bus through external, 74F245 bi-directional TTL buffers. The upper 22 data lines are also multiplexed to function as the upper 22 EISA address lines. The 22 upper address signals are latched into external 74F573 TTL latches during transfers as necessary by the BMIC. Both the external data buffers and the address latches are controlled by the BMIC during all slave and master mode data transfers. Upon reset, these pins are tri-stated.
IADS<11:10>	(INPUT) (EISA ADDRESS INPUT LINES) These two address lines are input only and are only used during slave mode. They are used along with IADS<9:2> and EISA byte enables IBE<3:0># for I/O address decoding. The corresponding EISA output address lines LA<11:10> are part of the upper 22 address lines that are multiplexed and sent out through the upper 22 data lines.
IADS<9:2>	I/O TRI-STATED (EISA LOWER ADDRESS LINES) These eight address lines are part of the lower EISA address lines and are connected directly to the EISA bus. When the BMIC is a master, it drives these lines directly to the EISA bus. The upper 22 addresses are latched from the data bus. IADS<9:2> are pipelined from one cycle to the next and should be latched by the addressed slave if required for the whole cycle. When the BMIC is a slave, it monitors these lines along with EISA address lines IADS<11:10> and EISA byte enables IBE<3:0># for I/O address decoding. Upon reset, these pins are tri-stated and placed in input mode. The following address lines are used during I/O decoding as shown: Slot specific I/O address decoding (expansion board)—IADS<11:2> Slot specific I/O address decoding (shared registers)—IADS<11:2>/IBE<3:0># General I/O address decoding (expansion board)—IADS<9:2>
IBE<3:0>#	I/O TRI-STATED (EISA BYTE ENABLES) IBE#<3:0> are the byte enables of the EISA bus and identify the specific bytes that are active during the current EISA bus cycle. During EISA master mode, the BMIC drives these signals. IBE#<3:0> are pipelined from one cycle to the next and should be latched by the addressed slave if required for the whole cycle. During EISA slave mode, the byte enables are inputs and are used along with EISA address lines IADS<11:2> for internal shared register decoding. Upon reset, these pins are tri-stated and placed in input mode.

9.2 EISA Buffer Control Signals

Pin Name	Description
UALOE#	OUTPUT (EISA UPPER ADDRESS LATCH STROBE AND OUTPUT ENABLE) The UALOE# signal is used by the BMIC to control the external latching of the upper 22 address lines LA<31:10>. UALOE# is designed to be connected to the latch enables and output enables of the 74F573 external address latches. The BMIC updates the external address latches at the beginning of all master mode transfers. The desired address value is placed on the IDAT<31:10> lines and latched by the external latches on the falling edge of UALOE# at the beginning of the transfer. During EISA master mode to enable the EISA address lines <31:10>, the BMIC drives UALOE# low on the rising edge of BCLK, one BCLK prior to the falling edge of START#. UALOE# will remain active until the end of the cycle. During slave mode, the BMIC holds UALOE# high to disable the latches. For additional information with regards to the timing for this signal, refer to the A.C. timing and Basic Function timing sections. Upon reset, this pin is driven inactive high.
IDDIR	OUTPUT (EISA DATA DIRECTION SIGNAL) The IDDIR signal is used by the BMIC to control the direction of the external 74F245 data buffers. During data transfers from the BMIC to the EISA bus, this signal will be driven low. During data transfers from the EISA bus to the BMIC, this signal will be driven high. For additional information regarding the timing for this signal, refer to the A.C. timing and Basic Function timing sections (master and slave). Upon reset, this pin is driven high.
IDOE23# IDOE1# IDOE0#	OUTPUT (EISA DATA BYTE LANE BUFFER ENABLES) The IDOE# signals are used by the BMIC to control the output enables on the external 74F245 data buffers. The IDOE# signals will be driven so that the data buffers are enabled at the appropriate times during master and slave transfers. For additional information with regards to the timing for these signals, refer to the A.C. timing and Basic Function timing sections. Upon reset, these signals are driven inactive high.

9.3 Address Decode Signals

Pin Name	Description
IOSEL# <1:0>	OUTPUT (ADDRESS RANGE DECODE OUTPUTS) The IOSEL# signals are used by the BMIC to enable external logic on the expansion board during slot specific and general purpose I/O decode. These pins become active when the LA<11:2> address lines on the EISA bus contain a value mapped into one of the two possible I/O address decode ranges provided by the BMIC (refer to Section 4.8). Upon reset, these pins are driven inactive high.

9.4 Transfer Buffer Interface Signals

Pin Name	Description
TRQ#	OUTPUT (LOCAL DATA TRANSFER REQUEST SIGNAL) When a data transfer is desired over the Transfer Buffer interface, TRQ# is driven low, indicating to the transfer buffer logic that a transfer is following. TRQ# will remain active until the data transfer is completed or a transfer interruption occurs. Upon reset, this pin is driven inactive high.
TACK#	INPUT (LOCAL DATA TRANSFER ACKNOWLEDGE SIGNAL) External logic uses this signal to acknowledge the transfer of a data item (16-bit word) over the Transfer Buffer interface.
TLD#	OUTPUT (LOCAL ADDRESS COUNTER LOAD SIGNAL) This signal when asserted (low) is used to load the transfer start address and the transfer real-time address into an external address counter as required for data transfers (refer to Section 5.3). TLD# is asserted at the beginning of all new channel accesses to the transfer buffer logic and will remain asserted until acknowledged by TACK#. Upon reset, this pin is driven inactive high.
TDIR	OUTPUT (DATA TRANSFER DIRECTION SIGNAL) This signal is used to inform the transfer buffer logic as to the direction of the current data transfer. When driven (high) data will be transferred from the EISA bus to the expansion board. When driven (low) data will be transferred from the expansion board to the EISA bus. TDIR will be held valid whenever TLD# and TRQ# are active. TDIR will not change states when TRQ# is active. Upon reset, this pin is driven high.
TCHAN	OUTPUT (TRANSFER CHANNEL SELECT SIGNAL) This signal is used by the BMIC to inform the transfer buffer logic as to which channel will be active during the transfer. When driven (low) transfer channel 0 is active and when driven (high) transfer channel 1 is active. TCHAN has the same timings as TDIR and will not change states when TLD# or TRQ# are active. Upon reset, this pin is driven low.
TDAT<15:0>	I/O TRI-STATED (TRANSFER DATA LINES) This bidirectional bus is the BMIC's Transfer Buffer interface data bus. It is used during data transfers between the external transfer buffer logic and the BMIC. The data transferred across the TDAT bus is word aligned. The data lines are also used to transport the transfer address to the transfer buffer logic on the expansion board (refer to Section 5.3). The TDAT bus can be unconditionally disabled by driving the TDOE# signal high. **NOTE:** During EISA write data transfers, the TDAT lines are inputs and operate independent of the value of TDOE#. Upon reset, the TDAT bus is tri-stated.
TDOE#	INPUT (TRANSFER INTERFACE DATA OUTPUT ENABLE) When driven high, this pin can be used by external logic to unconditionally disable the BMIC from driving the TDAT<15:0> lines. This feature eliminates the need for the BMIC to gain prior permission to drive the TDAT bus and also allows external logic the ability to time-share the TDAT bus.
TEOP#	OUTPUT OPEN COLLECTOR (TRANSFER END-OF-PROCESS SIGNAL) This signal is an open collector signal that indicates the end of a transfer to the external transfer buffer logic. TEOP# is driven low by the BMIC to indicate the end of transfer. The TEOP# pin requires an external 2.5K to 3.2K pullup resistor for proper operation.
TCLK	INPUT (TRANSFER CLOCK) All transfer control signals are synchronous to this clock. The frequency should be in the range of 16 MHz to 20 MHz to maintain a 33 Mbyte/sec burst transfer rate over the EISA bus. This clock may be completely asynchronous to the EISA BCLK signal.

9.5 Local Processor Interface Signals

Pin Name	Description
LDAT<7:0>	I/O TRI-STATED (LOCAL PROCESSOR INTERFACE DATA BUS) This bidirectional bus is used to transfer commands and status between the BMIC and the local processor on the expansion board. If a local Processor is not present, this bus will need to be connected to the ISA bus (refer to Section 7.3). Upon reset these pins are tri-stated.
LRD#	INPUT (LOCAL PROCESSOR INTERFACE READ STROBE) The local processor asserts LRD# to indicate to the BMIC that it should drive its data onto the LDAT bus. LRD# is asserted for register access to the BMIC's Local Processor interface. The LADS lines and the LCS# signal must be valid 10 ns before the falling edge of LRD# and remain valid until LRD# is deasserted.
LWR#	INPUT (LOCAL PROCESSOR INTERFACE STROBE) The local processor asserts LWR# to indicate to the BMIC that it may latch data from the LDAT bus. LWR# is asserted for write accesses to the BMIC's Local Processor interface. The LADS lines and the LCS signal must be valid 10 ns before the falling edge of LWR# and remain valid until LWR# is deasserted.
LCS#	INPUT (LOCAL PROCESSOR INTERFACE CHIP) A (low) on this pin enables LWR# and LRD# communication between the BMIC and the local processor on the expansion board. The LRD# and LWR# signals are ignored unless the LCS# signal is active. LCS# must be asserted 10 ns before LRD# and LWR# and remain active until the inactive edge of LRD# and LWR#.
LADS<1:0>	INPUT (LOCAL PROCESSOR ADDRESS SELECT) These address lines are used by the local processor to select the Local Data, Local Index, and Local Status/Control registers. The BMIC uses these registers as part of an indexing scheme to access all of its internal registers (refer to Sections 3.2.6.1 and 8.2.1). **LADS1 LADS0** 0 0 = Local Data register 0 1 = Local Index register 1 0 = Local Status/Control register 1 1 = Reserved
LINT	OUTPUT (LOCAL PROCESSOR INTERRUPT SIGNAL) This signal informs the local processor that an event has occurred which requires the local processor's attention. This pin can be programmed for either active high or active low level operations. After being asserted, LINT will not return to an inactive state until the interrupt has been serviced. The LINT signal is not an open collector output during active low operations and will require external logic if interrupts need to be tied together on the local side. Upon reset, this pin is driven high and placed in active low level mode.
LRDY	I/O (LOCAL PROCESSOR READY) This signal is the acknowledgement from the BMIC to the local processor that it is finished with the current Shared register access cycle. The LRDY pin is also used by external logic to indicate to the BMIC that a local processor is not present. If a local processor is not present, the LRDY signal must be driven low during reset (refer to Section 7.3). If a local processor is present, a weak pullup resistor must be connected to the LRDY output to insure that LRDY is high during the time reset is active.

9.6 Power Supplies

V_{CC} — 11 Power pins

V_{SS} — 13 Ground pins

Total number of power supply pins: 24

10.0 BASIC FUNCTION TIMING DIAGRAMS

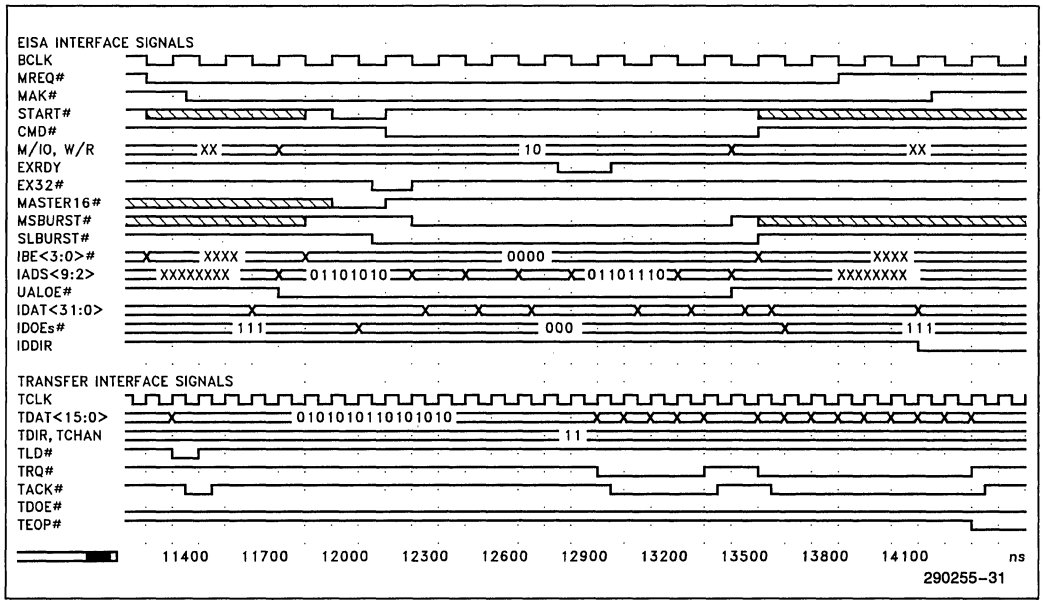


Figure 10-1. 32-Bit Burst Cycle (EISA Read)

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

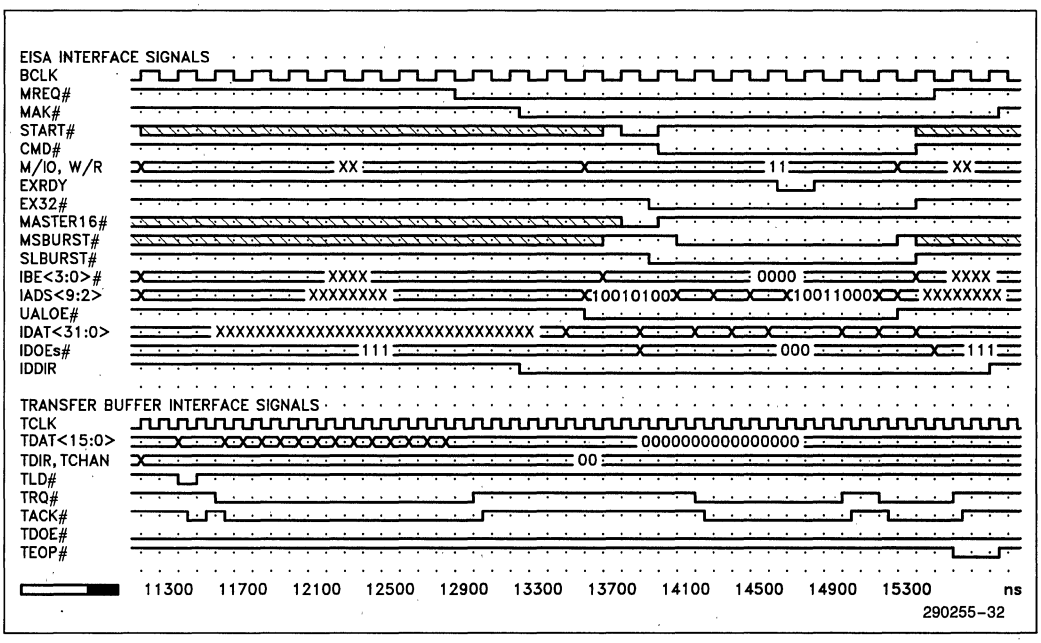


Figure 10-2. 32-Bit Burst Cycle (EISA Write)

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

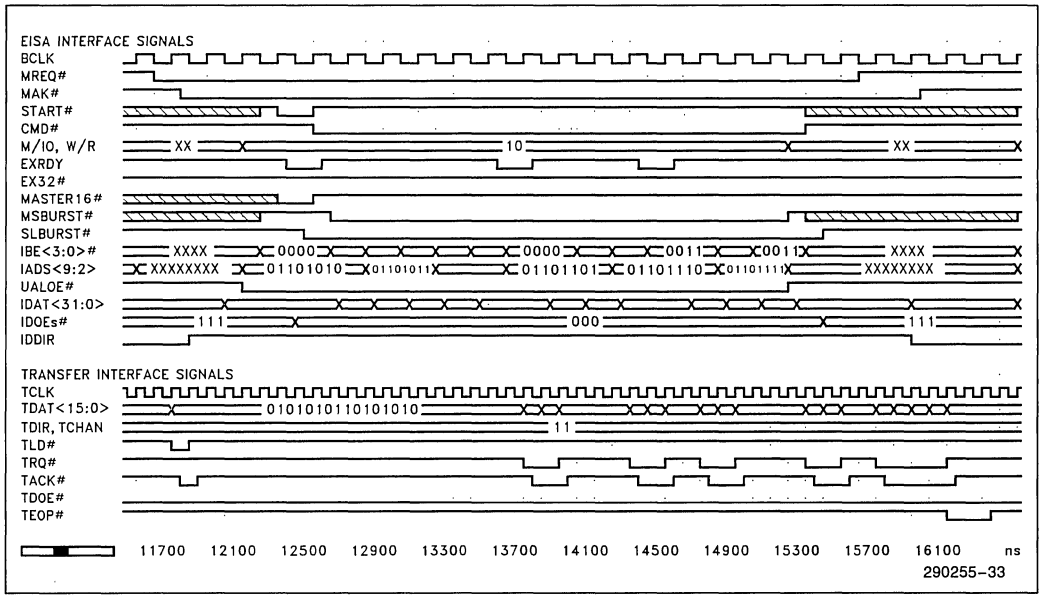


Figure 10-3. 16-Bit Burst Cycle (EISA Read)

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

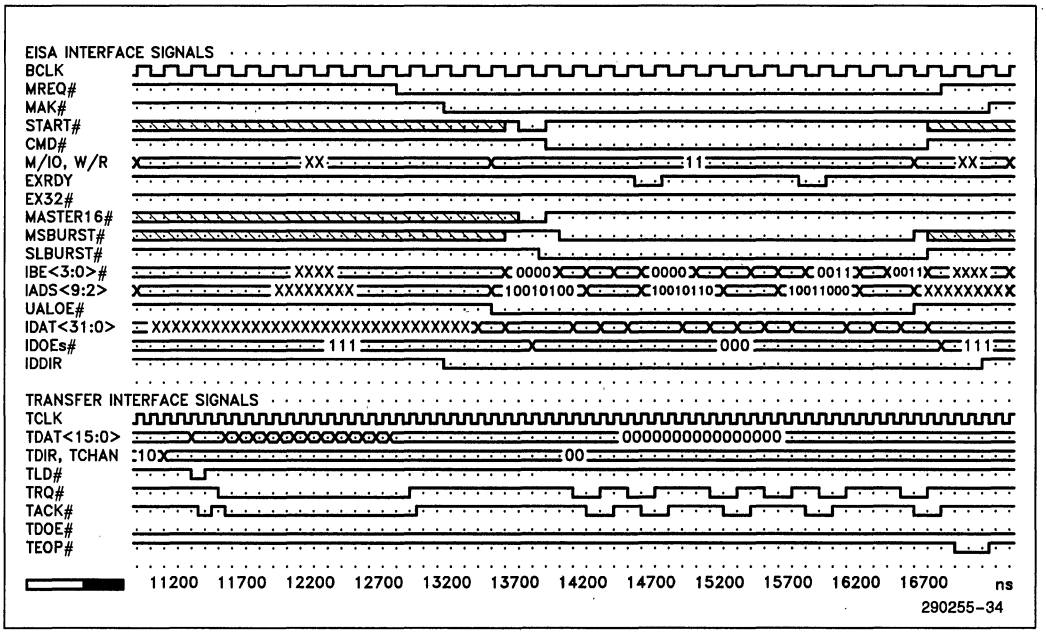


Figure 10-4. 16-Bit Burst Cycle (EISA Write)

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

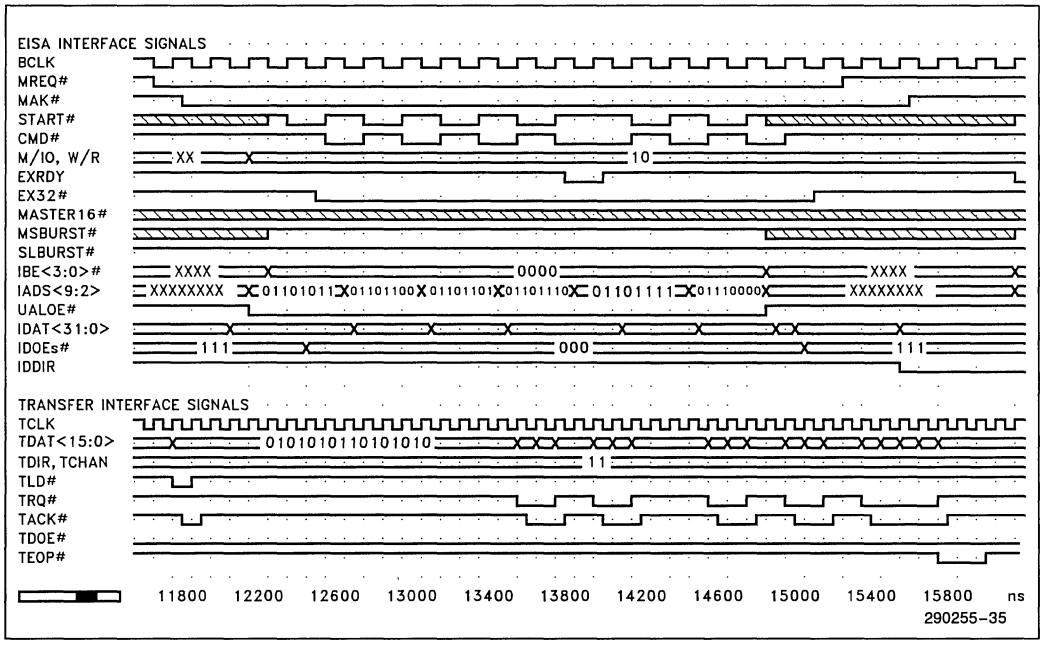


Figure 10-5. 32-Bit Non-Burst Cycle (EISA Read)

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

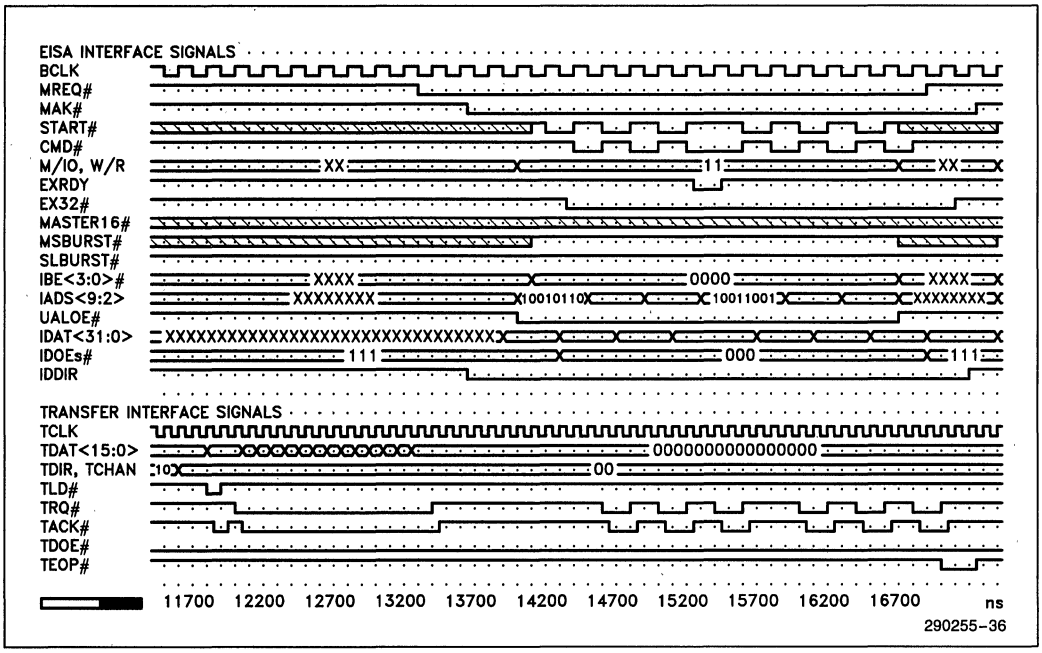


Figure 10-6. 32-Bit Non-Burst Cycle (EISA Write)

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

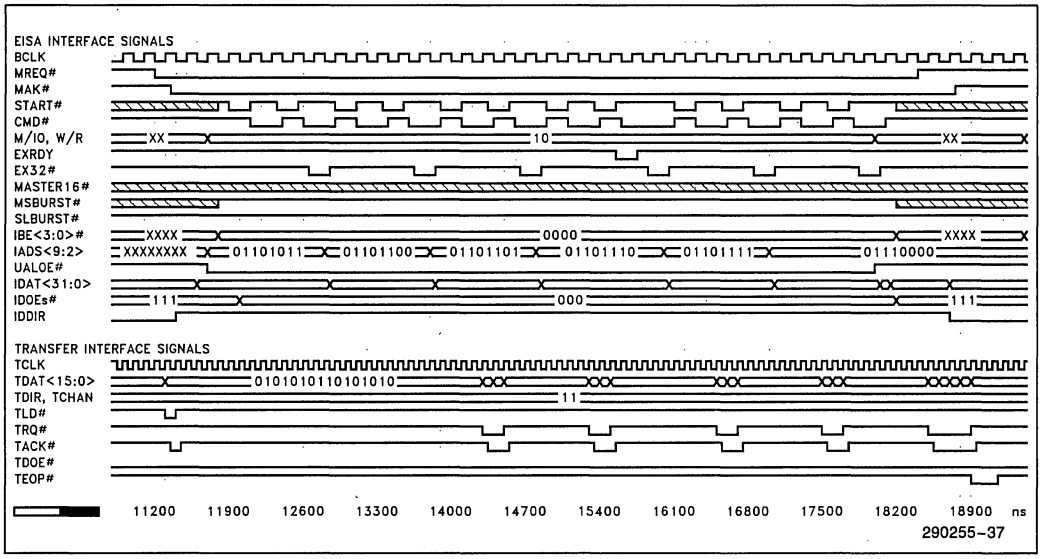


Figure 10-7. Mismatched Cycle (EISA Read)

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

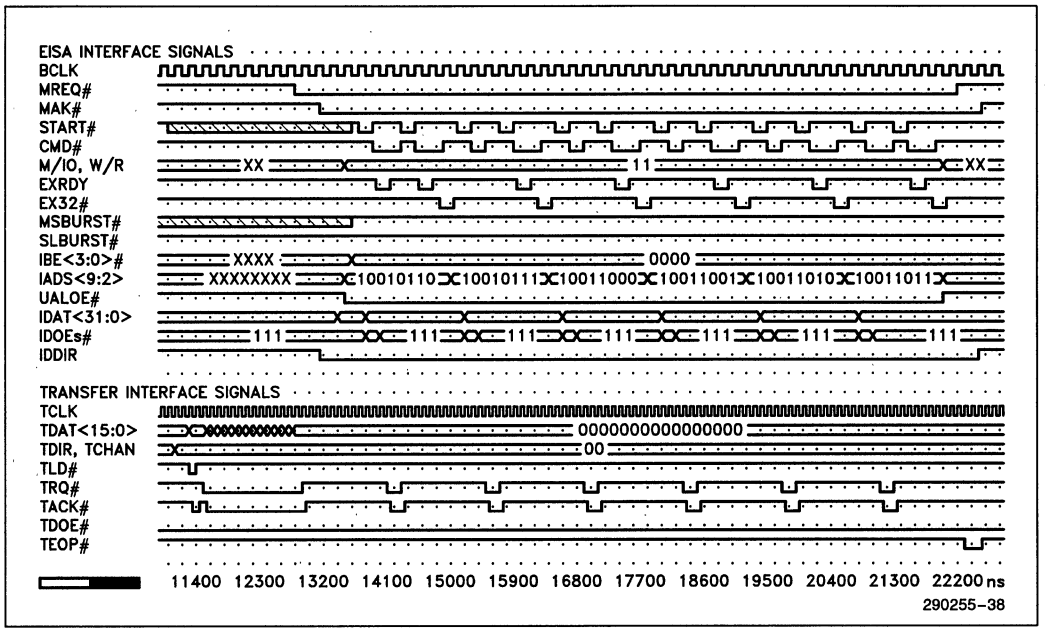


Figure 10-8. Mismatched Cycle (EISA Write)

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

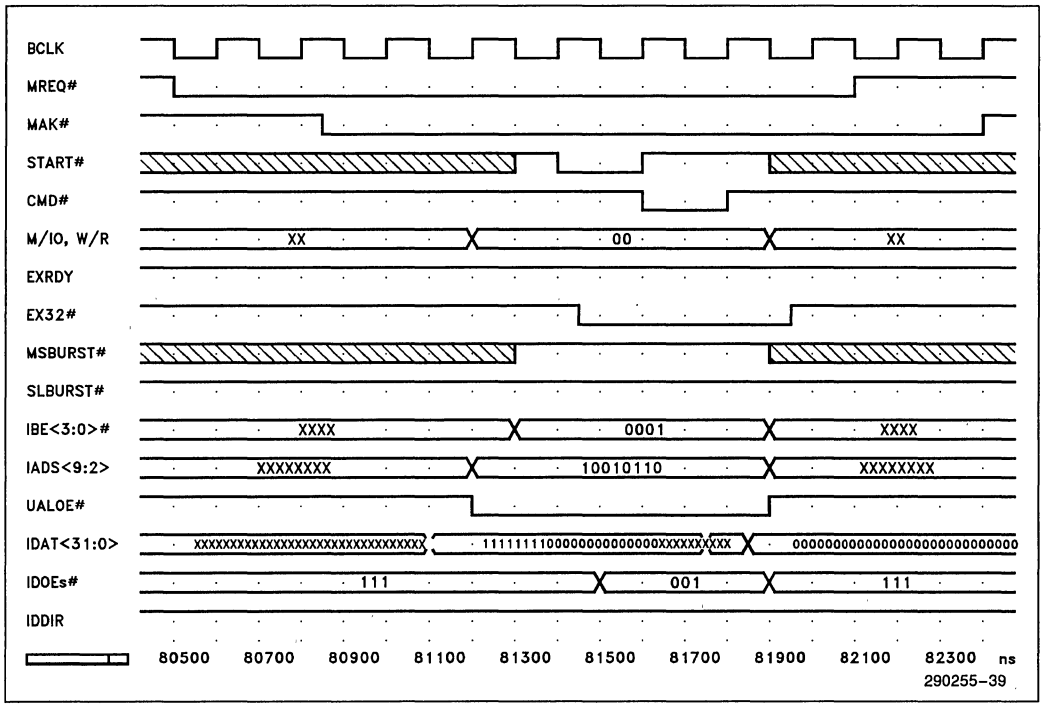


Figure 10-9. I/O Peek Cycle

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

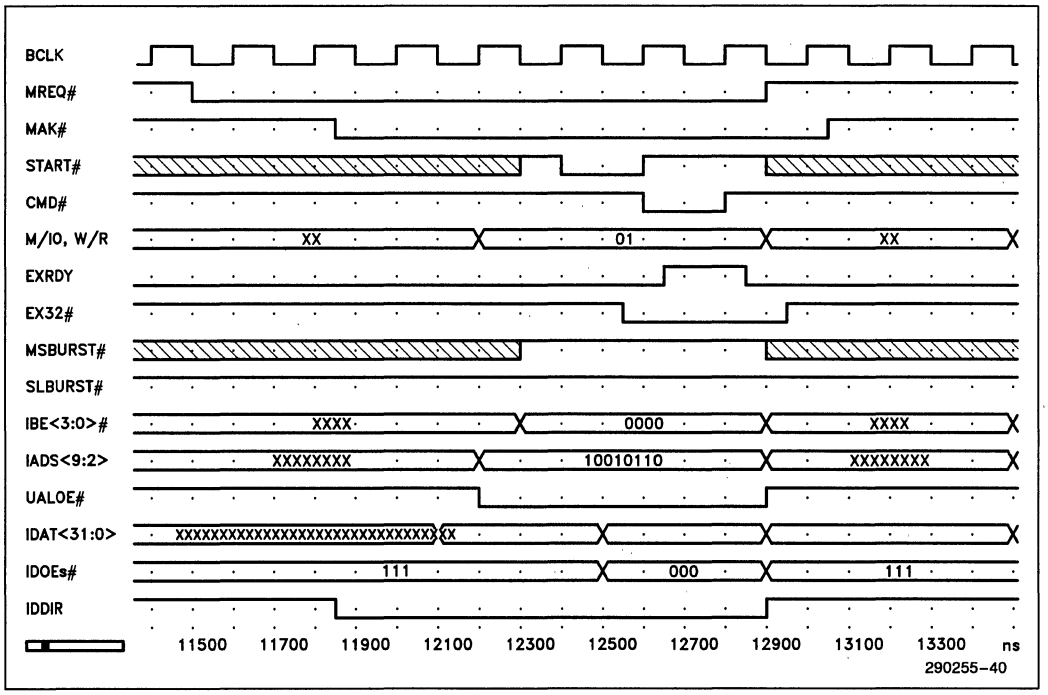


Figure 10-10. I/O Poke Cycle

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

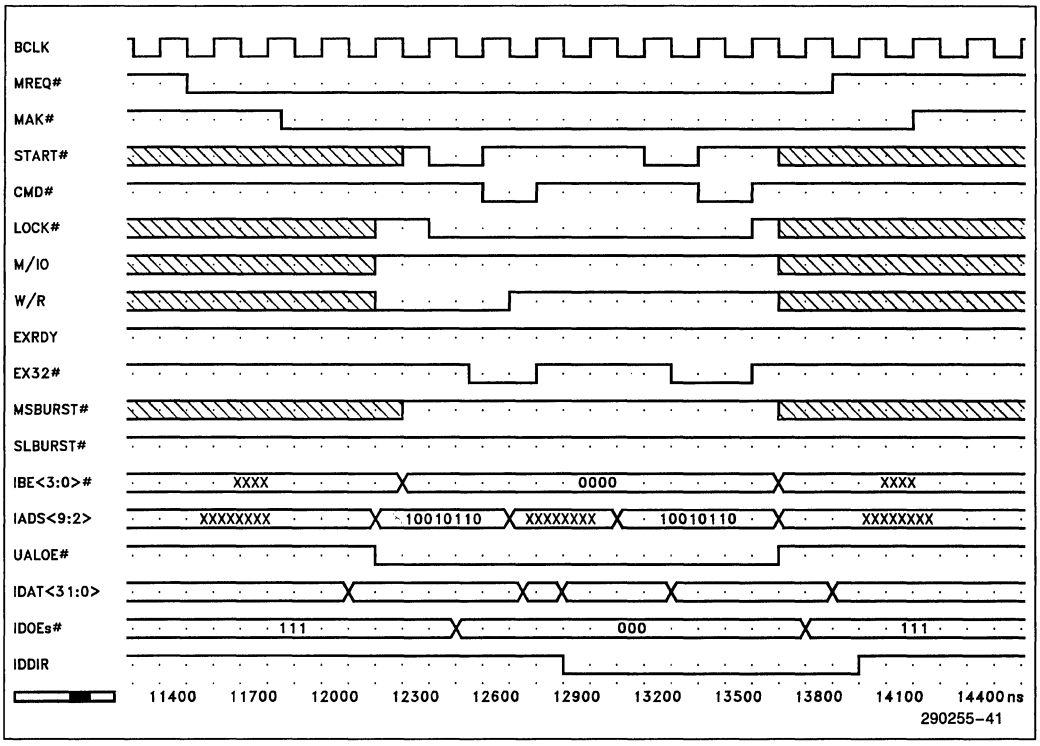


Figure 10-11. Locked Exchange Cycle

10.0 BASIC FUNCTION TIMING DIAGRAMS (Continued)

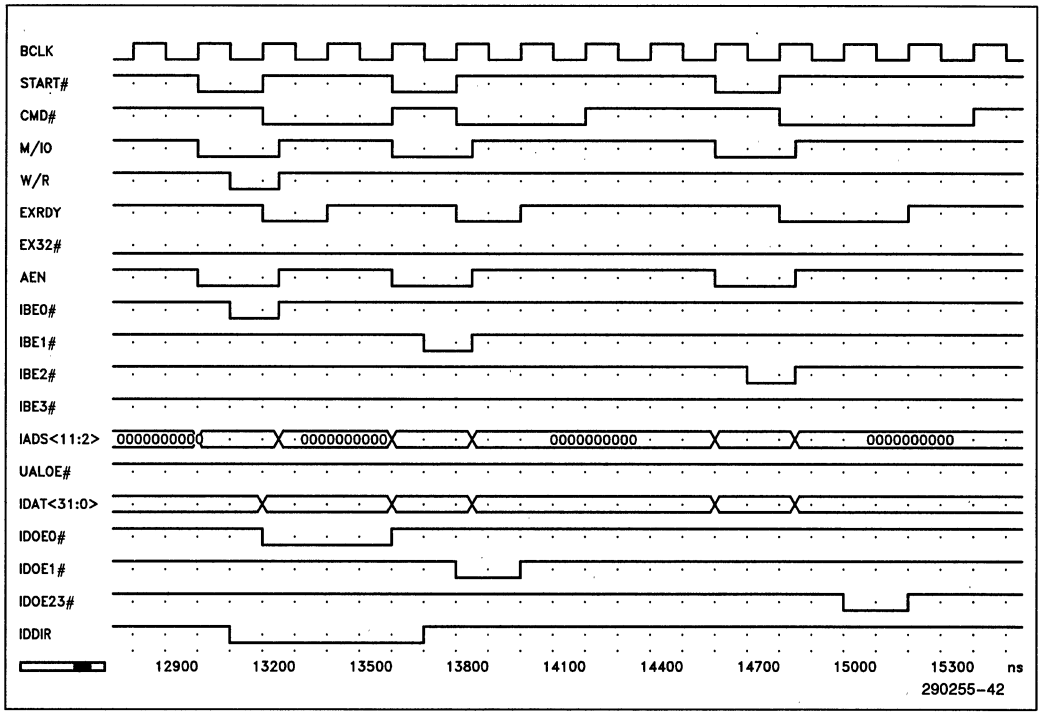


Figure 10-12. Slave Access to BMIC

11.0 D.C. SPECIFICATIONS

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

11.1 Maximum Ratings*

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

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

Supply Voltages with
Respect to Ground . . . . . −0.5V to +6.5V

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

**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.*

11.2 D.C. Characteristics Table

T_{CASE} = 0°C to 70°C, V_{CC} = 5V ±5%, $T_{AMBIENT}$ = 0°C to 55°C

Symbol	Parameter	Limits		Units	Test Conditions
		Min	Max		
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}+0.5	V	
V_{ILC}	CLOCK Input Low	−0.5	0.8	V	
V_{IHC}	CLOCK Input High	2.0	V_{CC}+0.5	V	
V_{OL1}	Output Low Voltage		0.45	V	I_{OL} = 2.5 mA
V_{OH1}	Output High Voltage	2.4		V	I_{OH} = −2.5 mA
V_{OL2}	Output Low Voltage		0.45	V	I_{OL} = 6 mA
V_{OH2}	Output High Voltage	2.4		V	I_{OH} = −4 mA
V_{OL3}	Output Low Voltage		0.45	V	I_{OL} = 24 mA
V_{OH3}	Output High Voltage	V_{CC}−0.4		V	I_{OH} = −100 μA
V_{OL4}	Output Low Voltage		0.45	V	I_{OL} = 4.0 mA
V_{OH4}	Output High Voltage	2.4		V	I_{OH} = −2.5 mA
ILI	Input Leakage		±10	μA	
ILO	Output Leakage		±10	μA	
C_{IN}	Capacitance Input		10	pF	@ 1 MHz(2)
C_{OUT}	Capacitance Output or I/O		12	pF	@ 1 MHz(2)
C_{CLK}	BCLK or TCLK		15	pF	@ 1 MHz(2)
I_{CC}	V_{CC} Supply Current		190	mA	(3)

NOTES:

1. V_{OL1} = IDOE23#, IDOE1#, IDOE0#, LRDY, LDAT<7:0>, IDAT<31:0>, TEOP#, TDIR, TCHAN, IOSEL0#, IOSEL1#, TRQ#, TLD#, and TDAT<15:0>
   V_{OL2} = MREQ#, EINT, and LINT
   V_{OL3} = IADS<9:2>, START#, M/IO, W/R, EXRDY, MASTER16#, EX32#, IBE#<3:0>, MSBURST#, and LOCK#
   V_{OL4} = UALOE#, IDDIR
   V_{OH1} = IDOE23#, IDOE1#, IDOE0#, LRDY, LDAT<7:0>, IDAT<31:0>, TDIR, TCHAN, TRQ#, TLD#, IOSEL0#, IOSEL1#, TDAT<15:0>, MREQ#, EINT, LINT, and TEOP#
   V_{OH2} = IADS<9:2>, START#, M/IO, W/R, IBE#<3:0>, MSBURST#, LOCK#, EXRDY, EX32#, and MASTER16#
   V_{OH3} = UALOE#, IDDIR, IDOE23#, IDOE1#, IDOE0#, IADS<9:2>, LRDY, LDAT<7:0>, IDAT<31:0>, TDIR, TCHAN, EINT, IOSEL0#, IOSEL1#, TRQ#, TLD#, TDAT<15:0>, MREQ#, LINT, IADS<9:2>, START#, M/IO, W/R, IBE#<3:0>, MSBURST#, LOCK#, and TEOP#
   V_{OH4} = UALOE#, IDDIR
   The following outputs are open collector: EXRDY, EX32#, MASTER16#, and TEOP#; EINT is an open collector output when programmed for active low operation.
2. Sampled only
3. Tested at V_{CC} = 5.30V and Frequency = BCLK (8.33 MHz) and TCLK (20 MHz)

12.0 A.C. SPECIFICATIONS

12.1 A.C. Characteristics Tables

The A.C. specifications given in the following tables consist of output delays/float times and input setup and hold times.

T_{CASE} = 0°C to 70°C, V_{CC} = 5V ±5%, $T_{AMBIENT}$ = 0°C to 55°C

BCLK Timing

Symbol	Parameter	Min	Max	Units	Notes
t1	Period	120	2500	ns	Typical = 125 ns
t2	High Time	50		ns	Measured @ 2.0V
t3	Low Time	50		ns	Measured @ 0.8V
t4	Rise Time		10	ns	(13)
t5	Fall Time		10	ns	(13)

Reset Timing

Symbol	Parameter	Min	Max	Units	Notes
t6	Pulse Width	8 (t1)		ns	

Master Timing

Symbol	Parameter	Min	Max	Units	Notes
	MREQ#				
t7	Delay ACT/Inact		33	ns	From BCLK Falling
	MAK#				
t8	Setup Time	10		ns	To BCLK Falling
t9	Hold Time	25		ns	From BCLK Falling
	IADS<9:2>, M/IO, W/R				
t10	Delay Valid	2	45	ns	From BCLK Falling(17)
	IADS<9:2>, M/IO, W/R				
t10a	Delay Valid		75	ns	From BCLK Rising(18)
t11	Delay Float		40	ns	From BCLK Falling(7)
	IBE#<3:0>				
t12	Delay Valid	2	45	ns	From BCLK Falling
t13	Delay Float		40	ns	From BCLK Falling(7, 8)
	START#				
t14	Delay Act/Inact		25	ns	From BCLK Rising
t15	Delay Float		40	ns	From BCLK Falling(7, 8)
	EX32#				
t16	Setup Time	15		ns	To BCLK Rising(9)
t17	Hold Time	50		ns	From BCLK Rising(9)
	EXRDY				
t18	Setup Time	15		ns	To BCLK Falling
t19	Hold Time	2		ns	From BCLK Falling
	IDAT<31:0>				
t20	Delay Valid	3	27	ns	From BCLK Falling(1)
t21	Delay Float		25	ns	From BCLK Falling(7, 8)
t22	Setup Time	4		ns	To BCLK Rising(2)
t23	Hold Time	6		ns	From BCLK Rising(2)
	IDAT<31:10>				
t24	Delay Valid		45	ns	From BCLK Falling(10)
	LOCK#				
t25	Delay Act/Inact	2	60	ns	From BCLK Rising
t26	Delay Float		40	ns	From BCLK Falling(7)
	IDOE#				
t27	Delay Act/Inact		25	ns	From BCLK Falling
	UALOE#				
t28a	Delay Active		60	ns	From BCLK Rising
t28b	Delay Inactive		35	ns	From BCLK Falling
	IDDIR				
t29	Delay Act/Inact		40	ns	From BCLK Falling

1

Master Timing (Burst)

Symbol	Parameter	Min	Max	Units	Notes
	MSBURST#				
t30	Delay ACT/INACT		35	ns	From BCLK Falling
t31	Delay Float		40	ns	From BCLK Rising(8)
	START#, IBE# <3:0>				
t31a	Delay Float		40	ns	From BCLK Rising(19)
	SLBURST#				
t32	Setup Time	15		ns	To BCLK Rising
t33	Hold Time	50		ns	From BCLK Rising
	IDOE#				
t34	Delay Act/Inact		25	ns	From BCLK Rising
	IDAT <31:0>				
t35	Setup Time (Read)	5		ns	To BCLK Rising(2)
t36	Hold Time (Read)	6		ns	From BCLK Rising(2)
t37	Delay Valid	3	27	ns	From BCLK Rising(1)
t38	Delay Invalid	0		ns	From BCLK Rising(1)
	MASTER16#				
t39	Delay Act		50	ns	From BCLK Rising
t40	Delay Float	1	40	ns	From BCLK Rising(7,8)

Slave Timing

Symbol	Parameter	Min	Max	Units	Notes
	IADS<11:12>, M/IO				
t41	Setup Time	120		ns	To CMD# Falling
t42	Hold Time	25		ns	From CMD# Falling
	EX32#				
t43	Delay Act/Float		54	ns	From IADS<11:2>, M/IO
t44	Delay Act/Float		34	ns	From AEN
	AEN				
t45	Setup Time	95		ns	To CMD# Falling
t46	Hold Time	25		ns	From CMD# Falling
	START#				
t47	Pulse Width	110		ns	
	IBE#<3:0>, W/R				
t48	Setup Time	80		ns	To CMD# Falling
t49	Hold Time	25		ns	From CMD# Falling
	EXRDY				
t50	Delay Negated		124	ns	From START# Falling(3)
t51	Delay Float	1	40	ns	From BCLK Falling
	CMD#				
t52	Pulse Width	110		ns	
	IDAT<31:0>				
t53	Setup Time	−35		ns	To CMD# Falling(2)
t54	Hold Time	0		ns	From CMD# Rising(2)
t55	Delay Valid		100	ns	From BCLK Rising(1)
t56	Delay Invalid	0		ns	From CMD# Rising(1)
t57	Delay Float		50	ns	From CMD# Rising
	IDDIR				
t58	Delay Valid		50	ns	From W/R Valid
t59	Delay Invalid	2		ns	From CMD# Rising
	IDOE#				
t60	Delay Act (Read)		25	ns	From CMD# Falling
t61	Delay Inact (Read)		20	ns	From CMD# Rising
t62	Delay Act/Inact (Write)		45	ns	From BCLK Rising
	IOSEL#				
t63	Delay Active		60	ns	From IADS<11:2>
t64	Delay Inactive	5		ns	From CMD# Rising If Latched

Transfer Buffer Interface Timing

Symbol	Parameter	Min	Max	Units	Notes
	TCLK				
t65	Period	50	250	ns	
t66	High Time	18		ns	Measured @ 2.0V
t67	Low Time	20		ns	Measured @ 0.8V
	TRQ#				
t68	Delay Act/Inact		15	ns	From TCLK Rising
	TLD#				
t69	Delay Act/Inact		25	ns	From TCLK Rising
	TEOP#				
t70	Delay Act/Float		25	ns	From TCLK Rising
	TCHAN, TDIR				
t73	Setup Time	25		ns	To TLD# or TRQ# Active(11)
	TACK#				
t74	Setup Time	15		ns	To TCLK Rising
t75	Hold Time	1		ns	To TCLK Rising
	TDAT<15:0>				
t76	Delay Valid	4	25	ns	From TCLK Rising/TDOE# Falling
t77	Delay Float		25	ns	From TCLK/TDOE# Rising
t78	Setup Time	10		ns	To TCLK Rising
t79	Hold Time	1		ns	From TCLK Rising
t80	Ratio of TCLK to BCLK	1.1			

Local Processor Interface Timing (Read Cycle)

Symbol	Parameter	Min	Max	Units	Notes
	LADS<1:0>, LCS#				
t81	Setup Time	10		ns	To LRD# Falling
t82	Hold Time	0		ns	From LRD# Rising
	LRD#				
t83	Pulse Width	150		ns	
	LDAT<7:0>				
t84	Delay Valid		130	ns	From LRD# Falling(4)
t85	Max Delay Valid		2.5 (t1) + 120	ns	From LRD# Falling(5)
t86	Delay Float		40	ns	From LRD# Rising
	LRD# (Inact) to LRD# (Act) or LWR# (Act)				
t87	Recovery Time	60		ns	

Local Processor Interface (Write Cycle)

Symbol	Parameter	Min	Max	Units	Notes
	LADS<1:0>, LCS#				
t88	Setup Time	10		ns	To LWR# Falling
t89	Hold Time	0		ns	From LWR# Rising
	LWR#				
t90	Pulse Width	100		ns	(4)
	LDAT<7:0>				
t91	Setup Time	60		ns	To LWR# Rising(4)
t92	Hold Time	10		ns	From LWR# Rising
t93	Data Valid		70	ns	From LWR# Falling(5)
	LWR# (Inact) to LWR# (Act) or LRD# (Act)				
t94	Recovery Time	60		ns	

Local Processor Ready Timing

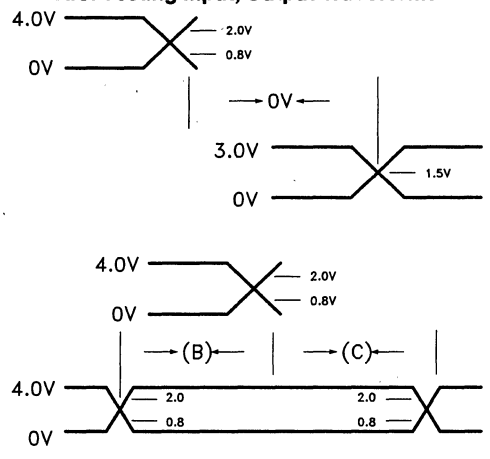
Symbol	Parameter	Min	Max	Units	Notes
	LRDY				
t95	Delay Inactive		50	ns	From LADS and LCS# Valid(5)
	Delay Active				From LRD# or LWR# Active
t96	Max Delay		3.5 (t1)+60	ns	(5, 6)
t97	Max Delay		2.5 (t1)+60	ns	(5, 6)
t98	Min Delay	1.5 (t1)		ns	(5, 6)
	LDAT<7:0>				
t99	Delay Valid		0	ns	From LRDY Rising(5, 12)

NOTES FOR A.C TIMINGS:

1. Specification does not include allowance for 13 ns max. and 2 ns min. into 240 pF for external buffer delay to EISA bus.
2. Specification does not include allowance for 8 ns max. and 1 ns min. into 25 pF for external input delay from EISA bus.
3. Delay includes 40 ns for pull-up rise time (300Ω into 240 pF, 2V rise).
4. Applies to all non-shared registers excluding the Peek/Poke Data registers. LRDY will remain active.
5. Applies to the Peek/Poke Data and Shared Registers. LRDY will be taken inactive as soon as LA<1:0> and LCS# are valid, and remain inactive until valid data is available, or has been written. The deassertion of the local read strobe (LRD#) or local write strobe (LWR#) indicates the end of the current shared register or peek/poke data register access. If the local chip select (LCS#) input remains asserted and the local address selects remain low (LADS<1:0>) after LRD# or LWR# deasserts, a new shared register or peek/poke data register cycle begins. Under these conditions, the LRDY output will become inactive again (driven low) within the time specified by t95.
6. The maximum LRDY delay, 3.5 (t1) + 60 ns from LRD# or LWR#, only occurs if the local processor access loses the internal register access arbitration to an EISA access and if the following BCLK cycle is stretched. Without BCLK stretching, the maximum delay is 2.5 (t1) +60 ns. The minimum LRDY delay is 1.5 (t1). **NOTE:** The maximum BCLK stretch that will be seen by the BMIC is one BCLK period; this is assuming that the bus controller is the 82358 (EBC). If the 82358 is not used as the bus controller, the LRDY and data delay max. specs (t96/t85) will not necessarily be valid.
7. Exiting master mode, the address lines <31:2>, M/IO, LOCK# START#, IBE#<3:0>, MSBURST#, IDAT<31:0>, and W/R will float no later than the falling edge of BCLK after CMD# is deasserted.
8. During a mismatched cycle START#, IBE#<3:0>, and IDAT<31:0> will float from the first falling edge of BCLK after START# is negated.
9. Includes mismatched cycles.
10. Refers to the upper 22 EISA address lines which are multiplexed into the upper 22 data lines IDAT<31:10>. The address will be available for latching into the external address latches 45 ns from the falling edge of BCLK.
11. The TDIR and TCHAN signals are referenced to the falling edge of TRQ# during the cycles that TLD# is not requested.
12. LRDY going active will always be delayed from data valid. The maximum delay seen will be no greater than one (t1) period.
13. Characterized, not tested.
14. Under non-preempt, MREQ# will deassert a minimum of 0.5 BCLKs after the negating edge of the last CMD# of the transfer, depending on the cycle type (refer to the Basic Function Timings, Section 10.0).
15. During an EISA read transfer, the BMIC will assert TEOP# typically eight TCLKs after CMD# is deasserted from the last EISA cycle, indicating end of transfer (refer to the Basic Function Timings, Section 10.0).
16. During an EISA write transfer, the BMIC will assert TEOP# two TCLKs after CMD# is deasserted, indicating end of transfer (refer to the Basic Function Timings, Section 10.0).
17. For address changes while CMD# is active.
18. During an upper address load cycle, at the beginning of a transfer sequence, CMD# is inactive.
19. For "Downshifting Cases" where the transfer is misaligned.

12.2 A.C. Characteristics Waveforms

A.C. Testing Input, Output Waveforms



290255-12

NOTE:
The input waveforms have tr < 2.0 ns from 0.8V to 2.0V

A. Output delay specification referenced from one of the following signals: BCLK, TCLK, CMD#, START#, AEN, IADS<11:2>, W/R, TDOE#, LRD#, LWR#, LADS<1:0>, LCS#, LRD#, or LWR#.

B. Minimum input setup specification referenced to one of the following signals: BCLK, TCLK, CMD#, LWR#, LRD#, TLD#, or TRQ#.

C. Minimum input hold specification referenced to one of the following signals: BCLK, TCLK, CMD#, LWR#, LRD#, TLD#, or TRQ#.

A.C. Testing: All inputs are driven at 4V for a logic "1" and 0V for a logic "0". A.C. Timings are measured from the 0.8V and 2.0V levels on the source signal to either the 0.8V and 2V or 1.5V level on the signal under test; except as noted by the following:

1. BCLK and TCLK high time measurements are made at 2.0V
2. BCLK and TCLK low time measurements are made at 0.8V
3. START#, CMD#, LRD#, and LWR# pulse width measurements are made at 0.8V

A.C. TEST LOADS

CL = 25 pF on IDAT<31:0>, IDOE#, IOSEL#<1:0>, TRQ#, TLD#, TEOP#, TDAT<15:0>, TCHAN, TDIR, LRDY, and LDAT<7:0>

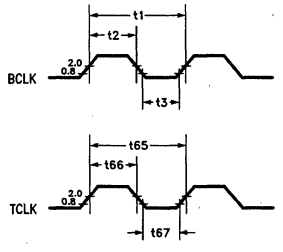
CL = 35 pF on IDDIR

CL = 50 pF on UALOE# and LINT

CL = 120 pF on MREQ# and EINT

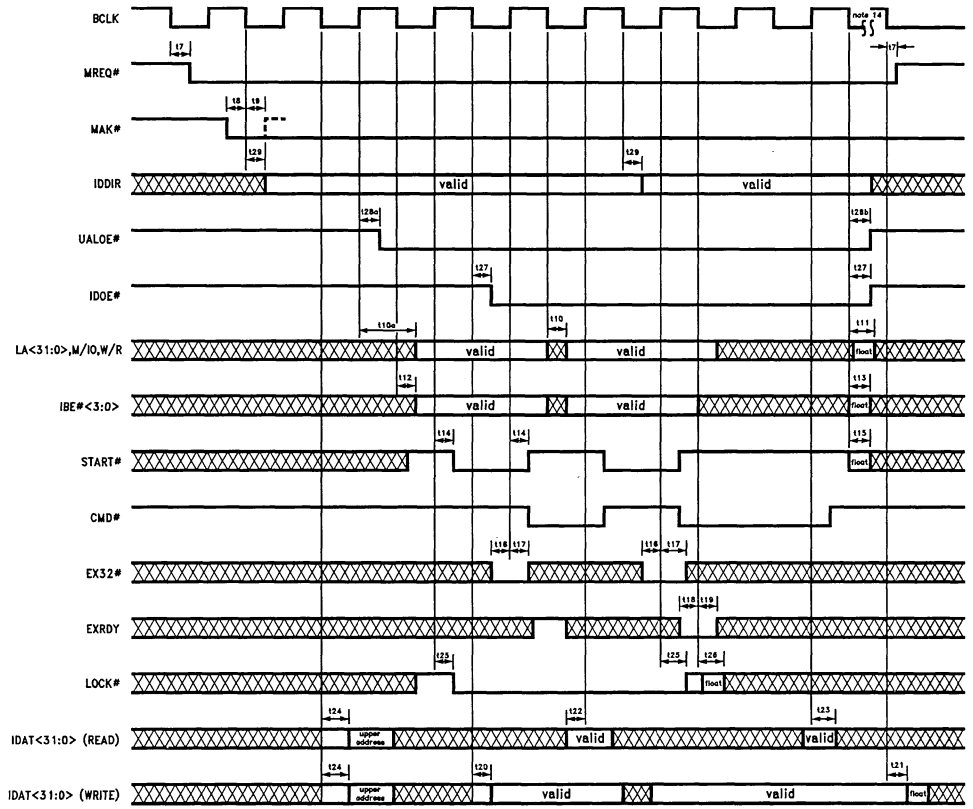
CL = 240 pF on IADS<9:2>, BE#<3:0>, W/R, START#, EX32#, LOCK, MSBURST#, MASTER16#, EXRDY, and M/IO

BCLK, TCLK Timing



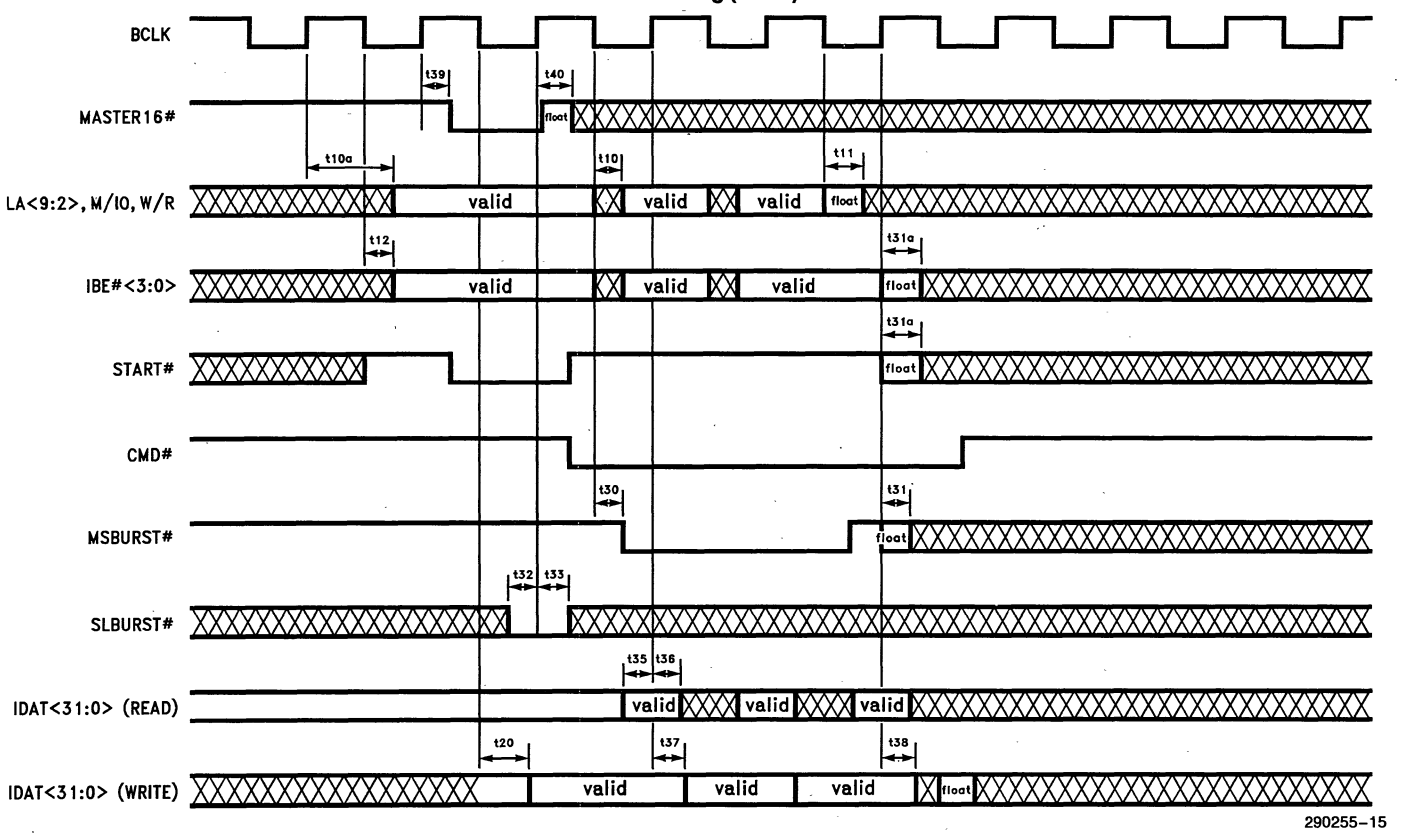
290255-13

Master Timing
(Includes: all Cycle Types—Initial Burst, Non-Burst, Peek/Poke, and Mismatched)

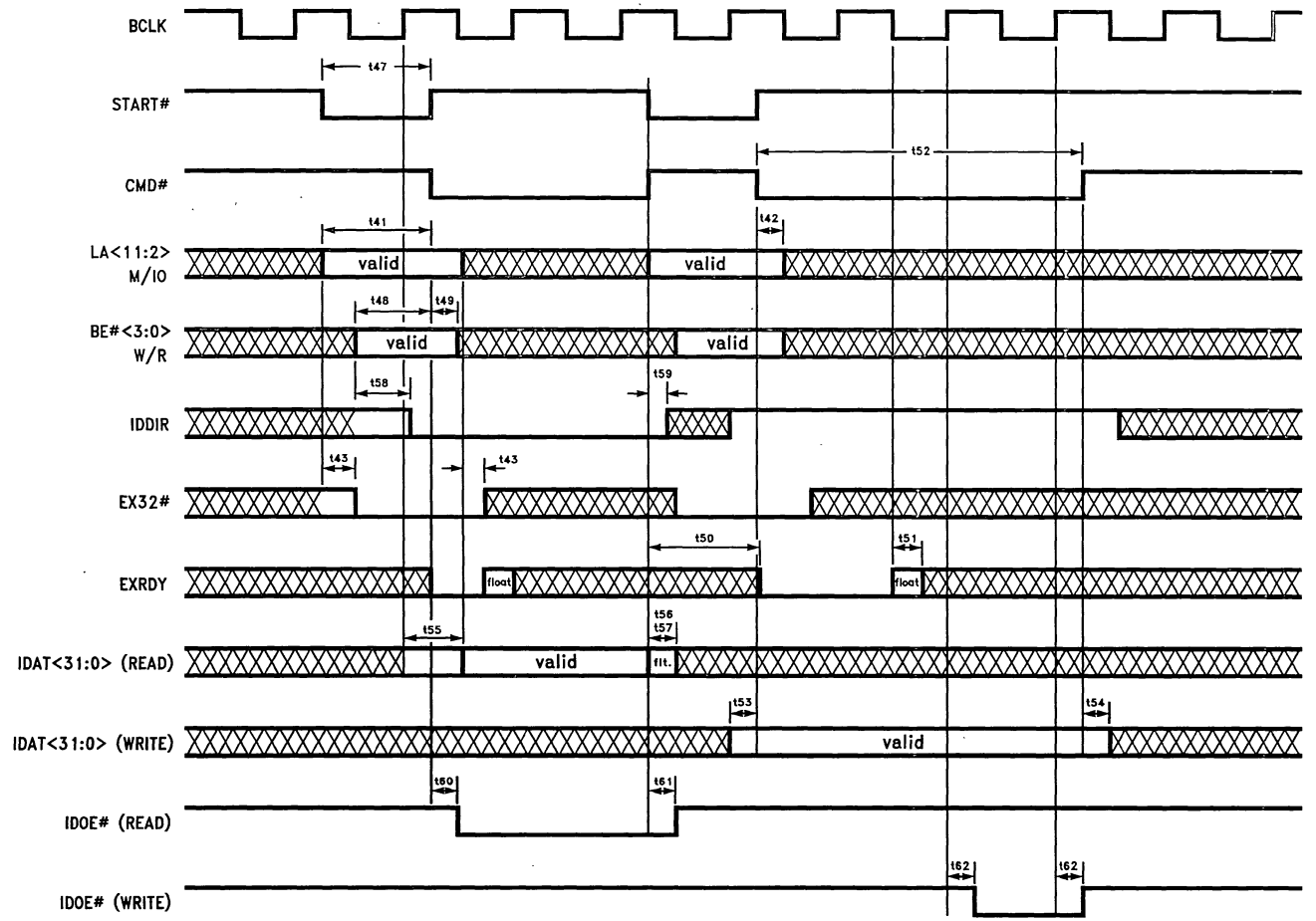


290255-14

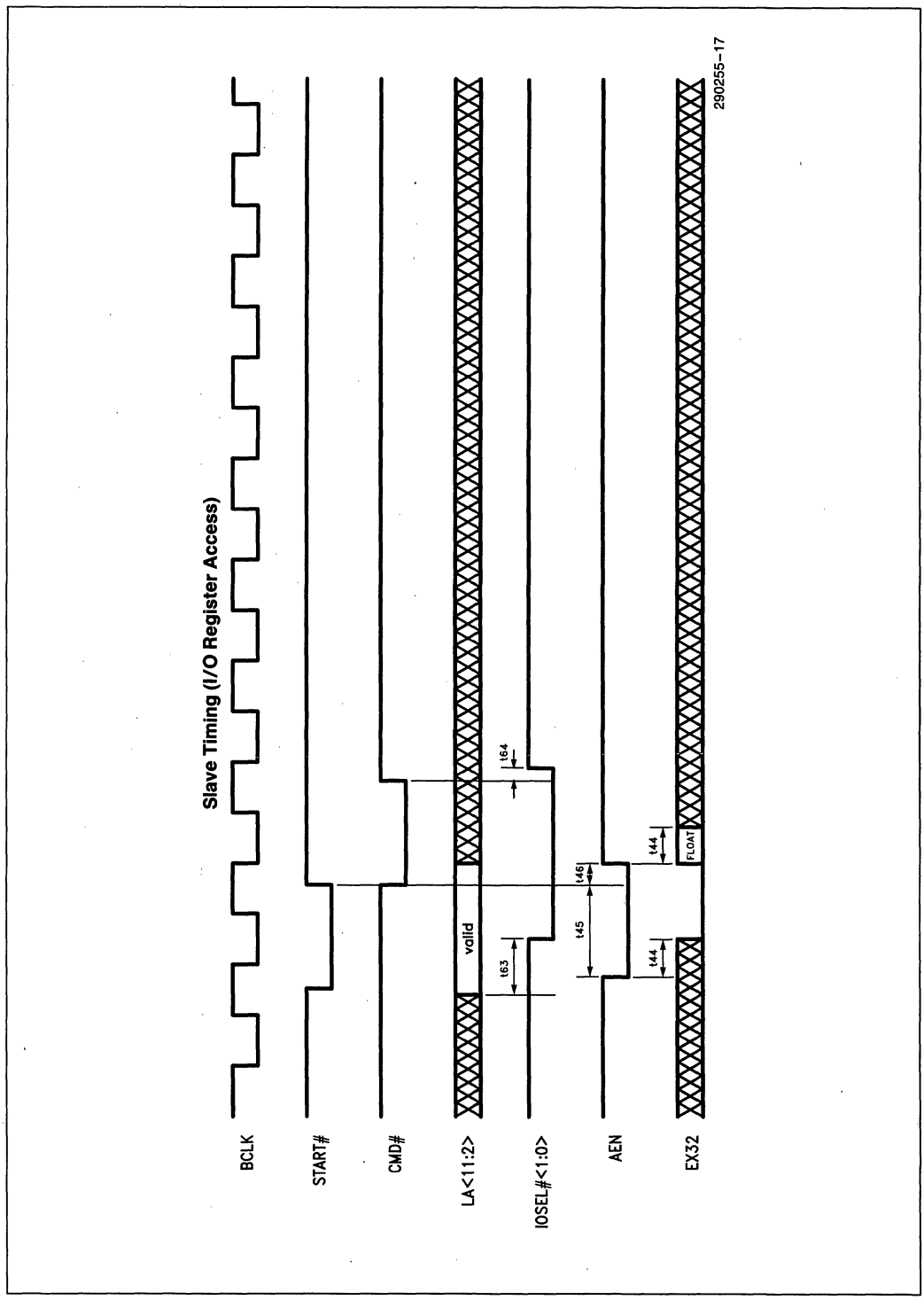
Master Timing (Burst)



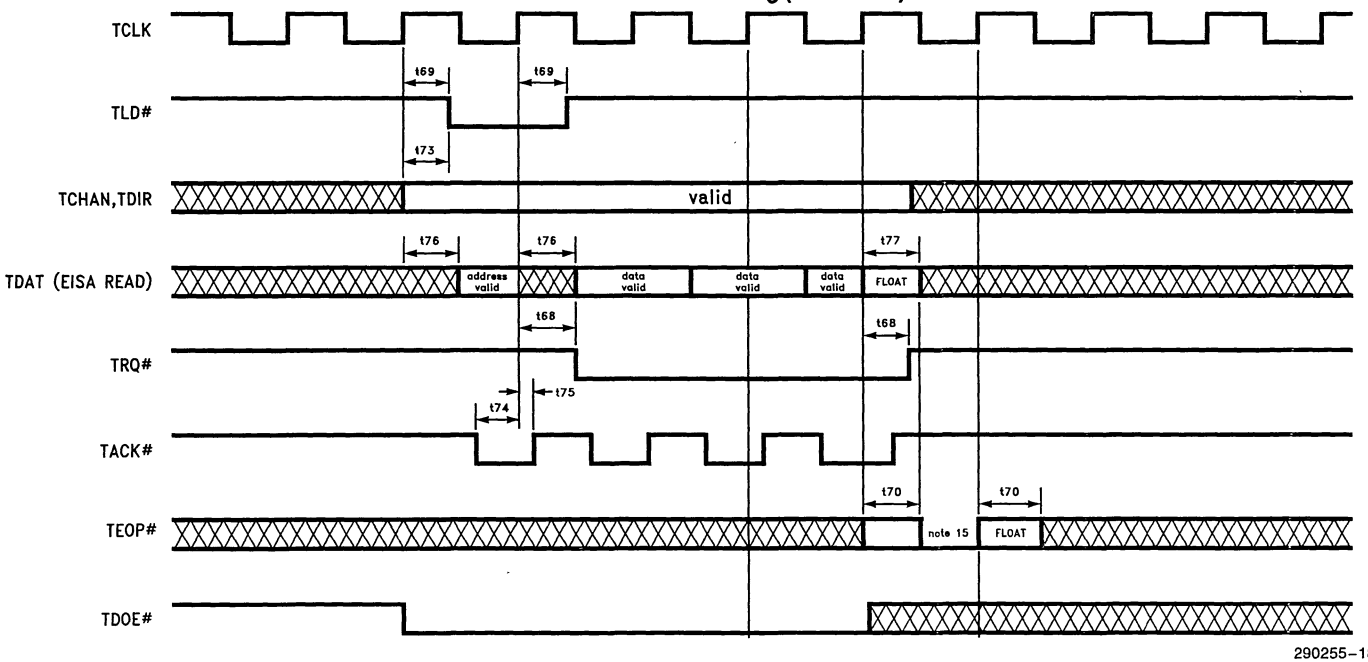
Slave Timing (Shared Register Access)



290255-16

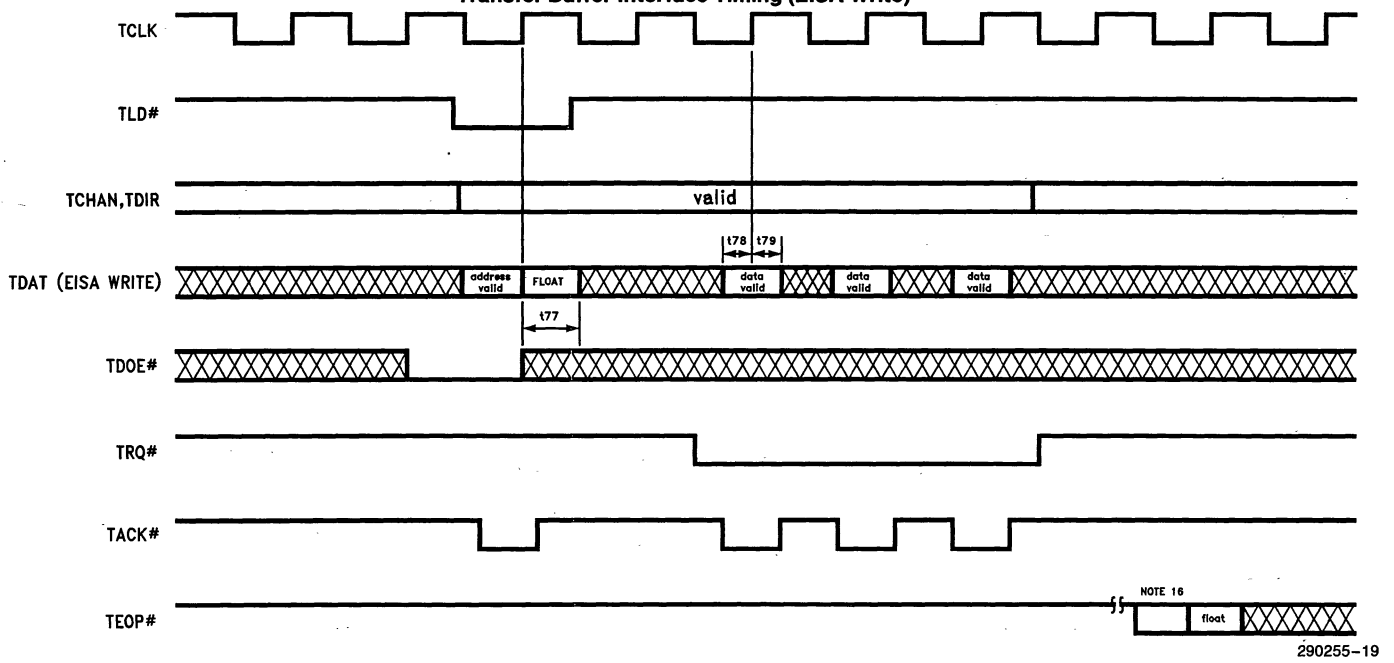

Slave Timing (I/O Register Access)
BCLK
START#
CMD#
LA<11:2>
IOSEL#<1:0>
AEN
EX32
valid
FLOAT
t63
t64
t45
t46
t44
t44
290255-17

Transfer Buffer Interface Timing (EISA Read)

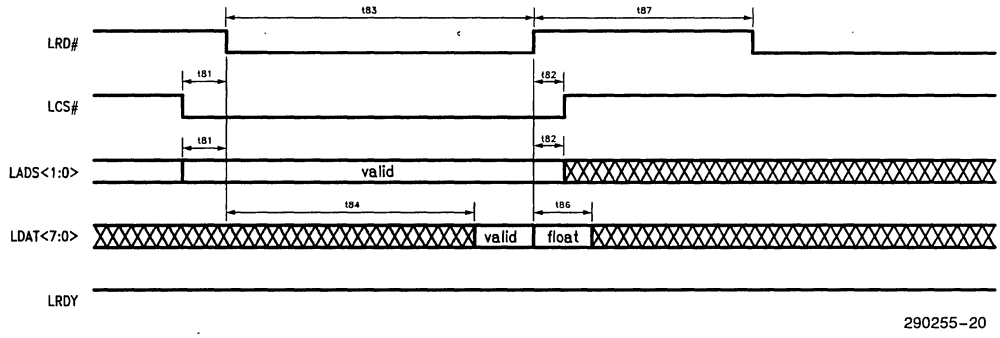


290255-18

Transfer Buffer Interface Timing (EISA Write)

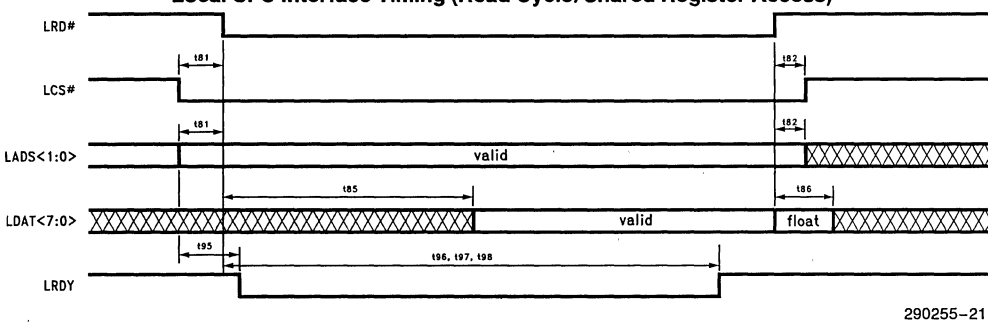


Local CPU Interface Timing (Read Cycle/Non-Shared Register Access)

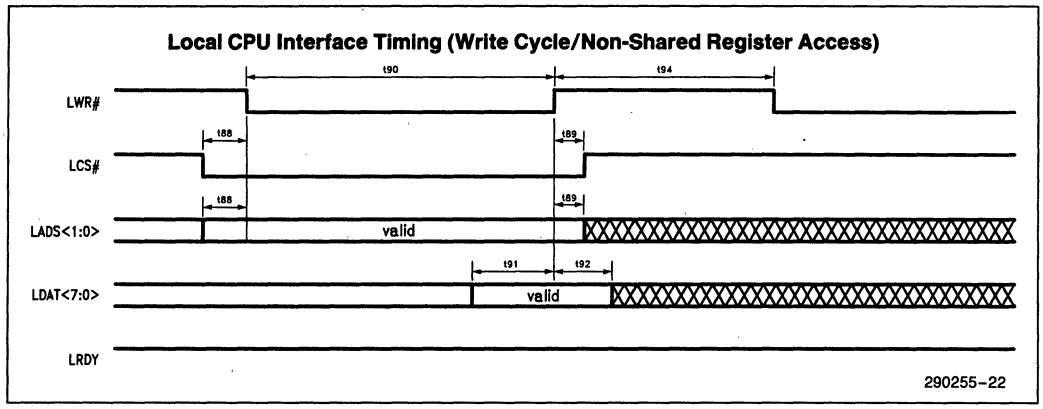


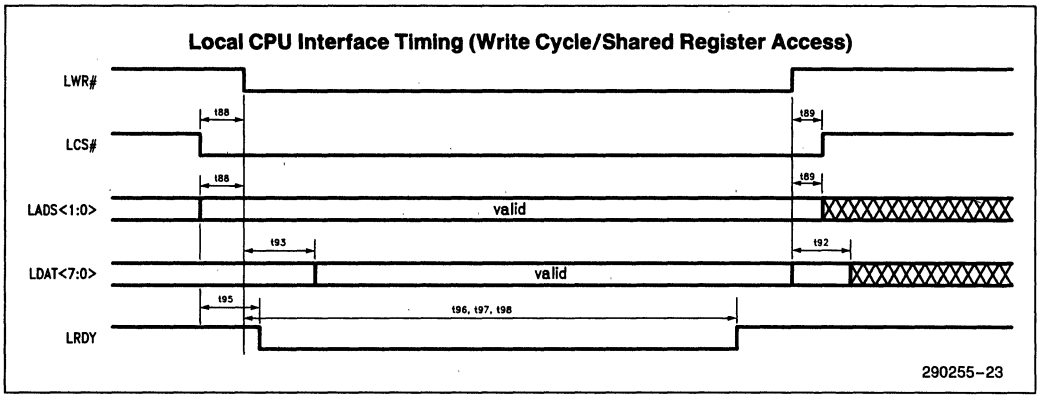
290255-20

Local CPU Interface Timing (Read Cycle/Shared Register Access)



290255-21


Local CPU Interface Timing (Write Cycle/Non-Shared Register Access)
t90
t94
LWR#
t88
t89
LCS#
t88
t89
LADS<1:0>
valid
t91
t92
LDAT<7:0>
valid
LRDY
290255-22


Local CPU Interface Timing (Write Cycle/Shared Register Access)
LWR#
t88
t89
LCS#
t88
t89
LADS<1:0>
valid
t93
t92
LDAT<7:0>
valid
t95
t96, t97, t98
LRDY
290255-23

13.0 BMIC PN AND PACKAGE INFORMATION

13.1 Signal Overview

Name = Pin Name, Type = I—Input, O—Output, OC—Open Collector, B—Both Input and Output, BC—Both and Open Collector, Pin = Pin Location

Name	Type	Pin	Description
EISA BUS INTERFACE SIGNALS			
START#	B	84	EISA Start of Cycle
CMD#	I	102	EISA Command Strobe
M/IO	B	81	EISA Memory/IO Cycle Status Signal
W/R	B	80	EISA Write/Read Status Signal
EXRDY	I, OC	103	EISA Ready Signal
EX32#	I, OC	104	EISA 32-Bit Slave Response Signal
MASTER16#	OC	82	EISA 16-Bit Master Control Signal
IBE# <3:0>	B	64, 61, 60 59	EISA Byte Enable Lines
AEN	I	107	EISA Address Enable Signal
MSBURST#	O	96	EISA Master Burst Signal
SLBURST#	I	97	EISA Slave Burst Signal
LOCK#	O	98	EISA Resource Lock Signal
MREQ#	O	99	EISA Bus Master Request Signal
MAK#	I	100	EISA Master Bus Acknowledge Signal
EINT	OC	109	EISA Interrupt Request Signal
BCLK	I	101	EISA Bus Clock
RESET	I	125	EISA Reset Signal
IDAT<31:0>	B	Section	EISA Data Lines
IADS<11:10>	I	105, 106	EISA Address Input Lines
IADS<9:2>	B	57–55, 53, 44, 40–38	EISA Lower Address Lines
EISA BUFFER CONTROL SIGNALS			
UALOE#	O	78	EISA Upper Address Latch and Output Enable
IDDIR	O	79	EISA Data Buffer Direction Signal
IDOE23#	O	75	EISA Data Byte Line Buffer Enable (Bytes 3, 2)
IDOE# <1:0>	O	76, 77	EISA Data Byte Line Buffer Enables (Bytes 1, 0)
TRANSFER BUFFER INTERFACE SIGNALS			
TCLK	I	32	Transfer Clock
TRQ#	O	7	Transfer Data Request Signal
TACK#	I	6	Transfer Data Acknowledge Signal
TDIR	O	3	Transfer Data Direction Signal
TCHAN	O	4	Transfer Data Channel Select Signal
TLD#	O	5	Transfer Address Counter Load Signal
TDOE#	I	2	Transfer Data Bus Output Enable
TEOP#	I, OC	1	Transfer End-of-Process
TDAT<15:0>	B	Section	Transfer Data Bus Lines

13.1 Signal Overview (Continued)

Name = Pin Name, Type = I—Input, O—Output, OC—Open Collector, B—Both Input and Output, BC—Both and Open Collector, Pin = Pin Location

Name	Type	Pin	Description
LOCAL PROCESSOR INTERFACE SIGNALS			
LRD#	I	130	Local Read Signal
LWR#	I	129	Local Write Signal
LCS#	I	128	Local Chip Select Signal
LDAT<7:0>	B	121–118 115–112	Local Data Bus Lines
LADS<1:0>	I	127, 126	Local Address Register Select Signals
LRDY#	B	122	Local Ready Signal
LINT#	O	123	Local Processor Interrupt Signal
MISCELLANEOUS SIGNALS			
IOSEL#<1:0>	O	111, 110	Expansion Board Address Range Decode Signals
POWER PINS			
V_{CC}		108, 124	Power Pins for the Internal Logic
V_{SS}		42, 58	Ground Pins for the Internal Logic
V_{CCB}		12, 23, 41, 63, 74, 83, 94, 117, 132	Power Pins for the Output Buffers
V_{SSB}		13, 22, 33, 43, 54, 62, 73, 85, 95, 116, 131	Ground Pins for the Output Buffers

13.2 Device Pinout

I = Input, O = Output, OC = Open Collector, B = Both Input and Output, BC = Both and Open Collector

Device Pinout—132 Lead PQFP

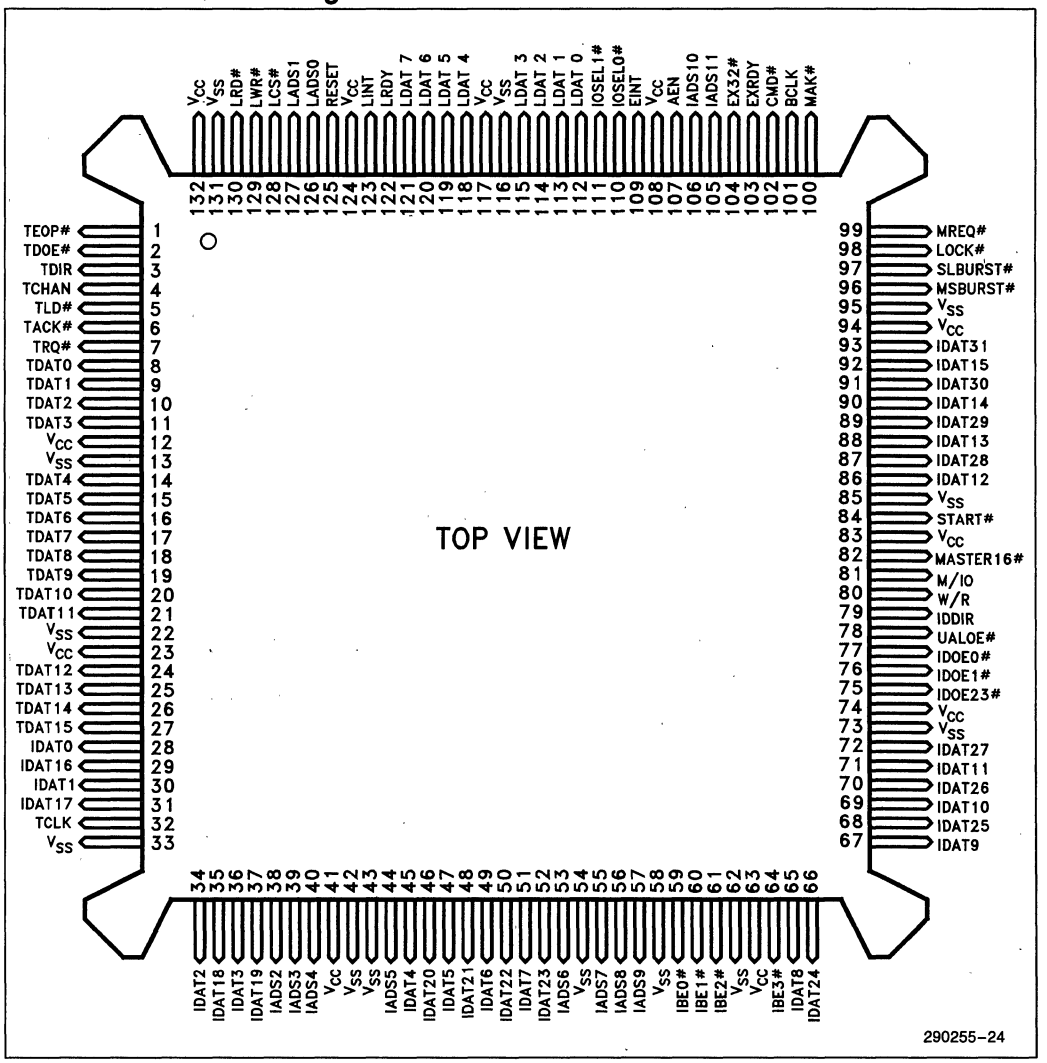
A Row		
Pin	Label	Type
1	TEOP#	I, OC
2	TDOE#	I
3	TDIR	O
4	TCHAN	O
5	TLD#	O
6	TACK#	I
7	TRQ#	O
8	TDAT0	B
9	TDAT1	B
10	TDAT2	B
11	TDAT3	B
12	V_{CCB}	
13	V_{SSB}	
14	TDAT4	B
15	TDAT5	B
16	TDAT6	B
17	TDAT7	B
18	TDAT8	B
19	TDAT9	B
20	TDAT10	B
21	TDAT11	B
22	V_{SSB}	
23	V_{CCB}	
24	TDAT12	B
25	TDAT13	B
26	TDAT14	B
27	TDAT15	B
28	IDAT0	B
29	IDAT16	B
30	IDAT1	B
31	IDAT17	B
32	TCLK	I
33	V_{SSB}	

B Row		
Pin	Label	Type
34	IDAT2	B
35	IDAT18	B
36	IDAT3	B
37	IDAT19	B
38	IADS2	B
39	IADS3	B
40	IADS4	B
41	V_{CCB}	
42	V_{SS}	
43	V_{SSB}	
44	IADS5	B
45	IDAT4	B
46	IDAT20	B
47	IDAT5	B
48	IDAT21	B
49	IDAT6	B
50	IDAT22	B
51	IDAT7	B
52	IDAT23	B
53	IADS6	B
54	V_{SSB}	
55	IADS7	B
56	IADS8	B
57	IADS9	B
58	V_{SS}	
59	IBE0#	B
60	IBE1#	B
61	IBE2#	B
62	V_{SSB}	
63	V_{CCB}	
64	IBE3#	B
65	IDAT8	B
66	IDAT24	B

C Row		
Pin	Label	Type
67	IDAT9	B
68	IDAT25	B
69	IDAT10	B
70	IDAT26	B
71	IDAT11	B
72	IDAT27	B
73	V_{SSB}	
74	V_{CCB}	
75	IDOE23#	O
76	IDOE1#	O
77	IDOE0#	O
78	UALOE#	O
79	IDDIR	O
80	W/R	B
81	M/IO	B
82	MASTER16#	OC
83	V_{CCB}	
84	START#	B
85	V_{SSB}	
86	IDAT12	B
87	IDAT28	B
88	IDAT13	B
89	IDAT29	B
90	IDAT14	B
91	IDAT30	B
92	IDAT15	B
93	IDAT31	B
94	V_{CCB}	
95	V_{SSB}	
96	MSBURST#	O
97	SLBURST#	I
98	LOCK#	O
99	MREQ#	O

D Row		
Pin	Label	Type
100	MAK#	I
101	BCLK	I
102	CMD#	I
103	EXRDY	I, OC
104	EX32#	I, OC
105	IADS11	I
106	IADS10	I
107	AEN	I
108	V_{CC}	
109	EINT	OC
110	IOSEL0#	O
111	IOSEL1#	O
112	LDAT0	B
113	LDAT1	B
114	LDAT2	B
115	LDAT3	B
116	V_{SSB}	
117	V_{CCB}	
118	LDAT4	B
119	LDAT5	B
120	LDAT6	B
121	LDAT7	B
122	LRDY	B
123	LINT	O
124	V_{CC}	
125	RESET	I
126	LADS0	I
127	LADS1	I
128	LCS#	I
129	LWR#	I
130	LRD#	I
131	V_{SSB}	
132	V_{CCB}	

13.3 132-Pin PQFP Package Pinout



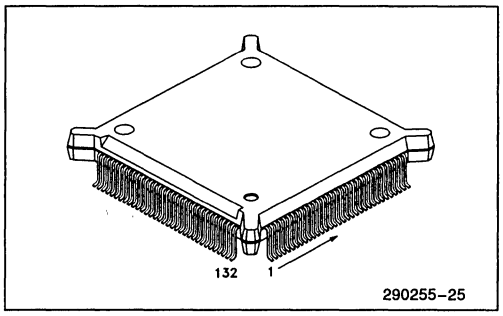
PACKAGING INFORMATION

(See Packaging Specification Order # 240800, Package Type NG)

Introduction

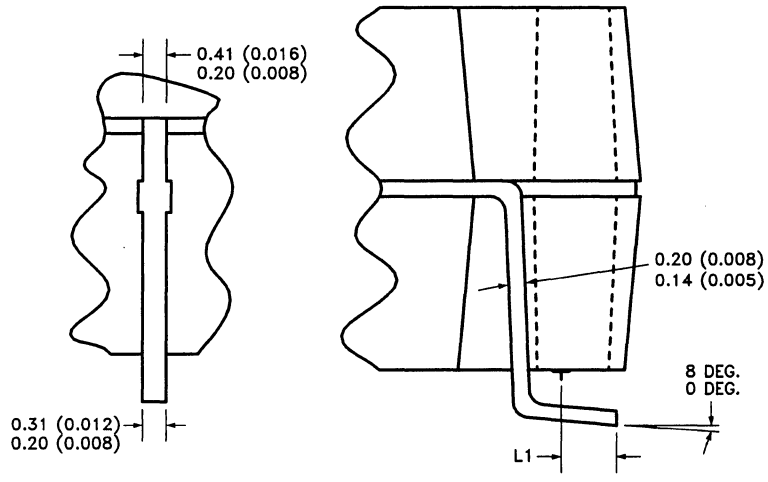
The individual components of Intel's EISA Chip Set come in JEDEC standard Gull Wing packages (25 MIL pitch), with "bumpers" on the corners for ease of handling. Please refer to the accompanying table for the package associated with each device, and to the individual component specifications for pinouts. (Note that the individual pinouts are numbered consistently with the numbering scheme depicted in the accompanying figures.)

PLASTIC QUAD FLAT PACK (PQFP)



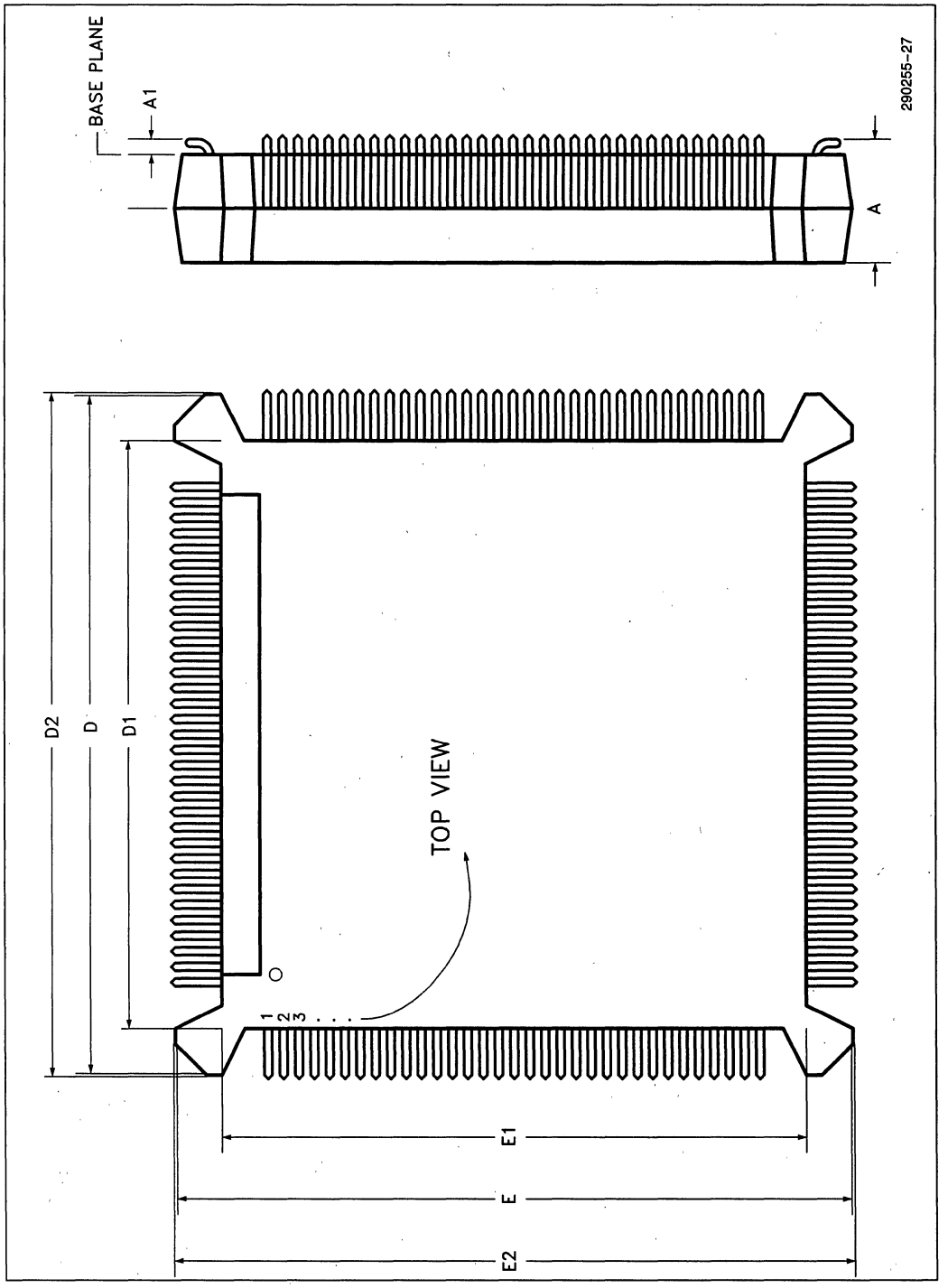
TYPICAL LEAD

Case Outline Drawings
Plastic Fine Pitch Chip Carrier
0.84 mm Pitch

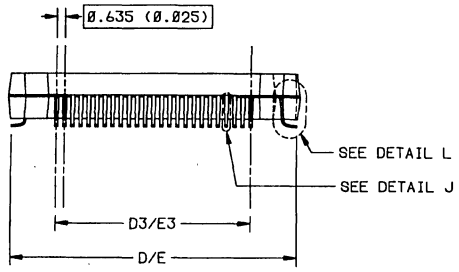


Symbol	Description	Inch		mm	
		Min	Max	Min	Max
N	Lead Count	132		132	
A	Package Height	0.160	0.170	4.06	4.32
A1	Standoff	0.020	0.030	0.51	0.76
D, E	Terminal Dimension	1.075	1.085	27.31	27.56
D1, E1	Package Body	0.947	0.953	24.05	24.21
D2, E2	Bumper Distance	1.097	1.103	27.86	28.02
D3, E3	Lead Dimension	0.800 Ref		20.32 Ref	
L1	Foot Length	0.020	0.030	0.51	0.76

13.4 PRINCIPAL DIMENSIONS & DATUMS



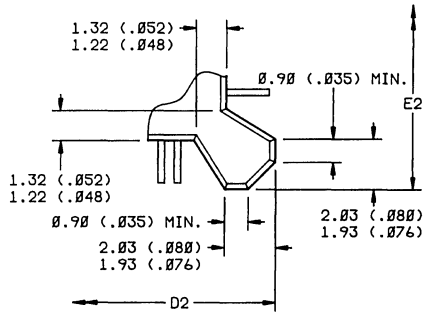
TERMINAL DETAILS



mm (inch)

290255-28

BUMPER DETAIL



mm (inch)

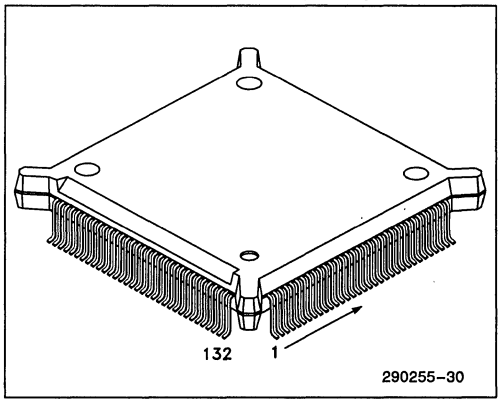
290255-29

13.5 Package Thermal Specification

The 82355 (BMIC) is specified for operation when the case temperature is within the range of 0°C–70°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 in the figure below.

PLASTIC QUAD FLAT PACK (PQFP)



82355 PQFP Package Thermal Characteristics

Thermal Resistance—°C/W							
Parameter	Air Flow Rate (ft/min)						
	0	50	100	200	400	600	800
θ Junction—Case	7	7	7	7	7	7	7
θ Case to Ambient	22	21	19.5	17.5	14.5	12	10

NOTES:
1. Table applies to 82355 PQFP plugged into socket or soldered directly into board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$.

Process Name:

1.2μ CHMOS III P-well

I_{CC} at Hot with no Resistive Loads:

150 mA max at 70°C
Measure PQFP case temperature
at center of top surface

14.0 BMIC REGISTER ADDRESS MAP

14.1 Index Register Set

The following registers are mapped directly into the local processor interface:

Local Address	Type	Register Description
0	R/W	Local Data Register
1	R/W	Local Index Register
2	R/W	Local Status/Control Register
3	—	Reserved

14.2 Shared Register Set

EISA Address	Type	Index Address	Type	Register Description
XC80	R	00	R/W	ID Byte 0
XC81	R	01	R/W	ID Byte 1
XC82	R	02	R/W	ID Byte 2
XC83	R	03	R/W	ID Byte 3
XC84	—	04	—	Non BMIC Register (For Expansion Board Use)
XC85	—	05	—	Non BMIC Register (For Expansion Board Use)
XC86	—	06	—	Non BMIC Register (For Expansion Board Use)
XC87	—	07	—	Non BMIC Register (For Expansion Board Use)
XC88	R	08	R/W	Global Configuration Register
XC89	R/W	09	R	System Interrupt Enable/Control Register
XC8A	R/W	0A	R/W	Semaphore Port 0
XC8B	R/W	0B	R/W	Semaphore Port 1
XC8C	R	0C	R/W	Local Doorbell Enable Register
XC8D	R/W	0D	R/W	Local Doorbell Interrupt/Status Register
XC8E	R/W	0E	R	EISA System Doorbell Enable Register
XC8F	R/W	0F	R/W	EISA System Doorbell Interrupt/Status Register
XC90	R/W	10	R/W	Mailbox Register (1)
XC91	R/W	11	R/W	Mailbox Register (2)
XC92	R/W	12	R/W	Mailbox Register (3)
XC93	R/W	13	R/W	Mailbox Register (4)
XC94	R/W	14	R/W	Mailbox Register (5)
XC95	R/W	15	R/W	Mailbox Register (6)
XC96	R/W	16	R/W	Mailbox Register (7)
XC97	R/W	17	R/W	Mailbox Register (8)
XC98	R/W	18	R/W	Mailbox Register (9)
XC99	R/W	19	R/W	Mailbox Register (10)
XC9A	R/W	1A	R/W	Mailbox Register (11)
XC9B	R/W	1B	R/W	Mailbox Register (12)
XC9C	R/W	1C	R/W	Mailbox Register (13)
XC9D	R/W	1D	R/W	Mailbox Register (14)
XC9E	R/W	1E	R/W	Mailbox Register (15)
XC9F	R/W	1F	R/W	Mailbox Register (16)
XCA0–XCAF	R/W	20–2F	—	Reserved

14.3 Processor Only Register Set

Index Address	Type	Register Description
30	R/W	Peek/Poke Data Register Byte 0
31	R/W	Peek/Poke Data Register Byte 1
32	R/W	Peek/Poke Data Register Byte 2
33	R/W	Peek/Poke Data Register Byte 3
34	R/W	Peek/Poke Address Register Byte 0
35	R/W	Peek/Poke Address Register Byte 1
36	R/W	Peek/Poke Address Register Byte 2
37	R/W	Peek/Poke Address Register Byte 3
38	R/W	Peek/Poke Control Register
39	R/W	I/O Decode Range 0 Base Address Register
3A	R/W	I/O Decode Range 0 Control Register
3B	R/W	I/O Decode Range 1 Base Address Register
3C	R/W	I/O Decode Range 1 Control Register
3D	—	Reserved
3E	—	Reserved
3F	—	Reserved
40	R/W	Channel 0 Base Count Register Byte 0
41	R/W	Channel 0 Base Count Register Byte 1
42	R/W	Channel 0 Base Count Register Byte 2
43	R/W	Channel 0 Base Address Register Byte 0
44	R/W	Channel 0 Base Address Register Byte 1
45	R/W	Channel 0 Base Address Register Byte 2
46	R/W	Channel 0 Base Address Register Byte 3
47	—	Reserved
48	R/W	Channel 0 Configuration Register
49	W	Channel 0 Transfer Strobe Register
4A	R/W	Channel 0 Status Register
4B	R/W	Channel 0 TBI Base Address Register Byte 0
4C	R/W	Channel 0 TBI Base Address Register Byte 1
4D	—	Reserved
4E	—	Reserved
4F	—	Reserved
50	R	Channel 0 Current Count Register Byte 0
51	R	Channel 0 Current Count Register Byte 1
52	R	Channel 0 Current Count Register Byte 2
53	R	Channel 0 Current Address Register Byte 0
54	R	Channel 0 Current Address Register Byte 1
55	R	Channel 0 Current Address Register Byte 2
56	R	Channel 0 Current Address Register Byte 3
57	—	Reserved
58	R	Channel 0 TBI Current Address Register Byte 0
59	R	Channel 0 TBI Current Address Register Byte 1
5A	—	Reserved
5B	—	Reserved
5C	—	Reserved
5D	—	Reserved
5E	—	Reserved
5F	—	Reserved

14.3 Processor Only Register Set (Continued)

Index Address	Type	Register Description
60	R/W	Channel 1 Base Count Register Byte 0
61	R/W	Channel 1 Base Count Register Byte 1
62	R/W	Channel 1 Base Count Register Byte 2
63	R/W	Channel 1 Base Address Register Byte 0
64	R/W	Channel 1 Base Address Register Byte 1
65	R/W	Channel 1 Base Address Register Byte 2
66	R/W	Channel 1 Base Address Register Byte 3
67	—	Reserved
68	R/W	Channel 1 Configuration Register
69	W	Channel 1 Transfer Strobe Register
6A	R/W	Channel 1 Status Register
6B	R/W	Channel 1 TBI Base Address Register Byte 0
6C	R/W	Channel 1 TBI Base Address Register Byte 1
6D	—	Reserved
6E	—	Reserved
6F	—	Reserved
70	R	Channel 1 Current Count Register Byte 0
71	R	Channel 1 Current Count Register Byte 1
72	R	Channel 1 Current Count Register Byte 2
73	R	Channel 1 Current Address Register Byte 0
74	R	Channel 1 Current Address Register Byte 1
75	R	Channel 1 Current Address Register Byte 2
76	R	Channel 1 Current Address Register Byte 3
77	—	Reserved
78	R	Channel 1 TBI Current Address Register Byte 0
79	R	Channel 1 TBI Current Address Register Byte 1
7A	—	Reserved
7B	—	Reserved
7C	—	Reserved
7D	—	Reserved
7E	—	Reserved
7F	—	Reserved

NOTES:
1. TBI = Transfer Buffer Interface
2. X = Slot number
3. All the reserved locations, when read, will return a value of no practical use to the user.
4. The "non BMIC" register locations (XC84h–XC87h & 04h–07h) are locations to be used by registers implemented externally on the expansion board. The BMIC will not respond to these locations (XC84h–CC87h) when accessed from the EISA side. However, the BMIC can be programmed to support the decode of the EISA addresses (XC84h–XC87h) through its I/O decode register set (refer to Section 4.8). All "non BMIC" register locations (04h–07h) when read from the local side, will return a value of no practical use to the user.

82355 Revision Summary

The following changes have been made since revision 006:

Section 1.1 EISA READ definitions has been changed from "A data transfer (burst, non-burst (two BCLK), or mismatched) from system to the expansion board across one of the two transfer channels" to "A data transfer (burst, non-burst (two BCLK), or mismatched) from system to the expansion board across Channel 1 transfer channel 1."

Section 4.2 New paragraph added after the first paragraph:

Channel 0 can be used for EISA READ operation only. Channel 1 can be used for both EISA READ and EISA WRITE operations.

Section 8.1.2 Bits 7–4 has been changed to Third hex digit of product number

Bits 4–0 has been changed to Hexadecimal digit of product revision

Section 8.2.3.3 Sentence added to Bit 22:

This is applicable only to channel 1 and not for channel 0, as channel 0 can perform EISA WRITE transfers only.

Section 10.0 Figure 10-1 TDIR, TCHAN timing diagram has changed from 10 to 11.

Figure 10-3 TDIR, TCHAN timing diagram has changed from 10 to 11.

Figure 10-7 TDIR, TCHAN timing diagram has changed from 10 to 11.

The following changes have been made since revision 005:

Section 4.3 New paragraph added at the end. This paragraph reads:

Any consecutive Peek/Poke or Locked exchanged transfers must be initiated only after the previous Peek/Poke or Locked exchange has been completed. This can be accomplished by making sure that bit 2 of the local status/control register is set to a zero before initiating the transfer.

Section 4.8 New paragraph added after the third paragraph. This paragraph reads:

The IDOEs do not go active during an IOSEL cycle outside the shared register access space.

Section 7.4 The third paragraph had two sentences deleted that is replaced with Figure 7-2, IDOEO# Connection during ID Register Access. The sentences that were deleted read as follows:

The external lines connected to the IDAT <7:0> lines should be connected to the bus between the BMIC and the external F245 data buffers. The BMIC will enable the external data buffers to drive byte lane 0 of the EISA bus upon detection of the ID address.

The following changes have been made since revision 004:

Section 4.3 New paragraph added at the end. This paragraph reads:

Whenever the BMIC is commanded to do an EISA POKE cycle, the BMIC will assert the MREQ# signal low normally, transfer up to four bytes of data, and release the bus by de-asserting MREQ# high. A potential problem exists, however, when the slave device extends the cycle by de-asserting EXRDY low. If the slave holds this signal low past the time that the BMIC is forced to release MREQ# high (it has been preempted while waiting for the slave to assert EXRDY high), then the BMIC will drive MREQ# back low again immediately after this cycle ends if there is another transfer pending (TBI, PEEK, POKE or LOCKED-EXCHANGE). Note that according to the EISA spec, "A bus master must wait at least two BCLKs after releasing the bus before re-asserting its MREQx*"(EISA spec, MREQ* signal description). To adhere to EISA specifications, it is required that LOCKED-EXCHANGE cycles be used in lieu of POKE cycles.

Section 8.2.3.4 New sentence added at the end of paragraph one:

At the end of a transfer, this register contains the value of the number of bytes transferred during the last cycle.

Section 8.2.4.2 Two paragraphs added for the CFGCL bit:

Before a channel is issued a clear command, the channel must first be suspended by writing a "1" into Bit 0 (CFGSU) of the Transfer Channel Configuration Register. Next the Transfer Channel Status Register must be read. If the TSTTC (Bit 0) is set to a "1", then the channel has already completed the transfer. The channel is then unsuspended (write a "0" into Bit 0 [CFGSU] of the Transfer Channel Configuration Register), and the TSTTC bit is then cleared.

If the TSTTC bit is a "0", then the TSTEN bit is checked. If this bit is a "1", then the channel has not returned to idle yet, and the Transfer Channel Status Register is repolled. If the TSTEN bit is a "0", then the channel has successfully returned to idle and can now be cleared with no errors. This is done by setting Bit 2 (CFGCL bit) to a 1 in the Transfer Channel Configuration Register. A flowchart for this operation is shown in the following figure.

Figure 8-1 was added to the data sheet.

Section 8.2.4.3 Bit 4 has been changed to reserved.

First paragraph under bit description has been changed to read, "Bits (7), (6), TST1K, and (4) are reserved. Any data read from these bits should be ignored".

The paragraph following this one has been deleted.

Section 11.2 Max Limits on symbols C_{OUT} and C_{CLK} have been changed.

Note 1 has been corrected.

Section 12.1 Symbol t1 has been corrected to read 2500 instead of 250 for max.

Symbol t20 has been corrected to read 3 for min.

Symbol t22 has been corrected to 4 from 7 for min.

Symbol t35 has been corrected to 5 from 7 for min.

Symbol t37 has been corrected to 3 for min.

Symbol t40 has been corrected to 1 for min.

Symbol t50 has been corrected to 124 from 125 for max.

Symbol t51 has been corrected to 1 for min.

Section 12.2 The Local CPU Interface Timing (Read Cycle/Shared Register Access) table has been corrected. The signal LDAT<7:0> has changed one portion from valid to float.

intel®

2

Floppy Disk Controllers

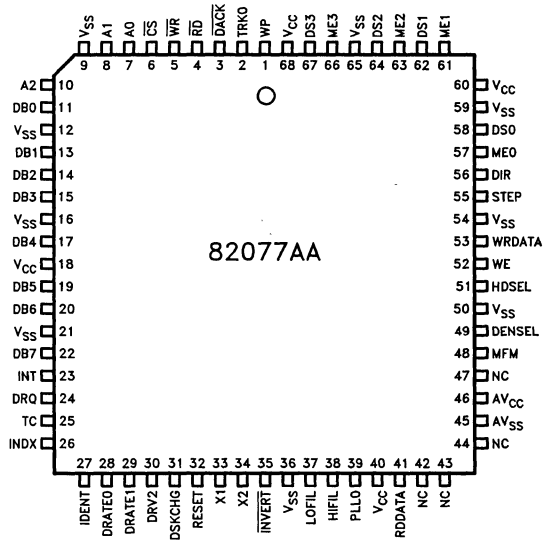
82077AA
CHMOS SINGLE-CHIP FLOPPY DISK CONTROLLER

- **Single-Chip Floppy Disk Solution**
  - **— 100% PC AT* Compatible**
  - **— 100% PS/2* Compatible**
  - **— 100% PS/2 Model 30 Compatible**
  - **— Integrated Drive and Data Bus Buffers**
- **Integrated Analog Data Separator**
  - **— 250 Kbits/sec**
  - **— 300 Kbits/sec**
  - **— 500 Kbits/sec**
  - **— 1 Mbits/sec**
- **High Speed Processor Interface**
- **Perpendicular Recording Support**
- **Integrated Tape Drive Support**
- **12 mA Host Interface Drivers, 40 mA Disk Drivers**
- **Four Fully Decoded Drive Select and Motor Signals**
- **Programmable Write Precompensation Delays**
- **Addresses 256 Tracks Directly, Supports Unlimited Tracks**
- **16 Byte FIFO**
- **68-Pin PLCC**
  (See Packaging Spec., Order #240800, Package Type N)

2

The 82077AA floppy disk controller has completely integrated all of the logic required for floppy disk control. The 82077AA, a 24 MHz crystal, a resistor package and a device chip select implements a PC AT or PS/2 solution. All programmable options default to compatible values. The dual PLL data separator has better performance than most board level/discrete PLL implementations. The FIFO allows better system performance in multi-master systems (e.g. PS/2, EISA). The 82077AA is available in three versions—82077AA-5, 82077AA and 82077AA-1. 82077AA-1 has all features listed in this data sheet. It supports both tape drives and 4 Mb floppy drives. The 82077AA supports 4 Mb floppy drives and is capable of operation at all data rates through 1 Mbps. The 82077AA-5 supports 500/300/250 Kbps data rates for high and low density floppy drives.

The 82077AA is fabricated with Intel's CHMOS III technology and is available in a 68-lead PLCC (plastic) package.



290166-1

Figure 1. 82077AA Pinout

*PS/2 and PC AT are trademarks of IBM.

The complete document for this product is available from Intel's Literature Center at 1-800-548-4725.

May 1994
Order Number: 290166-007

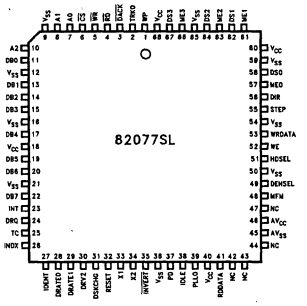
82077SL
CHMOS SINGLE-CHIP FLOPPY DISK CONTROLLER

- **Completely Compatible with Industry Standard 82077AA**
- **Single-Chip Laptop Desktop Floppy Disk Controller Solution**
  - **— 100% PC AT* Compatible**
  - **— 100% PS/2* Compatible**
  - **— 100% PS/2 Model 30 Compatible**
  - **— Fully Compatible with Intel's 386SL Microprocessor SuperSet**
  - **— Integrated Drive and Data Bus Buffers**
- **Power Management Features**
  - **— Application Software Transparency**
  - **— Programmable Powerdown Command**
  - **— Auto Powerdown and Wakeup Modes**
  - **— Two External Power Management Pins**
  - **— Typical Power Consumption in Power Down: 10 μA**
- **High Speed Processor Interface**
- **Integrated Analog Data Separator**
  - **— 250 Kbits/sec**
  - **— 300 Kbits/sec**
  - **— 500 Kbits/sec**
  - **— 1 Mbits/sec**
- **Programmable Crystal Oscillator for On or Off**
- **Integrated Tape Drive Support**
- **Perpendicular Recording Support**
- **12 mA Host Interface Drivers, 40 mA Disk Drivers**
- **Four Fully Decoded Drive Select and Motor Signals**
- **Programmable Write Precompensation Delays**
- **Addresses 256 Tracks Directly, Supports Unlimited Tracks**
- **16 Byte FIFO**
- **68-Pin PLCC**
  (See Packaging Handbook Order Number #240800, Package Type N)

The 82077SL, a 24 MHz crystal, a resistor package, and a device chip select implements a complete laptop solution. All programmable options default to 82077AA compatible values. The dual PLL data separator has better performance than most board level/discrete PLL implementations. The FIFO allows better system performance in multi-master systems (e.g., Microchannel, EISA).

The 82077SL is a superset of 82077AA. The 82077SL incorporates power management features while maintaining complete compatibility with the 82077AA/8272A floppy disk controllers. It contains programmable power management features while integrating all of the logic required for floppy disk control. The power management features are transparent to any application software. The 82077SL is available in three versions—82077SL-5, 82077SL and 82077SL-1. 82077SL-1 has all features listed in this data sheet. It supports both tape drives and 4 MB floppy drives. The 82077SL supports 4 MB floppy drives and is capable of operation at all data rates through 1 Mbps. The 82077SL-5 supports 500/300/250 Kbps data rates for high and low density floppy drives.

The 82077SL is fabricated with Intel's advanced CHMOS III technology and is available in a 68-lead PLCC (plastic) package.



290410-1

Figure 1. 82077SL Pinout

*PS/2 and PC AT are trademarks of IBM.

December 1994
Order Number: 290410-004

82077SL CHMOS Single-Chip Floppy Disk Controller

CONTENTS

CONTENTS

CONTENTS

CONTENTS

Table 1. 82077SL Pin Description

Symbol	Pin#	I/O	Description
HOST INTERFACE			
RESET	32	I	**RESET:** A high level places the 82077SL in a known idle state. All registers are cleared except those set by the Specify command.
$\overline{CS}$	6	I	**CHIP SELECT:** Decodes base address range and qualifies $\overline{RD}$ and $\overline{WR}$ inputs.
A0 A1 A2	7 8 10	I	**ADDRESS:** Selects one of the host interface registers:

A2	A1	A0	Access Type	Register	
0	0	0	R	Status Register A	SRA
0	0	1	R	Status Register B	SRB
0	1	0	R/W	Digital Output Register	DOR
0	1	1	R/W	Tape Drive Register	TDR
1	0	0	R	Main Status Register	MSR
1	0	0	W	Data Rate Select Register	DSR
1	0	1	R/W	Data (First In First Out)	FIFO
1	1	0		Reserved	
1	1	1	R	Digital Input Register	DIR
1	1	1	W	Configuration Control Register	CCR

DB0 DB1 DB2 DB3 DB4 DB5 DB6 DB7	11 13 14 15 17 19 20 22	I/O	**DATA BUS:** Data bus with 12 mA drive
$\overline{RD}$	4	I	**READ:** Control signal
$\overline{WR}$	5	I	**WRITE:** Control signal
DRQ	24	O	**DMA REQUEST:** Requests service from a DMA controller. Normally active high, but goes to high impedance in AT and Model 30 modes when the appropriate bit is set in the DOR.
$\overline{DACK}$	3	I	**DMA ACKNOWLEDGE:** Control input that qualifies the $\overline{RD}$, $\overline{WR}$ inputs in DMA cycles. Normally active low, but is disabled in AT and Model 30 modes when the appropriate bit is set in the DOR.
TC	25	I	**TERMINAL COUNT:** Control line from a DMA controller that terminates the current disk transfer. TC is accepted only while $\overline{DACK}$ is active. This input is active high in the AT, and Model 30 modes and active low in the PS/2 mode.
INT	23	O	**INTERRUPT:** Signals a data transfer in non-DMA mode and when status is valid. Normally active high, but goes to high impedance in AT, and Model 30 modes when the appropriate bit is set in the DOR.
X1 X2	33 34		**CRYSTAL 1,2:** Connection for a 24 MHz fundamental mode parallel resonant crystal. X1 may be driven with a MOS level clock and X2 would be left unconnected.

Table 1. 82077SL Pin Description (Continued)

<table>
<tr><th>Symbol</th><th>Pin#</th><th>I/O</th><th>Description</th></tr>
<tr><td colspan="4">HOST INTERFACE (Continued)</td></tr>
<tr><td>IDENT</td><td>27</td><td>I</td><td>IDENTITY: Upon Hardware RESET, this input (along with MFM pin) selects between the three interface modes. After RESET, this input selects the type of drive being accessed and alters the level on DENSEL. The MFM pin is also sampled at Hardware RESET, and then becomes an output again. Internal pull-ups on MFM permit a no connect.
<table>
<tr><th>IDENT</th><th>MFM</th><th>INTERFACE</th></tr>
<tr><td>1</td><td>1 or NC</td><td>AT Mode</td></tr>
<tr><td>1</td><td>0</td><td>ILLEGAL</td></tr>
<tr><td>0</td><td>1 or NC</td><td>PS/2 Mode</td></tr>
<tr><td>0</td><td>0</td><td>Model 30 Mode</td></tr>
</table>
AT MODE: Major options are: enables DMA Gate logic, TC is active high, Status Registers A & B not available.
PS/2 MODE: Major options are: No DMA Gate logic, TC is active low, Status Registers A & B are available.
MODEL 30 MODE: Major options are: enable DMA Gate logic, TC is active high, Status Registers A & B available.
After Hardware reset this pin determines the polarity of the DENSEL pin. IDENT at a logic level of "1", DENSEL will be active high for high (500 Kbps/1 Mbps) data rates (typically used for 5.25" drives). IDENT at a logic level of "0", DENSEL will be active low for high data rates (typically used for 3.5" drives). This assumes the $\overline{\text{INVERT}}$ pin to be tied to ground.</td></tr>
<tr><td colspan="4">DISK CONTROL (All outputs have 40 mA drive capability)</td></tr>
<tr><td>$\overline{\text{INVERT}}$</td><td>35</td><td>I</td><td>INVERT: Strapping option. Determines the polartity of all signals in this section. Should be strapped to ground when using the internal buffers and these signals become active LOW. When strapped to VCC, these signals become active high and external inverting drivers and receivers are required.</td></tr>
<tr><td>ME0
ME1
ME2
ME3</td><td>57
61
63
66</td><td>O</td><td>ME0–3: Decoded Motor enables for drives 0–3. The motor enable pins are directly controlled via the Digital Output Register.</td></tr>
<tr><td>DS0
DS1
DS2
DS3</td><td>58
62
64
67</td><td>O</td><td>DRIVE SELECT 0–3: Decoded drive selects for drives 0–3. These outputs are decoded from the select bits in the Digital Output Register and gated by ME0–3.</td></tr>
<tr><td>HDSEL</td><td>51</td><td>O</td><td>HEAD SELECT: Selects which side of a disk is to be used. An active level selects side 1.</td></tr>
<tr><td>STEP</td><td>55</td><td>O</td><td>STEP: Supplies step pulses to the drive.</td></tr>
<tr><td>DIR</td><td>56</td><td>O</td><td>DIRECTION: Controls the direction the head moves when a step signal is present. The head moves toward the center if active.</td></tr>
<tr><td>WRDATA</td><td>53</td><td>O</td><td>WRITE DATA: FM or MFM serial data to the drive. Precompensation value is selectable through software.</td></tr>
<tr><td>WE</td><td>52</td><td>O</td><td>WRITE ENABLE: Drive control signal that enables the head to write onto the disk.</td></tr>
<tr><td>DENSEL</td><td>49</td><td>O</td><td>DENSITY SELECT: Indicates whether a low (250/300 Kbps) or high (500 Kbps/1 Mbps) data rate has been selected.</td></tr>
<tr><td>DSKCHG</td><td>31</td><td>I</td><td>DISK CHANGE: This input is reflected in the Digital Input Register.</td></tr>
</table>

Table 1. 82077SL Pin Description (Continued)

Symbol	Pin#	I/O	Description
DISK CONTROL (All outputs have 40 mA drive capability) (Continued)			
DRV2	30	I	**DRIVE2:** This indicates whether a second drive is installed and is reflected in Status Register A.
TRK0	2	I	**TRACK0:** Control line that indicates that the head is on track 0.
WP	1	I	**WRITE PROTECT:** Indicates whether the disk drive is write protected.
INDX	26	I	**INDEX:** Indicates the beginning of the track.
PLL SECTION			
RDDATA	41	I	**READ DATA:** Serial data from the disk. INVERT also affects the polarity of this signal.
MFM	48	I/O	**MFM:** At Hardware RESET, aids in configuring the 82077SL. Internal pull-up allows a no connect if a "1" is required. After reset this pin becomes an output and indicates the current data encoding/decoding mode (Note: If the pin is held at logic level "0" during hardware RESET it must be pulled to "1" after reset to enable the output. The pin can be released on the falling edge of hardware RESET to enable the output). MFM is active high (MFM).
DRATE0 DRATE1	28 29	O	**DATARATE0–1:** Reflects the contents of bits 0,1 of the Data Rate Register. (Drive capability of +6.0 mA @ 0.4V and −4.0 mA @ 2.4V)
PLL0	39	I	**PLL0:** This input optimizes the data separator for either floppy disks or tape drives. A "1" (or V_{CC}) selects the floppy mode, a "0" (or GND) selects tape mode.
POWERDOWN STATUS			
IDLE	38	O	**IDLE:** This pin indicates that the part is in the IDLE state and can be powered down. IDLE state is defined as MSR = 80H, INT = 0, and the head being "unloaded" (as defined in Section 6.2.6). Whenever the part is in this state, IDLE pin is active high. If the part is powered down by the Auto Mode, IDLE pin is set high and if the part is powered down by setting the DSR POWERDOWN bit, IDLE pin is set low.
PD	37	O	**POWERDOWN:** This pin is active high whenever the part is in powerdown state, either via DSR POWERDOWN bit or via the Auto Mode. This pin can be used to disable external oscillator's output.
MISCELLANEOUS			
VCC	18 40 60 68		**Voltage:** +5V
GND	9 12 16 21 36 50 54 59 65		**Ground**
AVCC	46		**Analog Supply**
AVSS	45		**Analog Ground**
NC	42 43 44 47		**No Connection:** These pins **MUST** be left unconnected.

1.0 INTRODUCTION

The 82077SL is a single-chip floppy disk controller for portable PC designs, PC-AT, Microchannel and EISA systems. The 82077SL includes all the power management features necessary to implement a powerful laptop and notebook solution. The 82077SL is fully compatible with the 82077AA. The pin out remains the same with the exception of two new powerdown status pins, PD and IDLE. These pins will replace the LOFIL and HIFIL pins on the 82077AA that are used to connect an external capacitor.

The 82077SL, a 24 MHz crystal, a resistor package and a chip select implement a complete design. The power management features of the 82077SL are designed to be transparent to all application software. The 82077SL will seem awake to the software even when it is in powerdown mode. All drive control signals are fully decoded and have 40 mA drive buffers with selectable polarity. Signals returned from the drive are sent through on-chip input buffers with hysteresis for noise immunity. The integrated analog data separator needs no external compensation of components, yet allows for wide motor variation with exceptionally low soft error rates. The microprocessor interface has 12 mA drive buffers on the data bus plus 100% hardware register compatibility for PC-AT and Microchannel systems. The 16-byte FIFO with programmable thresholds is extremely useful in multi-master systems (Microchannel, EISA) or systems with large bus latency.

Upon hardware reset, (Pin 32) the 82077SL defaults to 8272A functionality. Figure 1-1 is a block diagram of the 82077SL.

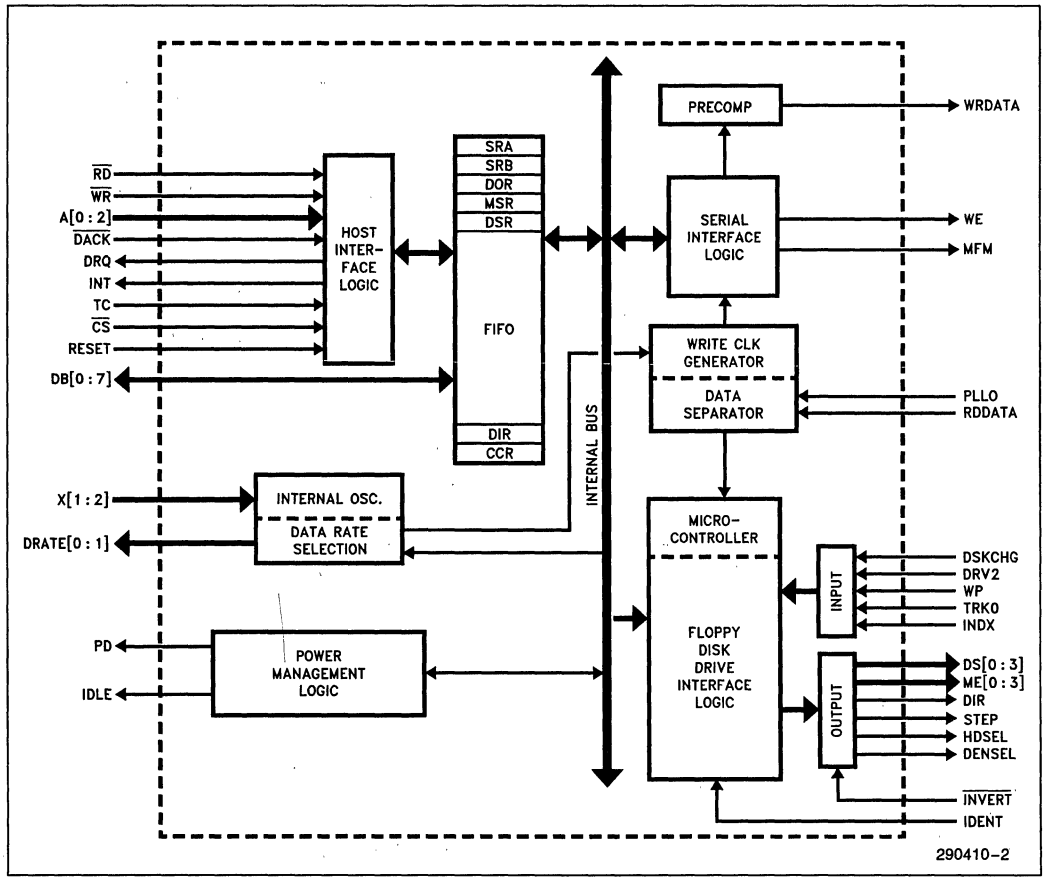


Figure 1-1. 82077SL Block Diagram

1.1 Perpendicular Recording Mode

An added capability of the 82077SL is the ability to interface directly to perpendicular recording floppy drives. Perpendicular recording differs from the traditional longitudinal method by orienting the magnetic bits vertically. This scheme packs in more data bits for the same area.

The 82077SL with perpendicular recording drives can read standard 3.5″ floppies as well as read and write perpendicular media. Some manufacturers offer drives that can read and write standard and perpendicular media in a perpendicular media drive.

A single command puts the 82077SL into perpendicular mode. All other commands operate as they normally do. The perpendicular mode requires the 1 Mbps data rate of the 82077SL. At this data rate, the FIFO eases the host interface bottleneck due to the speed of data transfer to or from the disk.

1.2 Power Management Scheme

While maintaining compatibility with 82077AA, the 82077SL contains a powerful set of features for conserving power. This enables the 82077SL to play an important role in the power sensitive environment of portable personal computers. These features are transparent to any application software.

The 82077SL supports two powerdown modes—direct powerdown and automatic powerdown. Direct powerdown refers to direct action by the software to powerdown without dependence on external factors. Automatic powerdown results from 82077SL's monitoring of the current conditions according to a previously programmed mode. The 82077SL contains a new powerdown command that via programming can be used to invoke auto powerdown. 82077SL is powered down whenever a set of conditions are satisfied. Any hardware reset disables the automatic powerdown command. Software resets have no effect on the POWERDOWN command parameters.

The 82077SL also supports powerdown of its internal crystal oscillator independent of the powerdown modes described above. By setting bit 5 in DSR register, the internal oscillator is turned off. This bit has sole control of the oscillator powerdown. This allows the internal oscillator to be turned off when an external oscillator is used.

2.0 MICROPROCESSOR INTERFACE

The interface consists of the standard asynchronous signals: $\overline{RD}$, $\overline{WR}$, $\overline{CS}$, A0–A2, INT, DMA control and a data bus. The address lines select between configuration registers, the FIFO and control/status registers. This interface can be switched between PC AT, Model 30, or PS/2 normal modes. The PS/2 register sets are a superset of the registers found in a PC-AT.

2.1 Status, Data and Control Registers

As shown below, the base address range is supplied via the $\overline{CS}$ pin. For PC-AT or PS/2 designs, the primary and secondary address ranges are 3F0 Hex to 37F Hex and 370 Hex to 377 Hex respectively.

A2	A1	A0	Access Type	Register	
0	0	0	R	Status Register A	SRA
0	0	1	R	Status Register B	SRB
0	1	0	R/W	Digital Output Register	DOR
0	1	1	R/W	Tape Drive Register	TDR
1	0	0	R	Main Status Register	MSR
1	0	0	W	Data Rate Select Register	DSR
1	0	1	R/W	Data (First In First Out)	FIFO
1	1	0		Reserved	
1	1	1	R	Digital Input Register	DIR
1	1	1	W	Configuration Control Register	CCR

In the following sections, the various registers are shown in their powerdown state. The "UC" notation stands for a value that is returned without change from the active mode. The notation " * " means that the value is reflecting the actual status of the 82077SL, but the value is determinable in the powerdown state. "N/A" reflects the values of the pins indicated. "X" indicates that the value is undefined.

2.1.1a STATUS REGISTER A (SRA, PS/2 MODE)

This register is read-only and monitors the state of the interrupt pin and several disk interface pins. This register is part of the register set, and is not accessible in PC-AT mode.

This register can be accessed during powerdown state without waking up the 82077SL from its powerdown state.

Bits	7	6°	5	4°	3	2°	1°C	0
Function	INT PENDING	$\overline{DRV2}$	STEP	$\overline{TRK0}$	HDSEL	$\overline{INDX}$	$\overline{WP}$	DIR
H/W Reset State	0	N/A	0	N/A	0	N/A	N/A	0
Auto PD State	0*	UC	0*	1	0*	1	1	0*

The INT PENDING bit is used by software to monitor the state of the 82077SL INTERRUPT pin. The bits marked with a " ° " reflect the state of drive signals on the cable and are independent of the state of the $\overline{INVERT}$ pin.

The INT PENDING bit is low by definition for 82077SL to be in powerdown. The bits reflecting the floppy disk drive input pins (TRK0, INDEX and WP) are forced to an inactive state. The floppy disk drive outputs (HDSEL, STEP, and DIR) also go to their inactive, default state.

As a read-only register, there is no default value associated with a reset other than some drive bits will change with a reset. The INT PENDING, STEP, HDSEL, and DIR bits will be low after reset.

2.1.1b STATUS REGISTER A (SRA, MODEL 30 MODE)

Bits	7	6	5	4	3	2	1	0
Function	INT PENDING	DRQ	STEP F/F	TRK0	$\overline{HDSEL}$	INDX	WP	$\overline{DIR}$
H/W Reset State	0	0	0	N/A	1	N/A	N/A	1
Auto PD State	0*	0*	0	0	1*	0	0	1*

This register has the following changes in PS/2 Model 30 Mode. Disk interface pins (Bits 0, 1, 2, 3, & 4) are inverted from PS/2 Mode. The DRQ bit monitors the status of the DMA Request pin. The STEP bit is latched with the Step output going active and is cleared with a read to the DIR register, Hardware or Software RESET.

The DRQ bit is low by definition for 82077SL to be in powerdown. The bits reflecting the floppy disk drive input pins (TRK0, INDEX and WP) are forced to reflect an inactive state. The floppy disk drive outputs (HDSEL, STEP, and DIR) also go to their inactive, default state.

2.1.2a STATUS REGISTER B (SRB, PS/2 MODE)

This register is read-only and monitors the state of several disk interface pins. This register is part of the PS/2 register set, and is not accessible in PC-AT mode.

Bits	7	6	5	4	3*	2	1	0
Function	1	1	DRIVE SEL 0	WRDATA TOGGLE	RDDATA TOGGLE	WE	MOT EN1	MOT EN0
H/W Reset State	1	1	0	0	0	0	0	0
Auto PD State	1	1	UC	0	0	0*	0	0

As the only drive input, RDDATA TOGGLE's activity is independent of the $\overline{INVERT}$ pin level and reflects the level as seen on the cable.

The two TOGGLE bits do not read back the state of their respective pins directly. Instead, the pins drive a Flip/Flop which produces a wider and more reliably read pulse. Bits 6 and 7 are undefined and always return a 1.

After any reset, the activity on the TOGGLE pins are cleared. Drive select and Motor bits cleared by the RESET pin and not software resets.

2.1.2b STATUS REGISTER B (SRB, MODEL 30 MODE)

Bits	7	6	5	4	3	2	1	0
Function	$\overline{DRV2}$	$\overline{DS1}$	$\overline{DS0}$	WRDATA F/F	RDDATA F/F	WE F/F	$\overline{DS3}$	$\overline{DS2}$
H/W Reset State	N/A	1	1	0	0	0	1	1
Auto PD State	UC	UC	UC	0	0	0	UC	UC

This register has the following changes in Model 30 Mode. Bits 0, 1, 5, and 6 return the decoded value of the Drive Select bits in the DOR register. Bits 2, 3, and 4 are set by their respective active going edges and are cleared by reading the DIR register. The WRDATA bit is triggered by raw WRDATA signals and is not gated by WE. Bits 2, 3, and 4 are cleared to a low level by either Hardware or Software RESET.

2.1.3 DIGITAL OUTPUT REGISTER (DOR)

The Digital Output Register contains the drive select and motor enable bits, a reset bit and a $\overline{DMA\ GATE}$ bit.

Bits	7	6	5	4	3	2	1	0
Function	MOT EN3	MOT EN2	MOT EN1	MOT EN0	DMA $\overline{GATE}$	$\overline{RESET}$	DRIVE SEL1	DRIVE SEL2
H/W Reset State	0	0	0	0	0	0	0	0
Auto PD State	0*	0*	0*	0*	UC	1*	UC	UC

The MOT ENx bits directly control their respective motor enable pins (ME0–3). A one means the pin is active, the $\overline{INVERT}$ pin determines the active level. The DRIVE SELx bits are decoded to provide four drive select lines and only one may be active at a time. A one is active and the $\overline{INVERT}$ pin determines the level on the cable. Standard programming practice is to set both MOT ENx and DRIVE SELx bits at the same time.

Table 2-1 lists a set of DOR values to activate the drive select and motor enable for each drive.

Table 2-1. Drive Activation Values

Drive	DOR Value
0	1CH
1	2DH
2	4EH
3	8FH

The $\overline{DMAGATE}$ bit is enabled only in PC-AT and Model 30 Modes. If $\overline{DMAGATE}$ is set low, the INT and DRQ outputs are tristated and the $\overline{DACK}$ and TC inputs are disabled. $\overline{DMAGATE}$ set high will enable INT, DRQ, TC, and $\overline{DACK}$ to the system. In PS/2 Mode $\overline{DMAGATE}$ has no effect upon INT, DRQ, TC or $\overline{DACK}$ pins and they are always active.

The DOR reset bit and the Motor Enable bits have to be inactive when the 82077SL is in powerdown. The $\overline{DMAGATE}$ and DRIVE SEL bits are unchanged. During powerdown, writing to the DOR does not awaken the 82077SL with the exception of activating any of the motor enable bits. Setting the motor enable bits active (high) will wake up the part.

This $\overline{RESET}$ bit clears the basic core of the 82077SL and the FIFO circuits when the LOCK bit is set to "0" (see Section 5.3.2 for LOCK bit definition). Once set, it remains set until the user clears this bit. This bit is set by a chip reset and the 82077SL is held in a reset state until the user clears this bit. The $\overline{RESET}$ bit has no effect upon this register.

2.1.4 TAPE DRIVE REGISTER (TDR)

Bits	7	6	5	4	3	2	1	0
Function	—	—	—	—	—	—	TAPE SEL1	TAPE SEL0
H/W Reset State	—	—	—	—	—	—	0	0
Auto PD State	—	—	—	—	—	—	UC	UC

This register allows the user to assign tape support to a particular drive during initialization. Any future references to that drive number automatically invokes tape support. Hardware reset clears this register; software resets have not effect. TDR[2:7] are not writable and remain tristated if read. The tape select bits are hardware RESET to zeros, making Drive 0 not available for tape support. Drive 0 is reserved for the floppy boot drive. The tuning of the PLL for tape characteristics can also be done in hardware. If a 0 (GND) is applied to pin 39 (PLL0) the PLL is optimized for tape drives, a 1 (V_{CC}) optimizes the PLL for floppies. This hardware selection mechanism overrides the software selection scheme. A typical hardware application would route the Drive Select pin used for tape drive support to pin 39 (PLL0). For further explanation on optimizing 82077 for tape drives please refer to Section 10.2.4.

2.1.5 DATARATE SELECT REGISTER (DSR)

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Function	S/W RESET	POWER DOWN	PDOSC	PRE-COMP2	PRE-COMP1	PRE-COMP0	DRATE SEL1	DRATE SEL0
H/W Reset State	0	0	0	0	0	0	1	0
Auto PD State	0	0	UC	UC	UC	UC	UC	UC

This register ensures backward compatibility with the 82072 floppy controller and is write-only. Changing the data rate changes the timings of the drive control signals. To ensure that drive timings are not violated when changing data rates, choose a drive timing such that the fastest data rate will not violate the timing.

This register is identical to the one used in 82077AA with the exception of bit 5. This bit in the 82077SL denoted by PDOSC is used to implement crystal oscillator power management. The internal oscillator in the 82077SL can be programmed to be either powered on or off via the PDOSC bit. This capability is independent of the chip's powerdown state. In other words, auto powerdown mode and powerdown via activating POWERDOWN bit has no effect over the power state of the oscillator.

In the default state the PDOSC bit is low and the oscillator is powered up. When this bit is programmed to a one, the oscillator is shut off. Hardware reset clears this bit to a zero. Neither of the software resets (via DOR or DSR) have any effect on this bit. When an external oscillator is used, this bit can be set to reduce power consumption. When an internal oscillator is used, this bit can be set to turn off the oscillator to conserve power. However, PDOSC must go high only when the part is in the powerdown state, otherwise the part will not function correctly and must be hardware reset once the oscillator has turned back on and stabilized. Setting the PDOSC bit has no effect on the clock input to the 82077SL (the X1 pin). The clock input is separately disabled when the part is powered down.

S/W RESET behaves the same as DOR RESET except that this reset is self clearing.

POWERDOWN bit implements direct powerdown. Setting this bit high will put the 82077SL into the powerdown state regardless of the state of the part. The part is internally reset and then put into powerdown. No status is saved and any operation in progress is aborted. Unlike the 82077AA this mode of powerdown does not turn off the internal oscillator. Any hardware or software reset will exit the 82077SL from this powerdown state. When 82077SL enters powerdown via this state it affects the floppy disk drive interface as suggested in Section 4.2.2. The state of the floppy disk drive pins during powerdown via the DSR register behaves similarly to that during auto powerdown.

PRECOMP 0–2 adjusts the WRDATA output to the disk to compensate for magnetic media phenomena known as bit shifting. The data patterns that are susceptible to bit shifting are well understood and the 82077SL compensates the data pattern as it is written to the disk. The amount of precompensation is dependent upon the drive and media but in most cases the default value is acceptable.

The 82077SL starts precompensating the data pattern starting on Track 0. The CONFIGURE command can change the track that precompensating starts on. Table 2-2 lists the precompensation values that can be selected and Table 2-3 lists the default precompensation values. The default value is selected if the three bits are zeros.

DRATE 0–1 select one of the four data rates as listed in Table 2-4. The default value is 250 Kbps upon a chip ("Hardware") reset. Other ("Software") Resets do not affect the DRATE or PRECOMP bits.

Table 2-2. Precompensation Delays

PRECOMP 432	Precompensation Delay
111	0.00 ns—DISABLED
001	41.67 ns
010	83.34 ns
011	125.00 ns
100	166.67 ns
101	208.33 ns
110	250.00 ns
000	DEFAULT

Table 2-3. Default Precompensation Delays

Data Rate	Precompensation Delays
1 Mbps	41.67 ns
500 Kbps	125 ns
300 Kbps	125 ns
250 Kbps	125 ns

Table 2-4. Data Rates

DRATESEL 1	DRATESEL 0	DATA RATE MFM
1	1	1 Mbps
0	0	500 Kbps
0	1	300 Kbps
1	0	250 Kbps

2.1.6 MAIN STATUS REGISTER (MSR)

Bits	7	6	5	4	3*	2	1	0
Function	RQM	DIO	NON DMA	CMD BSY	DRV3 BUSY	DRV2 BUSY	DRV1 BUSY	DRV0 BUSY
H/W Reset State	0	X	X	X	X	X	X	X
Auto PD State	1	0	0	0	0	0	0	0

The Main Status Register is a read-only register and is used for controlling command input and result output for all commands.

RQM—Indicates that the host can transfer data if set to a 1. No access is permitted if set to a 0.

DIO—Indicates the direction of a data transfer once RQM is set. A 1 indicates a read and a 0 indicates a write is required.

NON-DMA—This mode is selected in the SPECIFY command and will be set to a 1 during the execution phase of a command. This is for polled data transfers and helps differentiate between the data transfer phase and the reading of result bytes.

COMMAND BUSY—This bit is set to a one when a command is in progress. This bit will go active after the command byte has been accepted and goes inactive at the end of the results phase. If there is no result phase (SEEK, RECALIBRATE commands), this bit is returned to a 0 after the last command byte.

DRV x BUSY—These bits are set to ones when a drive is in the seek portion of a command, including seeks, and recalibrates.

2.1.7 FIFO (DATA)

All command parameter information and disk data transfers go through the FIFO. The FIFO is 16 bytes in size and has programmable threshold values. Data transfers are governed by the RQM and DIO bits in the Main Status Register.

The FIFO defaults to an 8272A compatible mode after a "Hardware" reset (Reset via pin 32). "Software" Resets (Reset via DOR or DSR register) can also place the 82077SL into 8272A compatible mode if the LOCK bit is set to "0" (See section 5.3.2 for the definition of the LOCK bit). This maintains PC-AT hardware compatibility. The default values can be changed through the CONFIGURE command (enable full FIFO operation with threshold control). The advantage of the FIFO is that it allows the system a larger DMA latency without causing a disk error. Table 2.5 gives several examples of the delays with a FIFO. The data is based upon the following formula:

$$\text{Threshold\#} \times \left| \frac{1}{\text{DATA RATE}} \times 8 \right| - 1.5\ \mu s = \text{DELAY}$$

Table 2-5. FIFO Service Delay

FIFO Threshold Examples	Maximum Delay to Servicing at 1 Mbps Data Rate
1 byte	$1 \times 8\ \mu s - 1.5\ \mu s = 6.5\ \mu s$
2 bytes	$2 \times 8\ \mu s - 1.5\ \mu s = 14.5\ \mu s$
8 bytes	$8 \times 8\ \mu s - 1.5\ \mu s = 62.5\ \mu s$
15 bytes	$15 \times 8\ \mu s - 1.5\ \mu s = 118.5\ \mu s$

FIFO Threshold Examples	Maximum Delay to Servicing at 500 Kbps Data Rate
1 byte	$1 \times 16\ \mu s - 1.5\ \mu s = 14.5\ \mu s$
2 bytes	$2 \times 16\ \mu s - 1.5\ \mu s = 30.5\ \mu s$
8 bytes	$8 \times 16\ \mu s - 1.5\ \mu s = 126.5\ \mu s$
15 bytes	$15 \times 16\ \mu s - 1.5\ \mu s = 238.5\ \mu s$

At the start of a command, the FIFO action is always disabled and command parameters must be sent based upon the RQM and DIO bit settings. As the 82077SL enters the command execution phase, it clears the FIFO of any data to ensure that invalid data is not transferred.

An overrun or underrun will terminate the current command and the transfer of data. Disk writes will complete the current sector by generating a 00 pattern and valid CRC.

2.1.8a DIGITAL INPUT REGISTER (DIR, PC-AT MODE)

This register is read only in all modes. In PC-AT mode only bit 7 is driven, all other bits remain tristated.

Bits	7	6	5	4	3*	2	1	0
Function	DSKCHG	—	—	—	—	—	—	—
H/W Reset State	N/A	—	—	—	—	—	—	—
Auto PD State	0	—	—	—	—	—	—	—

DSKCHG monitors the pin of the same name and reflects the opposite value seen on the disk cable, regardless of the value of $\overline{\text{INVERT}}$. The DSKCHG bit is forced inactive along with all the inputs from the floppy disk drive. All the other bits remain tristated.

2.1.8b DIGITAL INPUT REGISTER (DIR, PS/2 MODE)

Bits	7	6	5	4	3	2	1	0
Function	DSK CHG	1	1	1	1	DRATE SEL1	DRATE SEL0	HIGH $\overline{\text{DENS}}$
H/W Reset State	N/A	1	1	1	1	1	0	1
Auto PD State	0	1	1	1	1	UC	UC	UC

The following is changed in PS/2 Mode: Bits 6, 5, 4, and 3 return a value of "1", and the DRATE SEL1-0 return the value of the current data rate selected (see Table 2-4 for values).

$\overline{\text{HIGH DENS}}$ is low whenever the 500 Kbps or 1 Mbps data rates are selected. This bit is independent of the effects of the IDENT and $\overline{\text{INVERT}}$ pins.

The DSKCHG bit is forced inactive along with all the inputs from the floppy disk drive. All the other bits (as applicable) remain unchanged.

Table 2-6 shows the state of the DENSEL pin when $\overline{\text{INVERT}}$ is low.

This pin is set high after a pin RESET and is unaffected by DOR and DSR resets.

Table 2-6. DENSEL Encoding

Data Rate	IDENT*	DENSEL
1 Mbps	0	0
	1	1
500 Kbps	0	0
	1	1
300 Kbps	0	1
	1	0
250 Kbps	0	1
	1	0

*After ("Hardware") Chip Reset

2.1.8c DIGITAL INPUT REGISTER (DIR, MODEL 30 MODE)

Bits	7	6	5	4	3	2	1	0
Function	$\overline{DSK}$ $\overline{CHG}$	0	0	0	$\overline{DMA}$ $\overline{GATE}$	NOPREC	DRATE SEL1	DRATE SEL0
H/W Reset State	N/A	0	0	0	0	0	1	0
Auto PD State	1	0	0	0	UC	UC	UC	UC

The following is changed in Model 30 Mode: Bits 6, 5, and 4 return a value of "0", and Bit 7 ($\overline{DSKCHG}$) is inverted in Model 30 Mode.

The DSKCHG bit is forced inactive along with all the inputs from the floppy disk drive. All the other bits (as applicable) remain unchanged.

Bit 3 reflects the value of $\overline{DMAGATE}$ bit set in the DOR register.

Bit 2 reflects the value of NOPREC bit set in the CCR register.

2.1.9a CONFIGURATION CONTROL REGISTER (CCR, PC AT and PS/2 MODES)

This register sets the datarate and is write only. In the PC-AT it is named the DSR.

Bits	7	6	5	4	3	2	1	0
Function	—	—	—	—	—	—	DRATE SEL1	DRATE SEL0
H/W Reset State	—	—	—	—	—	—	1	0
Auto PD State	—	—	—	—	—	—	UC	UC

Refer to the table in the Data Rate Select Register for values. Unused bits should be set to 0.

2.1.9b CONFIGURATION CONTROL REGISTER (CCR, MODEL 30 MODE)

Bits	7	6	5	4	3	2	1	0
Function	—	—	—	—	—	NOPREC	DRATE SEL1	DRATE SEL0
H/W Reset State	—	—	—	—	—	0	1	0
Auto PD State	—	—	—	—	—	UC	UC	UC

NOPREC has no function, and is reset to "0" with a Hardware RESET only.

2.2 RESET

There are three sources of reset on the 82077SL; the RESET pin, a reset generated via a bit in the DOR and a reset generated via a bit in the DSR. All resets take the 82077SL out of the power down state.

On entering the reset state, all operations are terminated and the 82077SL enters an idle state. Activating reset while a disk write activity is in progress will corrupt the data and CRC.

On exiting the reset state, various internal registers are cleared, and the 82077SL waits for a new command. Drive polling will start unless disabled by a new CONFIGURE command.

2.2.1 RESET PIN ("HARDWARE") RESET

The RESET pin is a global reset and clears all registers except those programmed by the SPECIFY command. The DOR Reset bit is enabled and must be cleared by the host to exit the reset state.

2.2.2 DOR RESET vs DSR RESET ("SOFTWARE" RESET)

These two resets are functionally the same. The DSR Reset is included to maintain 82072 compatibility. Both will reset the 8272 core which affects drive status information. The FIFO circuits will also be reset if the LOCK bit is a "0" (See Section 5.3.2 for the definition of the LOCK bit). The DSR Reset clears itself automatically while the DOR Reset requires the host to manually clear it. DOR Reset has precedence over the DSR Reset. The DOR Reset is set automatically upon a pin RESET. The user must manually clear this reset bit in the DOR to exit the reset state.

The t30a specification in the A.C. Specifications gives the minimum amount of time that the DOR reset must be held active. This amount of time that the DOR reset must be held active is dependent upon the data rate. 82077SL requires that the DOR reset bit must be held active for at least 0.5 μs at 250 Kbps. This is less than a typical ISA I/O cycle time.

2.3 DMA Transfers

DMA transfers are enabled with the SPECIFY command and are initiated by the 82077SL by activating the DRQ pin during a data transfer command. The FIFO is enabled directly by asserting $\overline{\text{DACK}}$ and addresses need not be valid. $\overline{\text{CS}}$ can be held inactive during DMA transfers.

3.0 DRIVE INTERFACE

The 82077SL has integrated all of the logic needed to interface to a floppy disk or a tape drive which use floppy interface. All drive outputs have 40 mA drive capability and all inputs use a receive buffer with hysteresis. The internal analog data separator requires no external components, yet allows for an extremely wide capture range with high levels of read-data jitter, and ISV. The designer needs only to run the 82077SL disk drive signals to the disk or tape drive connector.

3.1 Cable Interface

The $\overline{\text{INVERT}}$ pin selects between using the internal buffers on the 82077SL or user supplied inverting buffers. $\overline{\text{INVERT}}$ pulled to V_{CC} disables the internal buffers; pulled to ground will enable them. There is no need to use external buffers with the 82077SL in typical PC applications.

The polarity of the DENSEL pin is controlled through the IDENT pin, after hardware reset. For 5.25" drives a high on DENSEL tells the drive that either the 500 Kbps or 1 Mbps data rate is selected. For some 3.5" drives the polarity of DENSEL changes to a low for high data rates. See **Table 2-6 DENSEL Encoding** for $\overline{\text{IDENT}}$ pin settings.

Additionally, the two types of drives have different electrical interfaces. Generally, the 5.25" drive uses open collector drivers and the 3.5" drives (as used on PS/2) use totem-pole drivers. The output buffers on the 82077SL do not change between open collector or totem-pole, they are always totem-pole. For design information on interfacing 5.25" and 3.5" drives to a single 82077SL, refer to Section 9.

3.2 Data Separator

The function of the data separator is to lock onto the incoming serial read data. When lock is achieved the serial front end logic of the chip is provided with a clock which is synchronized to the read data. The synchronized clock, called Data Window, is used to internally sample the serial data. One state of Data Window is used to sample the data portion of the bit cell, and the alternate state samples the clock portion. Serial to parallel conversion logic separates the read data into clock and data bytes.

To support reliable disk reads the data separator must track fluctuations in the read data frequency. Frequency errors primarily arise from two sources: motor rotation speed variation and instantaneous speed variation (ISV). A second condition, and one that opposes the ability to track frequency shifts is the response to bit jitter.

The internal data separator consists of two analog phase lock loops (PLLs) as shown in Figure 3-1. The two PLLs are referred to as the reference PLL and the data PLL. The reference PLL (the master PLL) is used to bias the data PLL (the slave PLL). The reference PLL adjusts the data PLL's operating point as a function of process, junction temperature and supply voltage. Using this architecture it was possible to eliminate the need for external trim components.

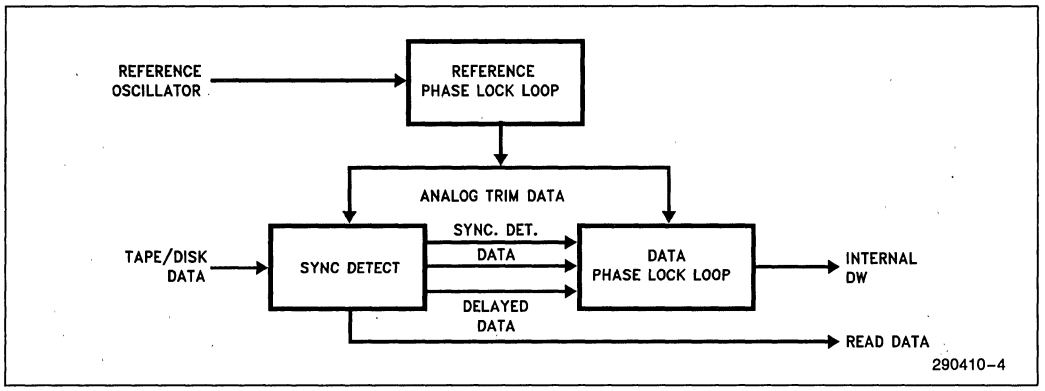


Figure 3-1. Data Separator Block Diagram

PHASE LOCK LOOP OVERVIEW

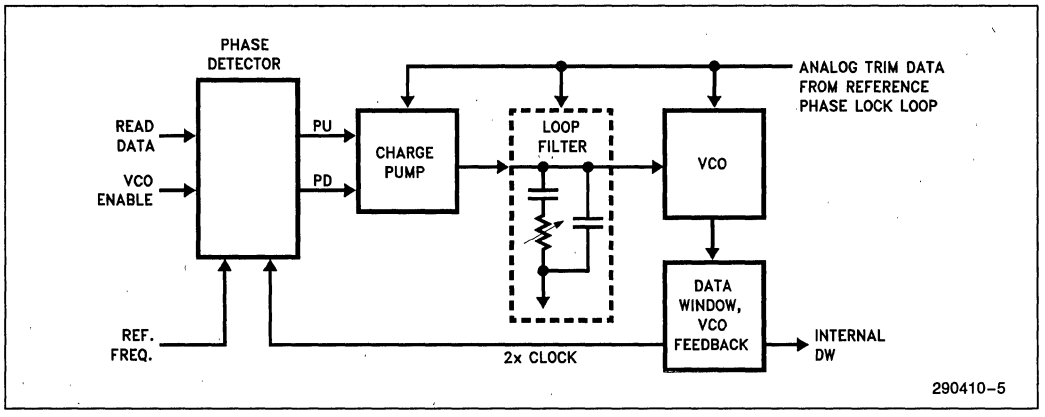


Figure 3-2. Data PLL

Figure 3-2 shows the data PLL. The reference PLL has control over the loop gain by its influence on the charge pump and the VCO. In addition the reference PLL controls the loop filter time constant. As a result the closed loop transfer function of the data PLL is controlled, and immune to the first order, to environmental factors and process variation.

Systems with analog PLLs are often very sensitive to noise. In the design of this data separator many steps were taken to avoid noise sensitivity problems. The analog section of the chip has a separate VSS pin (AVSS) which should be connected externally to a noise free ground. This provides a clean basis for VSS referenced signals. In addition many analog circuit features were employed to make the overall system as insensitive to noise as possible.

3.2.1 JITTER TOLERANCE

The jitter immunity of the system is dominated by the data PLL's response to phase impulses. This is measured as a percentage of the theoretical data window by dividing the maximum readable bit shift by a ¼ bitcell distance. For instance, if the maximum allowable bit shift is 300 ns for a 500 Kbps data stream, the jitter tolerance is 60%. The graph in Figures 13-1 thru 13-4 of the Data Separator Characteristics sections illustrate the jitter tolerance of the 82077SL across each frequency range.

3.2.2 LOCKTIME (tLOCK)

The lock, or settling time of the data PLL is designed to be 64 bit times. This corresponds to 8 sync bytes in the MFM mode. This value assumes that the sync field jitter is 5% the bit cell or less. This level of jitter should be easily achieved for a constant bit pattern, since intersymbol interference should be equal, thus nearly eliminating random bit shifting.

3.2.3 CAPTURE RANGE

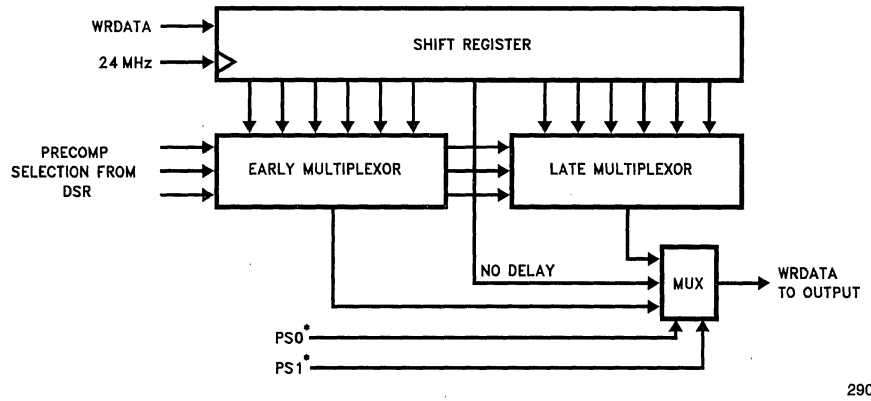
Capture Range is the maximum frequency range over which the data separator will acquire phase lock with the incoming RDDATA signal. In a floppy disk environment, this frequency variation is composed of two components: drive motor speed error and ISV. Frequency is a factor which may determine the maximum level of the ISV (Instantaneous Speed Variation) component. In general, as frequency increases the allowed magnitude of the ISV component will decrease. When determining the capture range requirements, the designer should take the maximum amount of frequency error for the disk drive and double it to account for media switching between drives.

3.3 Write Precompensation

The write precompensation logic is used to minimize bit shifts in the RDDATA stream from the disk drive. The shifting of bits is a known phenomena of magnetic media and is dependent upon the disk media AND the floppy drive.

The 82077SL monitors the bit stream that is being sent to the drive. The data patterns that require precompensation are well known. Depending upon the pattern, the bit is shifted either early or late (or not at all) relative to the surrounding bits. Figure 3-3 is a block diagram of the internal circuit.

The top block is a 13-bit shift register with the no delay tap being in the center. This allows 6 levels of early and late shifting with respect to nominal. The shift register is clocked at the main clock rate (24 MHz). The output is fed into 2 multiplexors—one for early and one for late. A final stage of multiplexors combines the early, late and normal data stream back into one which is the WRDATA output.



NOTE:
PS0,1 are 8272A control signals but are not available as outputs on the 82077SL.

Figure 3-3. Precompensation Block Diagram

4.0 POWER MANAGEMENT FEATURES

The 82077SL contains power management features that makes it ideal for design of portable personal computers. These features can be classified into power management of the part and that of the internal oscillator. The powerdown of the part is done independently of the internal oscillator in the 82077SL.

4.1 Oscillator Power Management

The 82077SL supports a built-in crystal oscillator that can be programmed to be either powered down or active, independent of the power state of the chip. This capability is implemented by the PDOSC bit in the DSR. When PDOSC is set low, the internal oscillator is on and when it is set high the internal oscillator is off. DSR powerdown does not turn off the oscillator.

When the external oscillator is used, power can be saved by turning off the internal oscillator. If the internal oscillator is used, the oscillator may be powered up (even when the rest of the chip is powered off) allowing the chip to wake up quickly and in a stable state. It is recommended to keep the internal oscillator on even when in the powerdown state. The main reason for this is that the recovery time of the oscillator during wake up may take tens of milliseconds under the worst case, which may create problems with any sensitive application software. In a typical application the internal oscillator should be on unless the system goes into a power saving or standby mode (such a mode request would be made by a system time out or by a user). In this case, the system software would take over and must turn on the oscillator sufficiently ahead of awakening the part.

In the case of the external oscillators, the power up characteristics are similar. If the external source remains active during the time the 82077SL is powered down, then the recovery time effect is minimized. The PD pin can be used to turn off the external source. While the PD pin is active 82077SL does not require a clock source. However, when the PD pin is inactive, the clocking source, once it starts oscillating, must be completely stable to ensure that the 82077SL operates properly.

4.2 Part Power Management

This section deals with the power management of the rest of the chip excluding the oscillator. This shows how powerdown modes and wake up modes are activated.

4.2.1 POWERDOWN MODES

The rest of the chip is powered down in two ways—direct powerdown and automatic powerdown. Direct powerdown results in immediate powerdown of the part without regard to the current state of the part. Automatic powerdown results when certain conditions become true within the part.

4.2.1.a Direct Powerdown

Direct powerdown is conducted via the POWERDOWN bit in the DSR register (bit 6). This mode is compatible to the 82077AA. Programming this bit high will powerdown 82077SL after the part is internally reset. All current status is lost if this type of powerdown mode is used. The part can exit powerdown from this mode via any hardware or software reset. This type of powerdown will override the automatic powerdown. If the part is in automatic powerdown when the DSR powerdown is issued then all the previous status of the part will be lost and the 82077SL will be reset to its default values.

4.2.1.b Auto Powerdown

Automatic powerdown is conducted via a "Set Powerdown Mode" command. There are four conditions required before the part will enter powerdown. All these conditions must be true for the part to initiate the powerdown sequence. These conditions are listed as follows:

1. The motor enable pins ME[0:3] must be inactive,
2. The part must be idle; this is indicated by MSR = 80H and INT = 0 (INT may be high even if MSR = 80H due to polling interrupt),
3. The head unload timer (HUT—explained in Section 6.2.6) must have expired, and
4. The auto powerdown timer must have timed out.

The command can be used to enable powerdown by setting the AUTO PD bit in the command to high. The command also provides a capability of programming a minimum power up time via the MIN DLY bit in the command. The minimum power up time refers to a minimum amount of time the part will remain powered up after being awakened or reset. An internal timer is initiated as soon as the auto powerdown command is enabled. The part is then powered down provided all the remaining conditions are met. Any software reset will reinitialize the timer. Changing of data rate extends the auto powerdown timer by up to 10 ms, but only if the data rate is changed during the countdown.

Disabling the auto powerdown mode cancels the timers and holds the 82077SL out of auto powerdown.

4.2.2 WAKE UP MODES

This section describes the conditions for awakening the part from both direct and automatic powerdown. Power conservation or extension of battery life is the main reason power management is required. This means that the 82077SL must be kept in powerdown state as long as possible and should be powered up as late as possible without compromising software transparency.

To keep the part in powerdown mode as late as possible implies that the part should wake up as fast as possible. However, some amount of time is required for the part to exit powerdown state and prepare the internal microcontroller to accept commands. Application software is very sensitive to such a delay and in order to maintain software transparency, the recovery time of the wake up process must be carefully controlled by the system software.

4.2.2.a Wake Up from DSR Powerdown

If the 82077SL enters the powerdown through the DSR powerdown bit, it must be reset to exit. Any form of software or hardware reset will serve, although DSR is recommended. No other register access will awaken the part, including writing to the DOR's motor enable (ME[0:3]) bits.

If DSR powerdown is used when the part is in auto powerdown, the DSR powerdown will override the auto powerdown. However, when the part is awakened by a software reset, the auto powerdown command (including the minimum delay timer) will once again become effective as previously programmed. If the part is awakened via a hardware reset, the auto powerdown is disabled.

After reset, the part will go through a normal sequence. The drive status will be initialized. The FIFO mode will be set to default mode on a hardware reset or on a software reset if the LOCK command has not blocked it. Finally, after a delay, the polling interrupt will be issued.

4.2.2.b Wake Up from Auto Powerdown

If the part enters the powerdown state through the auto powerdown mode, then the part can be awakened by reset or by appropriate access to certain registers.

If a hardware or software reset is used then the part will go through the normal reset sequence. If the access is through the selected registers, then the 82077SL resumes operation as though it was never in powerdown. Besides activating the RESET pin or one of the software reset bits in the DOR or DSR, the following register accesses will wake up the part:

1. Enabling any one of the motor enable bits in the DOR register (reading the DOR does not awaken the part)
2. A read from the MSR register
3. A read or write to the FIFO register

Any of these actions will wake up the part. Once awake, 82077SL will reinitiate the auto powerdown timer for 10 ms or 0.5 sec. (depending on the MIN DLY bit the auto powerdown command). The part will powerdown again when all the powerdown conditions stated in Section 4.2.1b are satisfied.

4.3 Register Behavior

The register descriptions and their values in the powerdown state were given in Section 2.1. Table 4.1 reiterates the AT and PS/2 (including model 30) configuration registers available. It also shows the type of access permitted. In order to maintain software transparency, access to all the registers must be maintained. As Table 4.1 shows, two sets of registers are distinguished based on whether their access results in the part remaining in powerdown state or exiting it.

Access to all other registers is possible without awakening the part. These registers can be accessed during powerdown without changing the status of the part. A read from these registers will reflect the true status as shown in the register description in Section 2.1. A write to the part will result in the part retaining the data and subsequently reflecting it when the part awakens. Accessing the part during powerdown may cause an increase in the power consumption by the part. The part will revert back to its low power mode when the access has been completed.

4.4 Pin Behavior

The 82077SL is specifically designed for the portable PC systems in which the power conservation is a primary concern. This makes the behavior of the pins during powerdown very important.

The pins of 82077SL can be divided into two major categories—system interface and floppy disk drive interface. The floppy disk drive pins are disabled such that no power will be drawn through the 82077SL as a result of any voltage applied to the pin within the 82077SL's power supply range. The floppy disk drive interface pins are configurable by the FDI TRI bit in the auto powerdown command. When the bit is set the output pins of the floppy disk drive retain their original state. All other pins are either disabled or unchanged as depicted in Table 4-4. Most of the system interface pins are left active to monitor system accesses that may wake up the part.

4.4.1 SYSTEM INTERFACE PINS

Table 4.2 gives the state of the system interface pins in the powerdown state. Pins unaffected by powerdown are labeled "UC". Input pins are "DISABLED" to prevent them from causing currents internal to the 82077SL when they have indeterminate input values.

Table 4-1. 82077SL Register Behavior

Address	Available Registers		Access Permitted
	PC-AT	PS/2 (Model 30)	
Access to these registers DOES NOT wake up the part			
000	—	SRA	R
001	—	SRB	R
010	DOR*	DOR*	R/W
011	TDR	TDR	R/W
100	DSR*	DSR*	W
110	—	—	—
111	DIR	DIR	R
111	CCR	CCR	W
Access to these registers wakes up the part			
100	MSR	MSR	R
101	FIFO	FIFO	R/W

NOTE:
*Writing to the DOR or DSR does not wake up the part, however, writing any of the motor enable bits or doing a software reset (either via DOR or DSR reset bits) will wake up the part.

Table 4-2. 82077SL System Interface Pins

System Pins	State in Power Down	System Pins	State in Power Down
Input Pins		**Output Pins**	
$\overline{CS}$	UC	DRQ	UC (Low)
$\overline{RD}$	UC	INT	UC (Low)
$\overline{WR}$	UC	PD	HIGH
A[0:2]	UC	IDLE	High (Auto PD) Low (DSR PD)
DB[0:7]	UC	DB[0:7]	UC
RESET	UC		
IDENT	UC		
$\overline{DACK}$	Disabled		
TC	Disabled		
X[1:2]	Programmable		

Two pins which can be used to indicate the status of the part are IDLE and PD. These pins have replaced the HIFIL and LOFIL pins in the 82077AA. The capacitor required on the 82077AA has been integrated on the chip. Table 4-3 shows how these pins reflect the 82077SL status.

Table 4-3. 82077SL Status Pins

PD	IDLE	MSR	Part Status
1	1	80H	Auto Powerdown
1	0	RQM = 1; MSR[6:0] = X	DSR Powerdown
0	1	80H	Idle
0	0	—	Busy

The IDLE pin indicates when the part is idle state and can be powered down. It is a combination of MSR equalling 80H, the head being unloaded and the INT pin being low. As shown in the table the IDLE pin will be low when the part is in DSR powerdown state. The PD pin is active whenever the part is in the powerdown state. It is active for either mode of powerdown. The PD pin can be used to turn off an external oscillator of other floppy disk drive interface hardware.

4.4.2 FDD INTERFACE PINS

The FDD interface "input" pins during powerdown are disabled or unchanged as shown in Table 4-4. The floppy disk drive "output" pins are programmable by the FDI TRI bit in the auto powerdown command. Setting of the FDI TRI bit in the auto power down command results in the interface retaining its normal state. When this bit is low (default state) all output pins in the FDD interface to the floppy disk drive itself are TRISTATED. Pins used for local logic control or part programming are unaffected. Table 4-4 depicts the state of the floppy disk interface pins in the powerdown state (FDI TRI is low).

Table 4-4. 82077SL FDD Interface Pins

FDD Pins	State in Powerdown	System Pins	State in Powerdown
Input Pins		**Output Pins (FDI TRI = 0)**	
RDDATA	Disabled	ME[0:3]	Tristated
WP	Disabled	DS[0:3]	Tristated
TRK0	Disabled	DIR	Tristated
INDX	Disabled	STEP	Tristated
DRV2	Disabled	WRDATA	Tristated
DSKCHG	Disabled	WE	Tristated
$\overline{INVERT}$	UC	HDSEL	Tristated
MFM	UC	DENSEL	Tristated
		DRATE[0:1]	Tristated

5.0 CONTROLLER PHASES

For simplicity, command handling in the 82077SL can be divided into three phases: Command, Execution and Result. Each phase is described in the following secions.

When there is no command in progress, the 82077SL can be in idle, drive polling or powerdown state.

5.1 Command Phase

After a reset, the 82077SL enters the command phase and is ready to accept a command from the host. For each of the commands, a defined set of command code bytes and parameter bytes has to be written to the 82077SL before the command phase is complete (Please refer to Section 6.0 for the command descriptions). These bytes of data must be transferred in the order prescribed.

Before writing to the 82077SL, the host must examine the RQM and DIO bits of the Main Status Register. RQM, DIO must be equal to "1" and "0" respectively before command bytes may be written. RQM is set false by the 82077SL after each write cycle until the received byte is processed. The 82077SL asserts RQM again to request each parameter byte of the command, unless an illegal command condition is detected. After the last parameter byte is received, RQM remains "0", and the 82077SL automatically enters the next phase as defined by the command definition.

The FIFO is disabled during the command phase to retain compatibility with the 8272A, and to provide for the proper handling of the "Invalid Command" condition.

5.2 Execution Phase

All data transfers to or from the 82077SL occur during the execution phase, which can proceed in DMA or non-DMA mode as indicated in the SPECIFY command.

Each data byte is transferred by an INT or DRQ depending on the DMA mode. The CONFIGURE command can enable the FIFO and set the FIFO threshold value.

The following paragraphs detail the operation of the FIFO flow control. In these descriptions, <threshold> is defined as the number of bytes available to the 82077SL when service is requested from the host, and ranges from 1 to 16. The parameter FIFOTHR which the user programs is one less, and ranges from 0 to 15.

A low threshold value (i.e. 2) results in longer periods of time between service requests, but requires faster servicing of the request, for both read and write cases. The host reads (writes) from (to) the FIFO until empty (full), then the transfer request goes inactive. The host must be very responsive to the service request. This is the desired case for use with a "fast" system.

A high value of threshold (i.e. 12) is used with a "sluggish" system by affording a long latency period after a service request, but results in more frequent service requests.

5.2.1 NON-DMA MODE, TRANSFERS FROM THE FIFO TO THE HOST

The INT pin and RQM bits in the Main Status Register are activated when the FIFO contains (16–<threshold>) bytes, or the last bytes of a full sector transfer have been placed in the FIFO. The INT pin can be used for interrupt driven systems and RQM can be used for polled sytems. The host must respond to the request by reading data from the FIFO. This process is repeated until the last byte is transferred out of the FIFO. The 82077SL will deactivate the INT pin and RQM bit when the FIFO becomes empty.

5.2.2 NON-DMA MODE, TRANSFERS FROM THE HOST TO THE FIFO

The INT pin and RQM bit in the Main Status Register are activated upon entering the execution phase of data transfer commands. The host must respond to the request by writing data into the FIFO. The INT pin and RQM bit remain true until the FIFO becomes full. They are set true again when the FIFO has <threshold> bytes remaining in the FIFO. The INT pin will also be deactivated if TC and DACK# both go inactive. The 82077SL enters the result phase after the last byte is taken by the 82077SL from the FIFO (i.e. FIFO empty condition).

5.2.3 DMA MODE, TRANSFERS FROM THE FIFO TO THE HOST

The 82077SL activates the DRQ pin when the FIFO contains (16–<threshold>) bytes, or the last byte of a full sector transfer has been placed in the FIFO. The DMA controller must respond to the request by reading data from the FIFO. The 82077SL will deactivate the DRQ pin when the FIFO becomes empty. DRQ goes inactive after DACK# goes active for the last byte of a data transfer (or on the active edge of RD#, on the last byte, if no edge is present on DACK#). A data underrun may occur if DRQ is not removed in time to prevent an unwanted cycle.

5.2.4 DMA MODE, TRANSFERS FROM THE HOST TO THE FIFO

The 82077SL activates the DRQ pin when entering the execution phase of the data transfer commands. The DMA controller must respond by activating the DACK# and WR# pins and placing data in the FIFO. DRQ remains active until the FIFO becomes full. DRQ is again set true when the FIFO has <threshold> bytes remaining in the FIFO. The 82077SL will also deactivate the DRQ pin when TC becomes true (qualified by DACK#), indicating that no more data is required. DRQ goes inactive after DACK# goes active for the last byte of a data transfer (or on the active edge of WR# of the last byte, if no edge is present on DACK#). A data overrun may occur if DRQ is not removed in time to prevent an unwanted cycle.

5.2.5 DATA TRANSFER TERMINATION

The 82077SL supports terminal count explicitly through the TC pin and implicitly through the underrun/overrun and end-of-track (EOT) functions. For full sector transfers, the EOT parameter can define the last sector to be transferred in a single or multi-sector transfer. If the last sector to be transferred is a partial sector, the host can stop transferring the data in mid-sector, and the 82077SL will continue to complete the sector as if a hardware TC was received. The only difference between these implicit functions and TC is that they return "abnormal termination" result status. Such status indications can be ignored if they were expected.

Note that when the host is sending data to the FIFO of the 82077SL, the internal sector count will be complete when 82077SL reads the last byte from its side of the FIFO. There may be a delay in the removal of the transfer request signal of up to the time taken for the 82077SL to read the last 16 bytes from the FIFO. The host must tolerate this delay.

5.3 Result Phase

The generation of INT determines the beginning of the result phase. For each of the commands, a defined set of result bytes has to be read from the 82077SL before the result phase is complete. (Refer to Section 6.0 on command descriptions.) These bytes of data must be read out for another command to start.

RQM and DIO must both equal "1" before the result bytes may be read from the FIFO. After all the result bytes have been read, the RQM and DIO bits switch to "1" and "0" respectively, and the CB bit is cleared. This indicates that the 82077SL is ready to accept the next command.

6.0 COMMAND SET/DESCRIPTIONS

Commands can be written whenever the 82077SL is in the command phase. Each command has a unique set of needed parameters and status results. The 82077SL checks to see that the first byte is a valid command and, if valid, proceeds with the command. If it was invalid, the next time the RQM bit in the MSR register is a "1" the DIO and CB bits will also be "1", indicating the FIFO must be read. A result byte of 80H will be read out of the FIFO, indicating an invalid command was issued. After reading the result byte from the FIFO the 82077SL will return to the command phase. Table 6-1 is a summary of the Command set.

Table 6-1. 82077SL Command Set

Phase	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
READ DATA										
Command	W	MT	MFM	SK	0	0	1	1	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W					C				Sector ID information prior to Command execution
	W					H				
	W					R				
	W					N				
	W					EOT				
	W					GPL				
	W					DTL				
Execution										Data transfer between the FDD and system
Result	R					ST 0				Status information after Command execution
	R					ST 1				
	R					ST 2				
	R					C				Sector ID information after Command execution
	R					H				
	R					R				
	R					N				
READ DELETED DATA										
Command	W	MT	MFM	SK	0	1	1	0	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W					C				Sector ID information prior to Command execution
	W					H				
	W					R				
	W					N				
	W					EOT				
	W					GPL				
	W					DTL				
Execution										Data transfer between the FDD and system
Result	R					ST 0				Status information after Command execution
	R					ST 1				
	R					ST 2				
	R					C				Sector ID information after Command execution
	R					H				
	R					R				
	R					N				
WRITE DATA										
Command	W	MT	MFM	0	0	0	1	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W					C				Sector ID information prior to Command execution
	W					H				
	W					R				
	W					N				
	W					EOT				
	W					GPL				
	W					DTL				
Execution										Data transfer between the system and FDD
Result	R					ST 0				Status information after Command execution
	R					ST 1				
	R					ST 2				
	R					C				Sector ID information after Command execution
	R					H				
	R					R				
	R					N				

Table 6-1. 82077SL Command Set (Continued)

Phase	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
WRITE DELETED DATA										
Command	W	MT	MFM	0	0	1	0	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W					C				Sector ID information prior to Command execution
	W					H				
	W					R				
	W					N				
	W					EOT				
	W					GPL				
	W					DTL				
Execution										Data transfer between the FDD and system
Result	R					ST 0				Status information after Command execution
	R					ST 1				
	R					ST 2				
	R					C				
	R					H				Sector ID information after Command execution
	R					R				
	R					N				
READ TRACK										
Command	W	0	MFM	0	0	0	0	1	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W					C				Sector ID information prior to Command execution
	W					H				
	W					R				
	W					N				
	W					EOT				
	W					GPL				
	W					DTL				
Execution										Data transfer between the FDD and system. FDC reads all of cylinders contents from index hole to EOT
Result	R					ST 0				Status information after Command execution
	R					ST 1				
	R					ST 2				
	R					C				
	R					H				Sector ID information after Command execution
	R					R				
	R					N				
VERIFY										
Command	W	MT	MFM	SK	1	0	1	1	0	Command Codes
	W	EC	0	0	0	0	HDS	DS1	DS0	
	W					C				Sector ID information prior to Command execution
	W					H				
	W					R				
	W					N				
	W					EOT				
	W					GPL				
	W					DTL/SC				
Execution										No data transfer takes place
Result	R					ST 0				Status information after Command execution
	R					ST 1				
	R					ST 2				
	R					C				
	R					H				Sector ID information after Command execution
	R					R				
	R					N				
VERSION										
Command	W	0	0	0	1	0	0	0	0	Command Code
Result	R	1	0	0	1	0	0	0	0	Enhanced Controller

Table 6-1. 82077SL Command Set (Continued)

Phase	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
FORMAT TRACK										
Command	W	0	MFM	0	0	1	1	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				N					Bytes/Sector
	W				SC					Sectors/Cylinder
	W				GPL					Gap 3
	W				D					Filler Byte
Execution For Each Sector Repeat:	W				C					Input Sector Parameters
	W				H					
	W				R					
	W				N					
										82077SL formats an entire cylinder
Result	R				ST 0					Status information after Command execution
	R				ST 1					
	R				ST 2					
	R				Undefined					
	R				Undefined					
	R				Undefined					
	R				Undefined					
SCAN EQUAL										
Command	W	MT	MFM	SK	1	0	0	0	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				C					Sector ID Information Prior to Command Execution
	W				H					
	W				R					
	W				N					
	W				EOT					
	W				GPL					
	W				STP					
Execution										Data Compared Between the FDO and Main-System
Result	R				ST 0					Status Information After Command Execution
	R				ST 1					
	R				ST 2					
	R				C					Sector ID Information After Command Execution
	R				H					
	R				R					
	R				N					

2

Table 6-1. 82077SL Command Set (Continued)

Phase	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
SCAN LOW OR EQUAL										
Command	W	MT	MFM	SK	1	1	0	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				C					Sector ID Information
	W				H					Prior to Command
	W				R					Execution
	W				N					
	W				EOT					
	W				GPL					
	W				STP					
Execution										Data Compared Between the FDO and Main-System
Result	R				ST 0					Status Information
	R				ST 1					After Command
	R				ST 2					Execution
	R				C					
	R				H					Sector ID Information
	R				R					After Command
	R				N					Execution
SCAN HIGH OR EQUAL										
Command	W	MT	MFM	SK	1	1	1	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				C					Sector ID Information
	W				H					Prior to Command
	W				R					Execution
	W				N					
	W				EOT					
	W				GPL					
	W				STP					
Execution										Data Compared Between the FDO and Main-System
Result	R				ST 0					Status Information
	R				ST 1					After Command
	R				ST 2					Execution
	R				C					
	R				H					Sector ID Information
	R				R					After Command
	R				N					Execution

Table 6-1. 82077SL Command Set (Continued)

Phase	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
RECALIBRATE										
Command	W	0	0	0	0	0	1	1	1	Command Codes
	W	0	0	0	0	0	0	DS1	DS0	
Execution										Head retracted to Track 0 Interrupt
SENSE INTERRUPT STATUS										
Command	W	0	0	0	0	1	0	0	0	Command Codes
Result	R					ST 0				Status information at the end of each seek operation
	R					PCN				
SPECIFY										
Command	W	0	0	0	0	0	0	1	1	Command Codes
	W	SRT					HUT			
	W	HLT							ND	
SENSE DRIVE STATUS										
Command	W	0	0	0	0	0	1	0	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
Result	R					ST 3				Status information about FDD
SEEK										
Command	W	0	0	0	0	1	1	1	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W					NCN				
Execution										Head is positioned over proper Cylinder on Diskette
CONFIGURE										
Command	W	0	0	0	1	0	0	1	1	Configure Information
	W	0	0	0	0	0	0	0	0	
	W	0	EIS	EFIFO	POLL		FIFOTHR			
	W				PRETRK					
RELATIVE SEEK										
Command	W	1	DIR	0	0	1	1	1	1	
	W	0	0	0	0	0	HDS	DS1	DS0	
	W					RCN				

2

Table 6-1. 82077SL Command Set (Continued)

Phase	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
DUMPREG										
Command	W	0	0	0	0	1	1	1	0	*Note
Execution										Registers placed in FIFO
Result	R				PCN-Drive 0					
	R				PCN-Drive 1					
	R				PCN-Drive 2					
	R				PCN-Drive 3					
	R		SRT				HUT			
	R				HLT				ND	
	R				SC/EOT					
	R	LOCK	0	D_3	D_2	D_1	D_0	GAP	WGATE	
	R	0	EIS	EFIFO	POLL			FIFOTHR		
	R				PRETRK					
READ ID										
Command	W	0	MFM	0	0	1	0	1	0	Commands
	W	0	0	0	0	0	HDS	DS1	DS0	
Execution										The first correct ID information on the Cylinder is stored in Data Register
Result	R				ST 0					Status information after Command execution
	R				ST 1					
	R				ST 2					
	R				C					
	R				H					Disk status after the Command has completed.
	R				R					
	R				N					
PERPENDICULAR MODE										
Command	W	0	0	0	1	0	0	1	0	Command Codes
	W	OW	0	D_3	D_2	D_1	D0	GAP	WGATE	
LOCK										
Command	W	LOCK	0	0	1	0	1	0	0	Command Code
Result	R	0	0	0	LOCK	0	0	0	0	
POWERDOWN MODE										
Command	W	0	0	0	1	0	1	1	1	Command Codes
	W	0	0	0	0	0	FDI TRI	MIN DLY	AUTO PD	
Result	R	0	0	0	0	0	FDI TRI	MIN DLY	AUTO PD	
INVALID										
Command	W				Invalid Codes					Invalid Command Codes (NoOp — 82077SL goes into Standby State)
Result	R				ST 0					ST 0 = 80H

SC is returned if the last command that was issued was the FORMAT command. EOT is returned if the last command was a READ or WRITE.

NOTE:
These bits are used internally only. They are not reflected in the Drive Select pins. It is the users responsibility to maintain correspondence between these bits and the Drive Select pins (DOR).

PARAMETER ABBREVIATIONS

Symbol — **Description**

AUTO PD — Auto powerdown control. If this bit is 0, then the automatic powerdown is disabled. If it is set to 1, then the automatic powerdown is enabled.

C — Cylinder address. The currently selected cylinder address, 0 to 255.

D_0, D_1 D_2, D_3 — Drive Select 0-3. Designates which drives are Perpendicular drives, a "1" indicating Perpendicular drive.

D — Data pattern. The pattern to be written in each sector data field during formatting.

DIR — Direction control. If this bit is 0, then the head will step out from the spindle during a relative seek. If set to a 1, the head will step in toward the spindle.

DS0, DS1 — Disk Drive Select.

DS1	DS0	
0	0	drive 0
0	1	drive 1
1	0	drive 2
1	1	drive 3

DTL — Special sector size. By setting N to zero (00), DTL may be used to control the number of bytes transferred in disk read/write commands. The sector size (N = 0) is set to 128. If the actual sector (on the diskette) is larger than DTL, the remainder of the actual sector is read but is not passed to the host during read commands; during write commands, the remainder of the actual sector is written with all zero bytes. The CRC check code is calculated with the actual sector. When N is not zero, DTL has no meaning and should be set to FF HEX.

EC — Enable Count. When this bit is "1" the "DTL" parameter of the Verify Command becomes SC (Number of sectors per track).

EFIFO — Enable FIFO. When this bit is 0, the FIFO is enabled. A "1" puts the 82077SL in the 8272A compatible mode where the FIFO is disabled.

EIS — Enable implied seek. When set, a seek operation will be performed before executing any read or write command that requires the C parameter in the command phase. A "0" disables the implied seek.

Symbol — **Description**

EOT — End of track. The final sector number of the current track.

FDI TRI — Floppy Drive Interface Tristate: If this bit is 0, then the output pins of the floppy disk drive interface are tristated. This is also the default state. If it is set to 1, then the floppy disk drive interface remains unchanged.

GAP — Alters Gap 2 length when using Perpendicular Mode.

GPL — Gap length. The gap 3 size. (Gap 3 is the space between sectors excluding the VCO synchronization field).

H/HDS — Head address. Selected head: 0 or 1 (disk side 0 or 1) as encoded in the sector ID field.

HLT — Head load time. The time interval that 82077SL waits after loading the head and before initiating a read or write operation. Refer to the SPECIFY command for actual delays.

HUT — Head unload time. The time interval from the end of the execution phase (of a read or write command) until the head is unloaded. Refer to the SPECIFY command for actual delays.

Lock — Lock defines whether EFIFO, FIFOTHR, and PRETRK parameters of the CONFIGURE command can be reset to their default values by a "Software Reset" (Reset made by setting the proper bit in the DSR or DOR registers).

MFM — MFM mode selector. A one selects the double density (MFM) mode.

MIN DLY — Minimum power up time control. This bit is active only if AUTO PD bit is enabled. Setting this bit to a 0, assigns a 10 ms minimum power up time and setting this bit to a 1, assigns a 0.5 sec. minimum power up time.

MT — Multi-track selector. When set, this flag selects the multi-track operating mode. In this mode, the 82077SL treats a complete cylinder, under head 0 and 1, as a single track. The 82077SL operates as if this expanded track started at the first sector under head 0 and ended at the last sector under head 1. With this flag set, a multitrack read or write operation will automatically continue to the first sector under head 1 when the 82077SL finishes operating on the last sector under head 0.

Symbol	Description
N	Sector size code. This specifies the number of bytes in a sector. If this parameter is "00", then the sector size is 128 bytes. The number of bytes transferred is determined by the DTL parameter. Otherwise the sector size is (2 raised to the "N'th" power) times 128. All values up to "07" hex are allowable. "07"h would equal a sector size of 16k. It is the users responsibility to not select combinations that are not possible with the drive.

N	Sector Size
00	128 bytes
01	256 bytes
02	512 bytes
03	1024 bytes
..	...
07	16 Kbytes

NCN	New cylinder number. The desired cylinder number.
ND	Non-DMA mode flag. When set to 1, indicates that the 82077SL is to operate in the non-DMA mode. In this mode, the host is interrupted for each data transfer. When set to 0, the 82077SL operates in DMA mode, interfacing to a DMA controller by means of the DRQ and DACK# signals.
OW	The bits denoted D_0, D_1, D_2, and D_3 of the **PERPENDICULAR MODE** command can only be overwritten when the OW bit is set to "1".
PCN	Present cylinder number. The current position of the head at the completion of SENSE INTERRUPT STATUS command.
POLL	Polling disable. When set, the internal polling routine is disabled. When clear, polling is enabled.
PRETRK	Precompensation start track number. Programmable from track 00 to FFH.
R	Sector address. The sector number to be read or written. In multi-sector transfers, this parameter specifies the sector number of the first sector to be read or written.
RCN	Relative cylinder number. Relative cylinder offset from present cylinder as used by the RELATIVE SEEK command.
SC	Number of sectors. The number of sectors to be initialized by the FORMAT command. The number of sectors to be verified during a Verify Command, when EC is set.

Symbol	Description
SK	Skip flag. When set to 1, sectors containing a deleted data address mark will automatically be skipped during the execution of READ DATA. If READ DELETED is executed, only sectors with a deleted address mark will be accessed. When set to "0", the sector is read or written the same as the read and write commands.
SRT	Step rate interval. The time interval between step pulses issued by the 82077SL. Programmable from 0.5 to 8 milliseconds, in increments of 0.5 ms at the 1 Mbit data rate. Refer to the SPECIFY command for actual delays.
ST0 ST1 ST2 ST3	Status register 0–3. Registers within the 82077SL that store status information after a command has been executed. This status information is available to the host during the result phase after command execution.
WGATE	Write gate alters timing of WE, to allow for pre-erase loads in perpendicular drives.

6.1 Data Transfer Commands

All of the READ DATA, WRITE DATA and VERIFY type commands use the same parameter bytes and return the same results information. The only difference being the coding of bits 0–4 in the first byte.

An implied seek will be executed if the feature was enabled by the CONFIGURE command. This seek is completely transparent to the user. The Drive Busy bit for the drive will go active in the Main Status Register during the seek portion of the command. If the seek portion fails, it will be reflected in the results status normally returned for a READ/WRITE DATA command. Status Register 0 (ST0) would contain the error code and C would contain the cylinder on which the seek failed.

6.1.1 READ DATA

A set of nine (9) bytes is required to place the 82077SL into the Read Data Mode. After the READ DATA command has been issued, the 82077SL loads the head (if it is in the unloaded state), waits the specified head settling time (defined in the SPECIFY command), and begins reading ID Address Marks and ID fields. When the sector address read off the diskette matches with the sector address specified in the command, the 82077SL reads the sector's data field and transfers the data to the FIFO.

After completion of the read operation from the current sector, the sector address is incremented by one, and the data from the next logical sector is read and output via the FIFO. This continuous read function is called "Multi-Sector Read Operation". Upon receipt of TC, or an implied TC (FIFO overrun/underrun), the 82077SL stops sending data, but will continue to read data from the current sector, check the CRC bytes, and at the end of the sector terminate the READ DATA Command.

N determines the number of bytes per sector (see Table 6-2 below). If N is set to zero, the sector size is set to 128. The DTL value determines the number of bytes to be transferred. If DTL is less than 128, the 82077SL transfers the specified number of bytes to the host. For reads, it continues to read the entire 128 byte sector and checks for CRC errors. For writes it completes the 128 byte sector by filling in zeroes. If N is not set to 00 Hex, DTL should be set to FF Hex, and has no impact on the number of bytes transferred.

Table 6-2. Sector Sizes

N	Sector Size
00	128 bytes
01	256 bytes
02	512 bytes
03	1024 bytes
..	...
07	16 Kbytes

The amount of data which can be handled with a single command to the 82077SL depends upon MT (multi-track) and N (Number of bytes/sector).

Table 6-3. Effects of MT and N Bits

MT	N	Max. Transfer Capacity	Final Sector Read from Disk
0	1	256 × 26 = 6,656	26 at side 0 or 1
1	1	256 × 52 = 13,312	26 at side 1
0	2	512 × 15 = 7,680	15 at side 0 or 1
1	2	512 × 30 = 15,360	15 at side 1
0	3	1024 × 8 = 8,192	8 at side 0 or 1
1	3	1024 × 16 = 16,384	16 at side 1

The Multi-Track function (MT) allows the 82077SL to read data from both sides of the diskette. For a particular cylinder, data will be transferred starting at Sector 1, Side 0 and completing at the last sector of the same track at Side 1.

If the host terminates a read or write operation in the 82077SL, then the ID information in the result phase is dependent upon the state of the MT bit and EOT byte. Refer to Table 6-6. The termination must be normal.

At the completion of the READ DATA Command, the head is not unloaded until after the Head Unload Time Interval (specified in the SPECIFY command) has elapsed. If the host issues another command before the head unloads then the head settling time may be saved between subsequent reads.

If the 82077SL detects a pulse on the IDX pin twice without finding the specified sector (meaning that the diskette's index hole passes through index detect logic in the drive twice), the 82077SL sets the IC code in Status Register 0 to "01" (Abnormal termination), and sets the ND bit in Status Register 1 to "1" indicating a sector not found, and terminates the READ DATA Command.

After reading the ID and Data Fields in each sector, the 82077SL checks the CRC bytes. If a CRC error occurs in the ID or data field, the 82077SL sets the IC code in Status Register 0 to "01" (Abnormal termination), sets the DE bit flag in Status Register 1 to "1", sets the DD bit in Status Register 2 to "1" if CRC is incorrect in the ID field, and terminates the READ DATA Command.

Table 6-4 below describes the affect of the SK bit on the READ DATA command execution and results.

Table 6-4. Skip Bit vs READ DATA Command

SK Bit Value	Data Address Mark Type Encountered	Results		
		Sector Read?	CM Bit of ST2 Set?	Description of Results
0	Normal Data	Yes	No	Normal Termination.
0	Deleted Data	Yes	Yes	Address Not Incremented. Next Sector Not Searched For.
1	Normal Data	Yes	No	Normal Termination.
1	Deleted Data	No	Yes	Normal Termination Sector Not Read ("Skipped").

Except where noted in Table 6-4, the C or R value of the sector address is automatically incremented (see Table 6-6).

6.1.2 READ DELETED DATA

This command is the same as the READ DATA command, only it operates on sectors that contain a Deleted Data Address Mark at the beginning of a Data Field.

Table 6-5 describes the affect of the SK bit on the READ DELETED DATA command execution and results.

Table 6-5. Skip Bit vs READ DELETED DATA Command

SK Bit Value	Data Address Mark Type Encountered	Results		
		Sector Read?	CM Bit of ST2 Set?	Description of Results
0	Normal Data	Yes	Yes	Address Not Incremented. Next Sector Not Searched For.
0	Deleted Data	Yes	No	Normal Termination.
1	Normal Data	No	Yes	Normal Termination Sector Not Read ("Skipped").
1	Deleted Data	Yes	No	Normal Termination.

Except where noted in Table 6-5 above, the C or R value of the sector address is automatically incremented (See Table 6-6).

6.1.3 READ TRACK

This command is similar to the READ DATA command except that the entire data field is read continuously from each of the sectors of a track. Immediately after encountering a pulse on the IDX pin, the 82077SL starts to read all data fields on the track as continuous blocks of data without regard to logical sector numbers. If the 82077SL finds an error in the ID or DATA CRC check bytes, it continues to read data from the track and sets the appropriate error bits at the end of the command. The 82077SL compares the ID information read from each sector with the specified value in the command, and sets the ND flag of Status Register 1 to a "1" if there is no comparison. Multi-track or skip operations are not allowed with this command. The MT and SK bits (Bits D7 and D5 of the first command byte respectively) should always be set to "0".

This command terminates when the EOT specified number of sectors have been read. If the 82077SL does not find an ID Address Mark on the diskette after the second occurrence of a pulse on the IDX pin, then it sets the IC code in Status Register 0 to "01" (Abnormal termination), sets the MA bit in Status Register 1 to "1", and terminates the command.

Table 6-6. Result Phase Table

MT	Head	Final Sector Transferred to Host	ID Information at Result Phase			
			C	H	R	N
0	0	Less than EOT	NC	NC	R+1	NC
		Equal to EOT	C+1	NC	01	NC
	1	Less than EOT	NC	NC	R+1	NC
		Equal to EOT	C+1	NC	01	NC
1	0	Less than EOT	NC	NC	R+1	NC
		Equal to EOT	NC	LSB	01	NC
	1	Less than EOT	NC	NC	R+1	NC
		Equal to EOT	C+1	LSB	01	NC

NC: no change, the same value as the one at the beginning of command execution.
LSB: least significant bit, the LSB of H is complemented.

6.1.4 WRITE DATA

After the WRITE DATA command has been issued, the 82077SL loads the head (if it is in the unloaded state), waits the specified head load time if unloaded (defined in the SPECIFY command), and begins reading ID Fields. When the sector address read from the diskette matches the sector address specified in the command, the 82077SL reads the data from the host via the FIFO, and writes it to the sector's data field.

After writing data into the current sector, the 82077SL computes the CRC value and writes it into the CRC field at the end of the sector transfer. The Sector Number stored in "R" is incremented by one, and the 82077SL continues writing to the next data field. The 82077SL continues this "Multi-Sector Write Operation". Upon receipt of a terminal count signal or if a FIFO over/under run occurs while a data field is being written, then the remainder of the data field is filled with zeros.

The 82077SL reads the ID field of each sector and checks the CRC bytes. If it detects a CRC error in one of the ID Fields, it sets the IC code in Status Register 0 to "01" (Abnormal termination), sets the DE bit of Status Register 1 to "1", and terminates the WRITE DATA command.

The WRITE DATA command operates in much the same manner as the READ DATA command. The following items are the same. Please refer to the READ DATA Command for details:

- Transfer Capacity
- EN (End of Cylinder) bit
- ND (No Data) bit
- Head Load, Unload Time Interval
- ID information when the host terminates the command.
- Definition of DTL when N = 0 and when N does not = 0.

6.1.5 WRITE DELETED DATA

This command is almost the same as the WRITE DATA command except that a Deleted Data Address Mark is written at the beginning of the Data Field instead of the normal Data Address Mark. This command is typically used to mark a bad sector containing an error on the floppy disk.

6.1.6 VERIFY

The VERIFY command is used to verify the data stored on a disk. This command acts exactly like a READ DATA command except that no data is transferred to the host. Data is read from the disk, CRC computed and checked against the previously stored value.

Because no data is transferred to the host, TC (pin 25) cannot be used to terminate this command. By setting the EC bit to "1" an implicit TC will be issued to the 82077SL. This implicit TC will occur when the SC value has decrement to 0 (an SC value of 0 will verify 256 sectors). This command can also be terminated by setting the EC bit to "0" and the EOT value equal to the final sector to be checked. If EC is set to "0" DTL/SC should be programmed to 0FFH. Refer to Table 6-6 and Table 6-7 for information concerning the values of MT and EC versus SC and EOT value.

Definitions:

Sectors Per Side = Number of formatted sectors per each side of the disk.

Sectors Remaining = Number of formatted sectors left which can be read, including side 1 of the disk if MT is set to "1".

Table 6-7. Verify Command Result Phase Table

MT	EC	SC/EOT Value	Termination Result
0	0	SC = DTL EOT ≤ # Sectors Per Side	Successful Termination Result Phase Valid
0	0	SC = DTL EOT > # Sectors Per Side	Unsuccessful Termination Result Phase Invalid
0	1	SC ≤ # Sectors Remaining AND EOT ≤ # Sectors Per Side	Successful Termination Result Phase Valid
0	1	SC > # Sectors Remaining OR EOT > # Sectors Per Side	Unsuccessful Termination Result Phase Invalid
1	0	SC = DTL EOT ≤ # Sectors Per Side	Successful Termination Result Phase Valid
1	0	SC = DTL EOT > # Sectors Per Side	Unsuccessful Termination Result Phase Invalid
1	1	SC ≤ # Sectors Remaining AND EOT ≤ # Sectors Per Side	Successful Termination Result Phase Valid
1	1	SC > # Sectors Remaining OR EOT > # Sectors Per Side	Unsuccessful Termination Result Phase Invalid

NOTE:
If MT is set to "1" and the SC value is greater than the number of remaining formatted sectors on Side 0, verifying will continue on Side 1 of the disk.

6.1.7 FORMAT TRACK

The FORMAT command allows an entire track to be formatted. After a pulse from the IDX pin is detected, the 82077SL starts writing data on the disk including Gaps, Address Marks, ID Fields and Data Fields, per the IBM System 34 (MFM). The particular values that will be written to the gap and data field are controlled by the values programmed into N, SC, GPL, and D which are specified by the host during the command phase. The data field of the sector is filled with the data byte specified by D. The ID Field for each sector is supplied by the host; that is, four data bytes per sector are needed by the 82077SL for C, H, R, and N (cylinder, head, sector number and sector size respectively).

After formatting each sector, the host must send new values for C, H, R and N to the 82077SL for the next sector on the track. The R value (sector number) is the only value that must be changed by the host after each sector is formatted. This allows the disk to be formatted with nonsequential sector addresses (interleaving). This incrementing and formatting continues for the whole track until the 82077SL encounters a pulse on the IDX pin again and it terminates the command.

Table 6-8 contains typical values for gap fields which are dependent upon the size of the sector and the number of sectors on each track. Actual values can vary due to drive electronics.

Table 6-8. Typical Values for Formatting

		Sector Size	N	SC	GPL1	GPL2
5.25″ Drives	MFM	256	01	12	0A	0C
		256	01	10	20	32
		512*	02	09	2A	50
		1024	03	04	80	F0
		2048	04	02	C8	FF
		4096	05	01	C8	FF
		...	...			
3.5″ Drives	MFM	256	1	0F	0E	36
		512**	2	09	1B	54
		1024	3	05	35	74

GPL1 = suggested GPL values in read and write commands to avoid splice point between data field and ID field of contiguous sections.
GPL2 = suggested GPL value in FORMAT TRACK command.
*PC-AT values (typical)
**PS/2 values (typical). Applies with 1.0 MB and 2.0 MB drives.

NOTE:
All values except Sector Size are in Hex.

2

6.1.7.1 Format Fields

GAP 4a	SYNC	IAM		GAP 1	SYNC	IDAM		C	H	S	N	C	GAP 2	SYNC	DATA AM		DATA	C	GAP 3	GAP 4b
								Y		E	O	R						R		
80x	12x	3x	FC	50x	12x	3x	FE	L	D	C		C	22x	12x	3x	FB		C		
4E	00	C2		4E	00	A1							4E	00	A1	F8				

Figure 6-1. System 34 Format Double Density

GAP 4a	SYNC	IAM		GAP 1	SYNC	IDAM		C	H	S	N	C	GAP 2	SYNC	DATA AM		DATA	C	GAP 3	GAP 4b
								Y		E	O	R						R		
80x	12x	3x	FC	50x	12x	3x	FE	L	D	C		C	41x	12x	3x	FB		C		
4E	00	C2		4E	00	A1							4E	00	A1	F8				

Figure 6-2. Perpendicular Format

6.1.8 SCAN COMMANDS

The SCAN Commands allow data which is being read from the diskette to be compared against data which is being supplied from the main system (Processor in NON-DMA mode, and DMA Controller in DMA mode). The FDC compares the data on a byte-by-byte basis, and looks for a sector of data which meets the conditions of $D_{FDO} = D_{Processor}$, $D_{FDO} \leq D_{Processor}$, or $D_{FDO} \geq D_{Processor}$. Ones complement arithmetic is used for comparison (FF = largest number, 00 = smallest number). After a whole sector of data is compared, if the conditions are not met, the sector number is incremented (R + STP → R), and the scan operation is continued. The scan operation continues until one of the following conditions occurs; the conditions for scan are met (equal, low, or high), the last sector on the track is reached (EOT), or the terminal count signal is received.

If the conditions for scan are met then the FDC sets the SH (Scan Hit) flag of Status Register 2 to a 1 (high), and terminates the Scan Command. If the conditions for scan are not met between the starting sector (as specified by R) and the last sector on the cylinder (EOT), then the FDC sets the SN (Scan Not Satisfied) flag of Status Register 2 to a 1 (high), and terminates the Scan Command. The receipt of a TERMINAL COUNT signal from the Processor or DMA Controller during the scan operation will cause the FDC to complete the comparison of the particular byte which is in process, and then to terminate the command. Table 6-9 shows the status of bits SH and SN under various conditions of SCAN.

If the FDC encounters a Deleted Data Address Mark on one of the sectors (and SK = 0), then it regards the sector as the last sector on the cylinder, sets CM (Control Mark) flag of Status Register 2 to a 1 (high) and terminates the command. If SK = 1, the FDC skips the sector with the Deleted Address Mark, and reads the next sector. In the second case (SK = 1), the FDC sets the CM (Control Mark) flag of Status Register 2 to a 1 (high) in order to show that a Deleted Sector has been encountered.

When either the STP (contiguous sectors STP = 01, or alternate sectors STP = 02) sectors are read or the MT (Multi-Track) is programmed, it is necessary to remember that the last sector on the track must be read. For example, if STP = 02, MT = 0, the sectors are numbered sequentially 1 through 26, and we start the Scan Command at sector 21; the following will happen. Sectors 21, 23, andf 25 will be read, then the next sector (26) will be skipped and the Index Hole will be encountered before the EOT value of 26 can be read. This will result in an abnormal termination of the command. If the EOT had been set at 25 or the scanning started at sector 20, then the Scan Command would be completed in a normal manner.

During the Scan Command data is supplied by either the processor or DMA Controller for comparison against the data read from the diskette. In order to avoid having the OR (Over Run) flag set in Status Register 1, it is necessary to ahve the data available in less than 13 μs (MFM Mode). If an Overrun occurs the FDC terminates the command.

Table 6-9. Scan Status Codes

Command	Status Register 2		Comments
	Bit 2 = SN	Bit 3 = SH	
Scan Equal	0	1	$D_{FDO} = D_{Processor}$
	1	0	$D_{FDO} \neq D_{Processor}$
Scan Low or Equal	0	1	$D_{FDO} = D_{Processor}$
	0	0	$D_{FDO} < D_{Processor}$
	1	0	$D_{FDO} \not> D_{Processor}$
Scan High or Equal	0	1	$D_{FDO} = D_{Processor}$
	0	0	$D_{FDO} > D_{Processor}$
	1	0	$D_{FDO} \not< D_{Processor}$

6.2 Control Commands

Control commands differ from the other commands in that no data transfer takes place. Three commands generate an interrupt when complete; READ ID, RECALIBRATE and SEEK. The other control commands do not generate an interrupt.

6.2.1 READ ID

The READ ID command is used to find the present position of the recording heads. The 82077SL stores the values from the first ID Field it is able to read into its registers. If the 82077SL does not find an ID Address Mark on the diskette after the second occurrence of a pulse on the IDX pin, it then sets the IC code in Status Register 0 to "01" (Abnormal termination), sets the MA bit in Status Register 1 to "1", and terminates the command.

The following commands will generate an interrupt upon completion. They do not return any result bytes. It is highly recommended that control commands be followed by the SENSE INTERRUPT STATUS command. Otherwise, valuable interrupt status information will be lost.

6.2.2 RECALIBRATE

This command causes the read/write head within the 82077SL to retract to the track 0 position. The 82077SL clears the contents of the PCN counter, and checks the status of the TRK0 pin from the FDD. As long as the TRK0 pin is low, the DIR pin remains 0 and step pulses are issued. When the TRK0 pin goes high, the SE bit in Status Register 0 is set to "1", and the command is terminated. If the TRK0 pin is still low after 79 step pulses have been issued, the 82077SL sets the SE and the EC bits of Status Register 0 to "1", and terminates the command. Disks capable of handling more than 80 tracks per side may require more than one RECALIBRATE command to return the head back to physical Track 0.

The RECALIBRATE command does not have a result phase. SENSE INTERRUPT STATUS command must be issued after the RECALIBRATE command to effectively terminate it and to provide verification of the head position (PCN). During the command phase of the recalibrate operation, the 82077SL is in the BUSY state, but during the execution phase it is in a NON BUSY state. At this time another RECALIBRATE command may be issued, and in this manner, parallel RECALIBRATE operations may be done on up to 4 drives at once.

Upon power up, the software must issue a RECALIBRATE command to properly initialize all drives and the controller.

6.2.3 SEEK

The read/write head within the drive is moved from track to track under the control of the SEEK Command. The 82077SL compares the PCN which is the current head position with the NCN and performs the following operation if there is a difference:

—PCN < NCN: Direction signal to drive set to "1" (step in), and issues step pulses.

—PCN > NCN: Direction signal to drive set to "0" (step out), and issues step pulses.

The rate at which step pulses are issued is controlled by SRT (Stepping Rate Time) in the SPECIFY command. After each step pulse is issued, NCN is compared against PCN, and when NCN = PCN, then the SE bit in Status Register 0 is set to "1", and the command is terminated.

During the command phase of the seek or recalibrate operation, the 82077SL is in the BUSY state, but during the execution phase it is in the NON BUSY state.

Note that if implied seek is not enabled, the read and write commands should be preceded by:

1) SEEK command;	Step to the proper track
2) SENSE INTERRUPT STATUS command;	Terminate the Seek command
3) READ ID.	Verify head is on proper track
4) Issue READ/WRITE command.	

The SEEK command does not have a result phase. Therefore, it is highly recommended that the SENSE INTERRUPT STATUS Command be issued after the SEEK command to terminate it and to provide verification of the head position (PCN). The H bit (Head Address) in ST0 will always return a "0". When exiting DSR POWERDOWN mode, the 82077SL clears the PCN value and the status information to zero. Prior to issuing the DSR POWERDOWN command, it is highly recommended that the user service all pending interrupts through the SENSE INTERRUPT STATUS command.

6.2.4 SENSE INTERRUPT STATUS

An interrupt signal on INT pin is generated by the 82077SL for one of the following reasons:

1. Upon entering the Result Phase of:
   a. READ DATA Command
   b. READ TRACK Command
   c. READ ID Command
   d. READ DELETED DATA Command
   e. WRITE DATA Command
   f. FORMAT TRACK Command
   g. WRITE DELETED DATA Command
   h. VERIFY Command
2. End of SEEK, RELATIVE SEEK or RECALIBRATE Command
3. 82077SL requires a data transfer during the execution phase in the non-DMA Mode

The SENSE INTERRUPT STATUS command resets the interrupt signal and via the IC code and SE bit of Status Register 0, identifies the cause of the interrupt. If a SENSE INTERRUPT STATUS command is issued when no active interrupt condition is present, the status register ST0 will return a value of 80H (invalid command).

Table 6-9. Interrupt Identification

SE	IC	Interrupt Due To
0	11	Polling
1	00	Normal Termination of SEEK or RECALIBRATE command
1	01	Abnormal Termination of SEEK or RECALIBRATE command

The SEEK, RELATIVE SEEK and the RECALIBRATE commands have no result phase. SENSE INTERRUPT STATUS command must be issued immediately after these commands to terminate them and to provide verification of the head position (PCN). The H (Head Address) bit in ST0 will always return a "0". If a SENSE INTERRUPT STATUS is not issued, the drive, will continue to be BUSY and may effect the operation of the next command.

6.2.5 SENSE DRIVE STATUS

SENSE DRIVE STATUS obtains drive status information. It has no execution phase and goes directly to the result phase from the command phase. STATUS REGISTER 3 contains the drive status information.

6.2.6 SPECIFY

The SPECIFY command sets the initial values for each of the three internal timers. The HUT (Head Unload Time) defines the time from the end of the execution phase of one of the read/write commands to the head unload state. The SRT (Step Rate Time) defines the time interval between adjacent step pulses. Note that the spacing between the first and second step pulses may be shorter than the remaining step pulses. The HLT (Head Load Time) defines the time between the Head Load signal goes high and the read, write operation starts. The values change with the data rate speed selection and are documented in Table 6-10.

Table 6-10. Drive Control Delays (ms)

	HUT				SRT			
	1M	500K	300K	250K	1M	500K	300K	250K
0	128	256	426	512	8.0	16	26.7	32
1	8	16	26.7	32	7.5	15	25	30
..	..	..	..	..	..	..	..	..
E	112	224	373	448	1.0	2	3.33	4
F	120	240	400	480	0.5	1	1.67	2

	HLT			
	1M	500K	300K	250K
00	128	256	426	512
01	1	2	3.3	4
02	2	4	6.7	8
..	..	..	..	.
7F	126	252	420	504
7F	127	254	423	508

The choice of DMA or NON-DMA operations is made by the ND bit. When this bit is "1", the NON-DMA mode is selected, and when ND is "0", the DMA mode is selected. In DMA mode, data transfers are signalled by the DRQ pin. Non-DMA mode uses the RQM bit and the INT pin to signal data transfers.

6.2.7 CONFIGURE

Issued to select the special features of the 82077SL. A CONFIGURE command need not be issued if the default values of the 82077SL meet the system requirements.

CONFIGURE DEFAULT VALUES:

EIS — No Implied Seeks
EFIFO — FIFO Disabled
POLL — Polling Enabled
FIFOTHR — FIFO Threshold Set to 1 Byte
PRETRK — Pre-Compensation Set to Track 0

EIS—Enable implied seek. When set to "1", the 82077SL will perform a SEEK operation before executing a read or write command. Defaults to no implied seek.

EFIFO—A "1" puts the FIFO into the 8272A compatible mode where the FIFO is disabled. This means data transfers are asked for on a byte by byte basis. Defaults to "1", FIFO disabled. The threshold defaults to one.

POLL—Disable polling of the drives. Defaults to "0", polling enabled. When enabled, a single interrupt is generated after a RESET. No polling is performed while the drive head is loaded and the head unload delay has not expired.

FIFOTHR—The FIFO threshold in the execution phase of read or write commands. This is programmable from 1 to 16 bytes. Defaults to one byte. A "00" selects one byte "0F" selects 16 bytes.

PRETRK—Pre-compensation start track number. Programmable from track 0 to 255. Defaults to track 0. A "00" selects track 0, "FF" selects 255.

6.2.8 VERSION

The VERSION command checks to see if the controller is an enhanced type or the older type (8272A/765A). A value of 90 H is returned as the result byte, defining an enhanced FDD controller is in use. No interrupts are generated.

6.2.9 RELATIVE SEEK

The command is coded the same as for SEEK, except for the MSB of the first byte and the DIR bit.

DIR Head Step Direction Control.

DIR	Action
0	Step Head Out
1	Step Head In

RCN Relative Cylinder Number that determines how many tracks to step the head in or out from the current track number.

The RELATIVE SEEK command differs from the SEEK command in that it steps the head the absolute number of tracks specified in the command instead of making a comparison against an internal register. The SEEK command is good for drives that support a maximum of 256 tracks. RELATIVE SEEKs cannot be overlapped with other RELATIVE SEEKs. Only one RELATIVE SEEK can be active at a time. Bit 4 of Status Register 0 (EC) will be set if RELATIVE SEEK attempts to step outward beyond Track 0.

As an example, assume that a floppy drive has 300 useable tracks and that the host needs to read track 300 and the head is on any track (0–255). If a SEEK command was issued, the head would stop at track 255. If a RELATIVE SEEK command was issued, the 82077SL would move the head the specified number of tracks, regardless of the internal cylinder position register (but would increment the register). If the head had been on track 40 (D), the maximum track that the 82077SL could position the head on using RELATIVE SEEK, would be 296 (D), the initial track, +256 (D). The maximum count that the head can be moved with a single RELATIVE SEEK command is 256 (D).

The internal register, PCN, would overflow as the cylinder number crossed track 255 and would contain 40 (D). The resulting PCN value is thus (NCN + PCN) mod 256. Functionally, the 82077SL starts counting from 0 again as the track number goes above 255(D). It is the users responsibility to compensate 82077SL functions (precompensation track number) when accessing tracks greater than 255. The 82077SL does not keep track that it is working in an "extended track area" (greater than 255). Any command issued would use the current PCN value except for the RECALIBRATE command which only looks for the TRACK0 signal. RECALIBRATE would return an error if the head was farther than 79 due to its limitation of issuing a maximum 80 step pulses. The user simply needs to issue a second RECALIBRATE command. The SEEK command and implied seeks will function correctly within the 44 (D) track (299–255) area of the "extended track area". It is the users responsibility not to issue a new track position that would exceed the maximum track that is present in the extended area.

To return to the standard floppy range (0–255) of tracks, a RELATIVE SEEK would be issued to cross the track 255 boundary.

A RELATIVE SEEK can be used instead of the normal SEEK but the host is required to calculate the difference between the current head location and the new (target) head location. This may require the host to issue a READ ID command to ensure that the head is physically on the track that software assumes it to be. Different 82077SL commands will return different cylinder results which may be difficult to keep track of with software without the READ ID command.

6.2.10 DUMPREG

The DUMPREG command is designed to support system run-time diagnostics and application software development and debug.

6.2.11 PERPENDICULAR MODE COMMAND

The PERPENDICULAR MODE command should be issued prior to executing READ/WRITE/FORMAT commands that access a disk drive with perpendicular recording capability. With this command, the length of the Gap2 field and VCO enable timing can be altered to accommodate the unique requirements of these drives. Table 6-11 describes the effects of the WGATE and GAP bits for the PERPENDICULAR MODE command. Upon a reset, the 82077SL will default to the conventional mode (WGATE = 0, GAP = 0).

Selection of the 500 Kbps and 1 Mbps perpendicular modes is independent of the actual data rate selected in the Data rate Select Register. The user must ensure that the two data rates remain consistent.

The Gap2 and VCO timing requirements for perpendicular recording type drives are dictated by the design of the read/write head. In the design of this head, a pre-erase head precedes the normal read/write head by a distance of 200 micrometers. This works out to about 38 bytes at a 1 Mbps recording density. Whenever the write head is enabled by the Write Gate signal the pre-erase head is also activated at the same time. Thus, when the write head is initially turned on, flux transitions recorded on the media for the first 38 bytes will not be preconditioned with the pre-erase head since it has not yet been activated. To accommodate this head activation and deactivation time, the Gap2 field is expanded to a length of 41 bytes. The format field shown in Figure 5-3 illustrates the change in the Gap2 field size for the perpendicular format.

On the read back by the 82077SL, the controller must begin synchronization at the beginning of the Sync field. For the conventional mode, the internal PLL VCO is enabled (VCOEN) approximately 24 bytes from the start of the Gap2 field. But when the controller operates in the 1 Mbps perpendicular mode (WGATE = 1, GAP = 1), VCOEN goes active after 43 bytes to accommodate the increased Gap2 field size. For both cases, an approximate 2 byte cushion is maintained from the beginning of the sync field for the purposes of avoiding write splices in the presence of motor speed variation.

For the WRITE DATA case, the 82077SL activates Write Gate at the beginning of the sync field under the conventional mode. The controller then writes a new sync field, data address mark, data field, and CRC as shown in Figure 6-1. With the pre-erase head of the perpendicular drive, the write head must be activated in the Gap2 field to insure a proper write of the new sync field. For the 1 Mbps perpendicular mode (WGATE = 1, GAP = 1), 38 bytes will be written in the Gap2 space. Since the bit density is proportional to the data rate, 19 bytes will be written in the Gap2 field for the 500 Kbps perpendicular mode (WGATE = 1, GAP = 0).

It should be noted that none of the alterations in Gap2 size, VCO timing, or Write Gate timing affect normal program flow. The information provided here is just for background purposes and is not needed for normal operation. Once the PERPENDICULAR MODE command is invoked, 82077SL software behavior from the user standpoint is unchanged.

Table 6-11. Effects of WGATE and GAP Bits

GAP	WGATE	MODE	VCO Low Time after Index Pulse	Length of Gap2 Format Field	Portion of Gap2 Written by Write Data Operation	Gap2 VCO Low Time for Read Operations
0	0	Conventional Mode	33 Bytes	22 Bytes	0 Bytes	24 Bytes
0	1	Perpendicular Mode (500 Kbps Data Rate)	33 Bytes	22 Bytes	19 Bytes	24 Bytes
1	0	Reserved (Conventional)	33 Bytes	22 Bytes	0 Bytes	24 Bytes
1	1	Perpendicular Mode (1 Mbps Data Rate)	18 Bytes	41 Bytes	38 Bytes	43 Bytes

NOTE:
When either GAP or WGATE bit is set, the current value of precompensation in the DSR is used.

6.2.12 POWERDOWN MODE COMMAND

The POWERDOWN MODE command allows the automatic power management of the 82077SL. This especially allows the extension of battery life in portable PC systems. This command should be issued during the BIOS power on self test (POST) to enable auto powerdown.

As soon as the command is enabled, a 10 ms or a 0.5 sec minimum power up timer is initiated depending on whether the MIN DLY bit is set to 0 or 1. This timer is one of the required conditions that has to be satisfied before the part will enter auto powerdown. Any software reset will reinitialize the timer. The timer countdown is also extended by up to 10 ms if the data rate is changed during the timer's countdown. Without this timer 82077SL would have been put to sleep immediately after 82077SL is idle. The minimum delay gives software a chance to interact with 82077SL without incurring an additional overhead due to recovery time.

The command also allows the output pins of floppy disk drive interface to be tristated or left unaltered during auto powerdown. This is done by the FDI TRI bit. In the default condition (FDI TRI = 0) the output pins of the floppy disk drive are tristated. Setting this bit leaves the interface unchanged from the normal state.

The results phase returns the values programmed for MIN DLY, FDI TRI and AUTO PD. The results phase of the auto powerdown mode command has its two most significant bits set to zero to distinguish it from the 82077AA's command of the same value which returns an "Illegal Command" status of 80H. The auto powerdown mode is disabled by a hardware reset. Software results have no effect on the POWERDOWN MODE command parameters.

6.3 Command Set Enhancements

The PERPENDICULAR MODE and DUMPREG commands were enhanced along with the addition of a new LOCK command in the 82077AA. These enhancements also hold for the 82077SL and are explained in this section of the data sheet. The commands were enhanced/added in order to provide protection against older software application package which could inadvertently cause system compatibility problems. The modifications/additions are fully backward compatible with the 82077AA which do not support the enhancements.

6.3.1 PERPENDICULAR MODE

The PERPENDICULAR MODE Command is enhanced to allow the system designers to designate specific drives as Perpendicular recording drives. This enhancement is made so that the system designer does not have to worry about older application software packages which bypass their system's FDC (Floppy Disk Controller) routines. The enhancement will also allow data transfers between Conventional and Perpendicular drives without having to issue PERPENDICULAR MODE commands between the accesses of the two different drives, nor having to change write pre-compensation values. The following is an explanation of how this enhancement is implemented:

With the old implementation, the user must properly program both the PERPENDICULAR MODE command and write pre-compensation value before accessing either a Conventional or Perpendicular drive. These programmed values apply to all drives (D0–D3) which the 82077SL may access. It should also be noted that any form of RESET "Hardware" or "Software" will configure the PERPENDICULAR MODE command for Conventional mode (GAP and WGATE = "0").

With the enhanced implementation, both the GAP and WGATE bits have the same affects as the old implementation except for when they are both programmed for value of "0" (Conventional mode). For the case when both GAP and WGATE equal "0" the PERPENDICULAR MODE command will have the following effect on the 82077SL: 1) If any of the new bits D0, D1, D2, and D3 are programmed to "1" the corresponding drive will automatically be programmed for Perpendicular mode (ie: GAP2 being

Old PERPENDICULAR MODE command:

Phase	R/W	Data Bus								Remarks
		D7	D6	D5	D4	D3	D2	D1	D0	
PERPENDICULAR MODE										
Command	W	0	0	0	1	0	0	1	0	Command
	W	0	0	0	0	0	0	GAP	WGATE	Code

NOTE:
For the definition of GAP and WGATE bits see Table 6-11 and Section 6.2.11 of the data sheet.
For the Enhanced PERPENDICULAR MODE command definition see Table 6-1.

written during a write operation, the programmed Data Rate will determine the length of GAP2.), and data will be written with 0 ns write pre-compensation. 2) any of the new bits (D0–D3) that are programmed for "0" the designated drive will be programmed for Conventional Mode and data will be written with the currently programmed write pre-compensation value. 3) Bits D0, D1, D2, and D3 can only be over written when the OW bit is written as a "1". The status of these bits can be determined by interpreting the eighth result byte of the enhanced DUMPREG Command (See Section 6.3.3). (Note: if either the GAP or WGATE bit is a "1", then bits D0–D3 are ignored.)

"Software" and "Hardware" RESET will have the following effects on the enhanced PERPENDICULAR MODE command:

1) "Software" RESETs (Reset via DOR or DSR registers) will only clear GAP and WGATE bits to "0", D3, D2, D1, and D0 will retain their previously programmed values.
2) "Hardware" RESETs (Reset via pin 32) will clear all bits (GAP, Wgate, D0, D1, D2, and D3) to "0" (All Drives Conventional Mode).

6.3.2 LOCK

In order to protect a system with long DMA latencies against older application software packages that can disable the 82077SL's FIFO the following LOCK Command has been has been retained in the 82077SL's command set: [Note: This command should only be used by the system's FDC routines, and ISVs (Independent Software Vendors) should refrain from using it. If an ISV's application calls for having the 82077SL FIFO disabled a CONFIGURE Command should be used to toggle the EFIFO (Enable FIFO) bit. ISV can determine the value of the LOCK bit by interpreting the eighth result byte of an DUMPREG Command (See Section 6.3.3).]

The LOCK command defines whether EFIFO, FIFOTHR, and PRETRK parameters of the CONFIGURE command can be RESET by the DOR and DSR registers. When the LOCK bit is set to a "1" all subsequent "software" RESETs by the DOR and DSR registers will not change the previously set parameter values in the CONFIGURE command. When the LOCK bit is set to a "0" "software" RESETs by the DOR or DSR registers will return these parameters to their default values (See Section 6.2.7). All "hardware" Resets by pin 32 will set the LOCK bit to a "0" value, and will return EFIFO, FIFOTHR, and PRETRK to their default values. A Status byte is returned immediately after issuing the command byte. This Status byte reflects the value of the Lock bit set by the command byte. (Note: No interrupts are generated at the end of this command.)

6.3.3 ENHANCED DUMPREG COMMAND

To accommodate the new LOCK command and enhanced PERPENDICULAR MODE command the eighth result byte of DUMPREG command has been modified in the following manner:

Phase	R/W	Data Bus								Remarks
		D7	D6	D5	D4	D3	D2	D1	D0	
		DUMPREG								
Result	R	Eighth Result Byte — Undefined —								Old
	R	LOCK	0	D3	D2	D1	D0	GAP	WGATE	Enhanced

NOTES:
1. Data bit 7 reflects the status of the new LOCK bit set by the LOCK Command.
2. Data Bits D0–D5 reflect the status for bits D3, D2, D1, D0, GAP and WGATE set by the PERPENDICULAR MODE Command.

7.0 STATUS REGISTER ENCODING

The contents of these registers are available only through a command sequence.

7.1 Status Register 0

Bit No.	Symbol	Name	Description
7, 6	IC	Interrupt Code	00-Normal termination of command. The specified command was properly executed and completed without error. 01-Abnormal termination of command. Command execution was started, but was not successfully completed. 10-Invalid command. The requested command could not be executed. 11-Abnormal termination caused by Polling.
5	SE	Seek End	The 82077SL completed a SEEK or RECALIBRATE command, or a READ or WRITE with implied seek command.
4	EC	Equipment Check	The TRK0 pin failed to become a "1" after: 1. 80 step pulses in the RECALIBRATE command. 2. The RELATIVE SEEK command causes the 82077SL to step outward beyond Track 0.
3	—	—	Unused. This bit is always "0".
2	H	Head Address	The current head address.
1, 0	DS1, 0	Drive Select	The current selected drive.

7.2 Status Register 1

Bit No.	Symbol	Name	Description
7	EN	End of Cylinder	The 82077SL tried to access a sector beyond the final sector of the track (255D). Will be set if TC is not issued after Read or Write Data Command.
6	—	—	Unused. This bit is always "0".
5	DE	Data Error	The 82077SL detected a CRC error in either the ID field or the data field of a sector.
4	OR	Overrun/ Underrun	Becomes set if the 82077SL does not receive CPU or DMA service within the required time interval, resulting in data overrun or underrun.
3	—	—	Unused. This bit is always "0".
2	ND	No Data	Any one of the following: 1. READ DATA, READ DELETED DATA command, the 82077SL did not find the specified sector. 2. READ ID command, the 82077SL cannot read the ID field without an error. 3. READ TRACK command, the 82077SL cannot find the proper sector sequence.
1	NW	Not Writable	WP pin became a "1" while the 82077SL is executing a WRITE DATA, WRITE DELETED DATA, or FORMAT TRACK command.
0	MA	Missing Address Mark	Any one of the following: 1. The 82077SL did not detect an ID address mark at the specified track after encountering the index pulse from the IDX pin twice. 2. The 82077SL cannot detect a data address mark or a deleted data address mark on the specified track.

7.3 Status Register 2

Bit No.	Symbol	Name	Description
7	—	—	Unused. This bit is always "0".
6	CM	Control Mark	Any one of the following: 1. READ DATA command, the 82077SL encounters a deleted data address mark. 2. READ DELETED DATA command, the 82077SL encounters a data address mark.
5	DD	Data Error in Data Field.	The 82077SL detected a CRC error in the data field.
4	WC	Wrong Cylinder	The track address from the sector ID field is different from the track address maintained inside the 82077SL.
3	—	—	Unused. This bit is always "0".
2	—	—	Unused. This bit is always "0".
1	BC	Bad Cylinder	The track address from the sector ID field is different from the track address maintained inside the 82077SL and is equal to FF hex which indicates a bad track with a hard error according to the IBM soft-sectored format.
0	MD	Missing Data Address Mark	The 82077SL cannot detect a data address mark or a deleted data address mark.

7.4 Status Register 3

Bit No.	Symbol	Name	Description
7	—	—	Unused. This bit is always "0".
6	WP	Write Protected	Indicates the status of the WP pin.
5	—	—	Unused. This bit is always "1".
4	T0	TRACK 0	Indicates the status of the TRK0 pin.
3	—	—	Unused. This bit is always "1".
2	HD	Head Address	Indicates the status of the HDSEL pin.
1, 0	DS1, 0	Drive Select	Indicates the status of the DS1, DS0 pins.

8.0 COMPATIBILITY

The 82077SL was designed with software compatibility in mind. It is a fully backwards compatible solution with the older generation 8272A and NEC765A/B disk controllers. The 82077SL also implements on-board registers for compatibility with the Personal System/2s as well as PC/AT and PC/XT floppy disk controller subsystems. The 82077SL is fully compatible with Intel's 386SL Microprocessor Superset. The 82077SL represents a superset of features that are available on 82077AA. Upon a hardware reset of the 82077SL, all registers, functions and enhancements default to a PS/2, PC/AT, or PS/2 Model 30 compatible operating mode depending on how the IDENT and MFM pins are sampled during Hardware Reset.

8.1 Register Set Compatibility

The register set contained within the 82077SL is a culmination of hardware registers based on the architectural growth of the IBM personal computer line. Table 8-1 indicates the registers required for compatibility based on the type of computer.

Table 8-1. 82077SL Register Support

82077SL Register	8272A	82072	PC/XT	PC/AT	PS/2	Mod 30
SRA					X	X
SRB					X	X
DOR			X	X	X	X
MSR	X	X	X	X	X	X
DSR		X				
Data (FIFO)	X	X	X	X	X	X
DIR				X	X	X
CCR		X*		X	X	X

*CCR is emulated by DSR in an 82072 PC/AT design.

8.2 PS/2 vs. AT vs. Model 30 Mode

To maintain compatibility between PS/2, PC/AT, and Model 30 environments the IDENT and MFM pins are provided. The 82077SL is placed into the proper mode of operations upon Hardware RESET with the appropriate settings of the IDENT and MFM pins. The proper settings of the IDENT and MFM pins are described in IDENT's pin description. Differences between the three modes are described in the following sections.

8.2.1 PS/2 MODE

IDENT strapped low causes the polarity of DENSEL to be active low for high (500 Kbps/1 Mbps) data rates (typically used for 3.5″ drives). This polalrity of DENSEL assumes INVERT# to be low. A comprehensive description of DENSEL behavior is given in Table 2-6.

The $\overline{\text{DMAGATE}}$ bit in the Digital Output Register (DOR) will not cause the DRQ or INT output signals to tristate. This maintains consistency with the operation of the floppy disk controller subsystem in the PS/2 architecture.

TC is an active low input signal that is internally qualified by $\overline{\text{DACK}}$ being active low.

8.2.2 PC/AT MODE

IDENT strapped high causes the polarity of DENSEL to be active high for high (500 Kbps/1 Mbps) data rates (typically used for 5.25″ drives). This polarity of DENSEL assumes INVERT# to be low. A comprehensive description of DENSEL behavior is given in Table 2-6.

If the $\overline{\text{DMAGATE}}$ bit is written to a "0" in the Digital Output Register (DOR), DRQ and INT will tristate. If $\overline{\text{DMAGATE}}$ is written to a "1", then DRQ and INT will be driven appropriately by the 82077SL.

TC is an active high input signal that is internally qualified by DACK# being active low.

8.2.3 MODEL 30 MODE

IDENT strapped low causes the polarity of DENSEL to be active low for high (500 Kbps/1 Mbps) data rates (typically used for 3.5″ drives). This polarity of DENSEL assumes INVERT# to be low. A comprehensive description of DENSEL behavior is given in Table 2-6.

$\overline{\text{DMAGATE}}$ and TC function the same as in PC/AT Mode.

8.3 Compatibility with the FIFO

The FIFO of the 82077SL is designed to be transparent to non-FIFO disk controller software developed on the older generation 8272A standard. Operation of the 82077SL FIFO can be broken down into two tiers of compatibility. For first tier compatibility, the FIFO is left in the default disabled condition upon a "Hardware" reset (via pin 32). In this mode the FIFO operates in a byte mode and provides complete compability with non-FIFO based software. For second tier compatibility, the FIFO is enabled via the CONFIGURE command. When the FIFO is enabled, it will temporarily enter a byte mode during the command and result phase of disk controller operation. This allows for compatible operation when interrogating the Main Status Register (MSR) for the purpose of transferring a byte at a time to or from the disk controller. For normal disk controller applications, the system designer can still take advantage of the FIFO for time critical data transfers during the execution phase and not create any conflicts with non-FIFO software during the command or result phase.

In some instances, use of the FIFO in any form has conflicted with certain specialized software. An example of a compatibility conflict using the FIFO is with software that monitors the progress of a data transfer during the execution phase. If the software assumed the disk controller was operating in a single byte mode and counted the number of bytes transferred to or from the disk controller to trigger some time dependent event on the disk media (i.e. head position over a specific data field), the same software will not have an identical time relationship if the FIFO is enabled. This is because the FIFO allows data to be queued up, and then burst trans-

ferred across the host bus. To accommodate software of this type, it is recommended that the FIFO be disabled.

8.4 Drive Polling

The 82077SL supports the polling mode of the older generation 8272A. This mode is enabled upon a reset and can be disabled via the CONFIGURE command. This mode is supported for the sole purpose of providing backwards compatibility with software that expects it's presence.

The intended purpose of drive polling dates back to 8" drives as a means to monitor any change in status for each disk drive present in the system. Each of the drives is selected for a period of time and its READY signal sampled. After a delay, the next drive is selected. Since the 82077SL does not support READY in this capacity (internally tied true), the polling sequence is only simulated and does not affect the drive select lines (DS0–DS3) when it is active. If enabled, it occurs whenever the 82077SL is waiting for a command or during SEEKs and RECALIBRATEs (but not IMPLIED SEEKs). Each drive is assumed to be not ready after a reset and a "ready" value for each drive is saved in an internal register as the simulated drive is polled. An interrupt will be generated on the first polling loop because of the initial "not ready" status. This interrupt must be followed with a SENSE INTERRUPT STATUS command from the host to clear the interrupt condition for each of the four logical drives.

9.0 PROGRAMMING GUIDELINES

Programming the 82077SL is identical to any other 8272A compatible disk controller with the exception of some additional commands. For the new designer it is useful to provide some guidelines on how to program the 82077SL. A typical disk operation involves more than issuing a command and waiting for the results. The control of the floppy disk drive is a low level operation that requires software intervention at different stages. New commands and features have been added to the 82077SL to reduce the complexity of this software interface.

9.1 Command and Result Phase Handshaking

Before a command or parameter byte can be issued to the 82077SL, the Main Status Register (MSR) must be interrogated for a ready status and proper FIFO direction. A typical floppy controller device driver should contain a subroutine for sending command or parameter bytes. For this discussion, the routine will be called "Send_byte" with the flowchart shown in Figure 9-1.

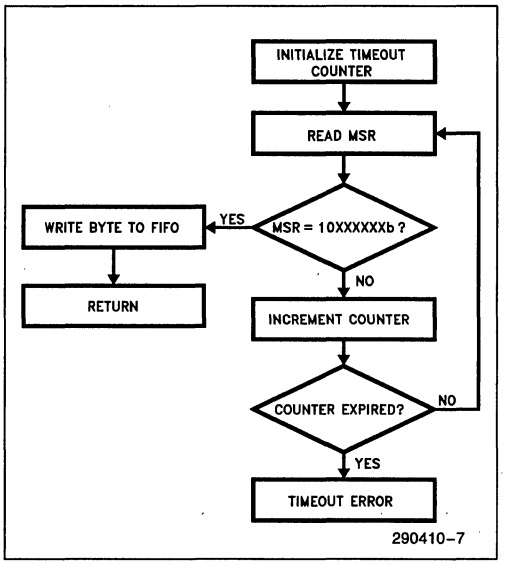


Figure 9-1. Send_Byte Routine

The routine loops until RQM is 1 and DIO is 0 indicating a ready status and FIFO direction is inward. If this condition is true, the 82077SL is ready to accept a command or parameter byte. A timeout counter is used to insure software response within a reasonable amount of time in case of no response by the 82077SL. As a note, the programmer must be careful how the maximum delay is chosen to avoid unnecessary timeouts. For example, if a new command is issued when the 82077SL is in the middle of a polling routine, the MSR will not indicate a ready status for the next parameter byte until the polling sequence completes the loop. This could cause a delay between the first and second bytes of up to 250 μs (@ 250 Kbps). If polling is disabled, this maximum delay is 175 μs. There should also be enough timeout margin to accommodate a shift of the software to a higher speed system. A timeout value that results in satisfactory operation on a 16 MHz CPU might fail when the software is moved to a system with a 25 MHz CPU. A recommended solution is to derive the timeout counter from a system hardware counter that is fixed in frequency from CPU clock to CPU clock.

For reading result bytes from the 82077SL, a similar routine is used. Figure 9-2 illustrates the flowchart for the routine "Get_byte". The MSR is polled until

RQM is 1 and DIO is 1, which indicates a ready status and outward FIFO direction. At this point, the host can read a byte from the FIFO. As in the Send_byte routine, a timout counter should be incorporated in case of a disk controller lock-up condition. For example, if a disk was not inserted into the disk drive at the time of a read operation, the controller would fail to receive the index pulse and lock-up since the index pulses are required for termination of the execution phase.

9.2 Initialization

Initializing the 82077SL involves setting up the appropriate configuration after a reset. Parameters set by the SPECIFY command are undefined after a system reset and will need to be reinitialized. CONFIGURE command parameters default to a known state after a system reset but will need to be reinitialized if the system requirements are different from the default settings. The flowchart for the recommended initialization sequence of the 82077SL is shown in Figure 9-3.

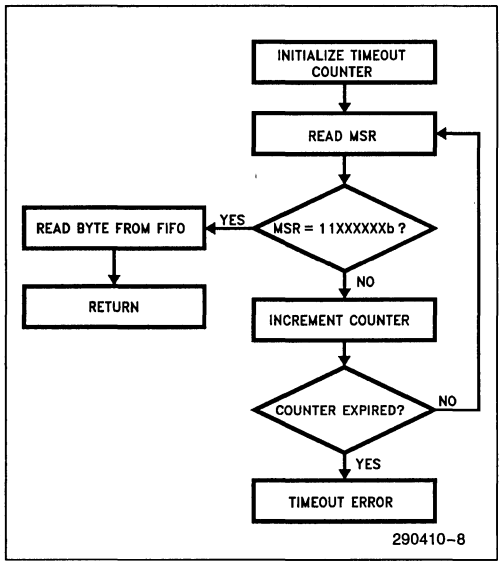


Figure 9-2. Get_Byte Routine

Following a reset of the 82077SL, the Configuration Control Register (CCR) should be reinitialized for the appropriate data rate. An external reset via the RESET pin will cause the data rate and write precompensation values to default to 250 Kbps (10b) and 125 ns (000b) respectively. Since the 125 ns write precompensation value is optimal for the 5¼" and 3½" disk drive environment, most applications will not require the value to be changed in the initialization sequence. As a note, a software reset issued via the DOR or DSR will not affect the data rate or write precompensation values. But it is recommended as a safe programming practice to always program the data rate after a reset, regardless of the type.

Since polling is enabled after a reset of the 82077SL, four SENSE INTERRUPT STATUS commands need to be issued afterwards to clear the status flags for each drive. The flowchart in Figure 9-3 illustrates how the software clears each of the four interrupt status flags internally queued by the 82077SL. It should be noted that although four SENSE INTERRUPT STATUS commands are issued, the INT pin is only active until the first SENSE INTERRUPT STATUS command is executed.

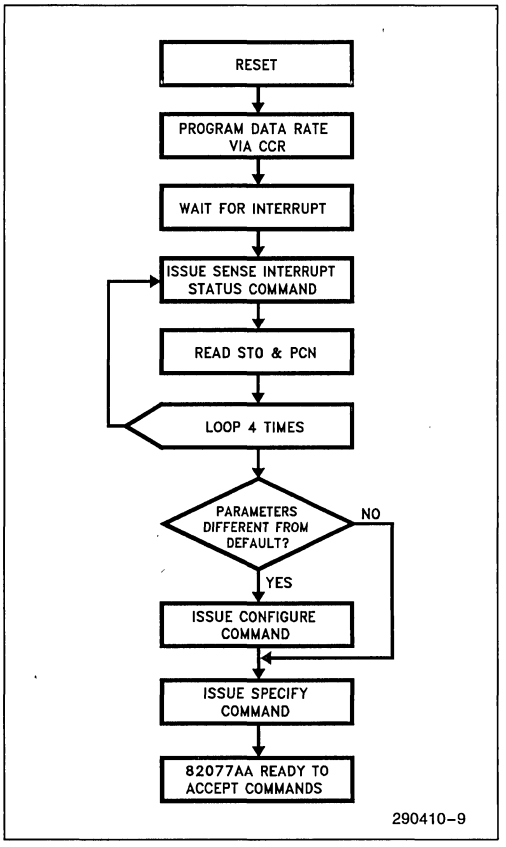


Figure 9-3. Initialization Flowchart

As a note, if the CONFIGURE command is issued within 250 μs of the trailing edge of reset (@ 1 Mbps), the polling mode of the 82077SL can be disabled before the polling initiated interrupt occurs. Since polling stops when the 82077SL enters the command phase, it is only time critical up to the first byte of the CONFIGURE command. If disabled in time, the system software no longer needs to issue the four SENSE INTERRUPT STATUS commands to clear the internal interrupt flags normally caused by polling.

The CONFIGURE command should also be issued if the system requirements are different from the default settings (as described in Section 6.2.7). For example, the CONFIGURE command can be used to enable the FIFO, set the threshold, and enable Implied Seeks.

The non-DMA mode flag, step rate (SRT), head load (HLT), and head unload times (HUT) programmed by the SPECIFY command do not default to a known state after a reset. This behavior is consistent with the 8272A and has been preserved here for compatibility. Thus, it is necessary to always issue a SPECIFY command in the initialization routine.

9.3 Recalibrates and Seeks

Commands that position the disk head are different from the typical READ/WRITE/FORMAT command in the sense that there is no result phase. Once a RECALIBRATE, SEEK, or RELATIVE SEEK command has been issued, the 82077SL will return a ready status in the Main Status Register (MSR) and perform the head positioning operation as a background task. When the seek is complete, the 82077SL will assert the INT signal to request service. A SENSE INTERRUPT STATUS command should then be asserted to clear the interrupt and read the status of the operation. Since the drive and motor enable signals are directly controlled through the Digital Output Register (DOR) on the 82077SL, a write to the DOR will need to precede the RECALIBRATE or SEEK command if the drive and motor is not already enabled. Figure 9-4 shows the flow chart for this operation.

9.4 Read/Write Data Operations

A read or write data operation requires several steps to complete successfully. The motor needs to be turned on, the head positioned to the correct cylinder, the DMA controller initialized, the read or write command initiated, and an error recovery scheme implemented. The flowchart in Figure 9-5 highlights a recommended algorithm for performing a read or write data operation.

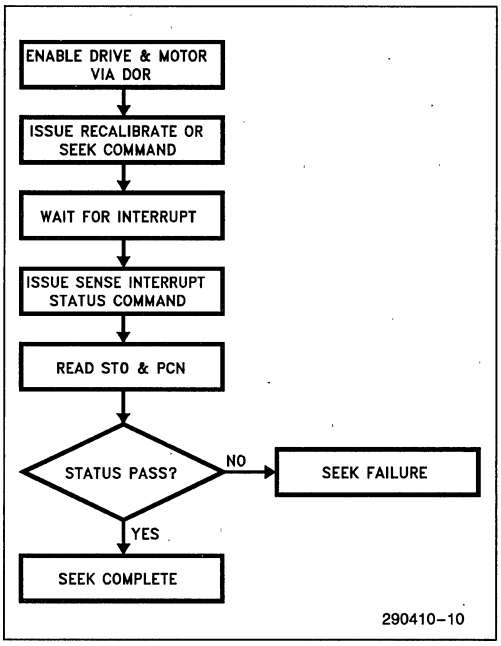


Figure 9-4. Recalibrate and Seek Operations

Before data can be transferred to or from the diskette, the disk drive motor must be brought up to speed. For most 3½" disk drives, the spin-up time is 300 ms, while the 5¼" drive usually requires about 500 ms due to the increased moment of inertia associated with the larger diameter diskette.

One technique for minimizing the motor spin-up delay in the read data case is to begin the read operation immediately after the motor is turned on. When the motor is not initially up to speed, the internal data separator will fail to lock onto the incoming data stream and report a failure in the status registers. The read operation is then repeated until successful status is obtained. There is no risk of a data integrity problem since the data field is CRC validated. But, it is not recommended to use this technique for the write data operation even though it requires successful reading of the ID field before the write takes place. The data separator performance of the 82077SL is such that locking to the data stream could take place while the motor speed variation is still significant. This could result in errors when an attempt is made to read the disk media by other disk controllers that have a narrower incoming data stream frequency bandwidth.

After the motor has been turned on, the matching data rate for the media inserted into the disk drive should then be programmed to the 82077SL via the Configuration Control Register (CCR). The 82077SL

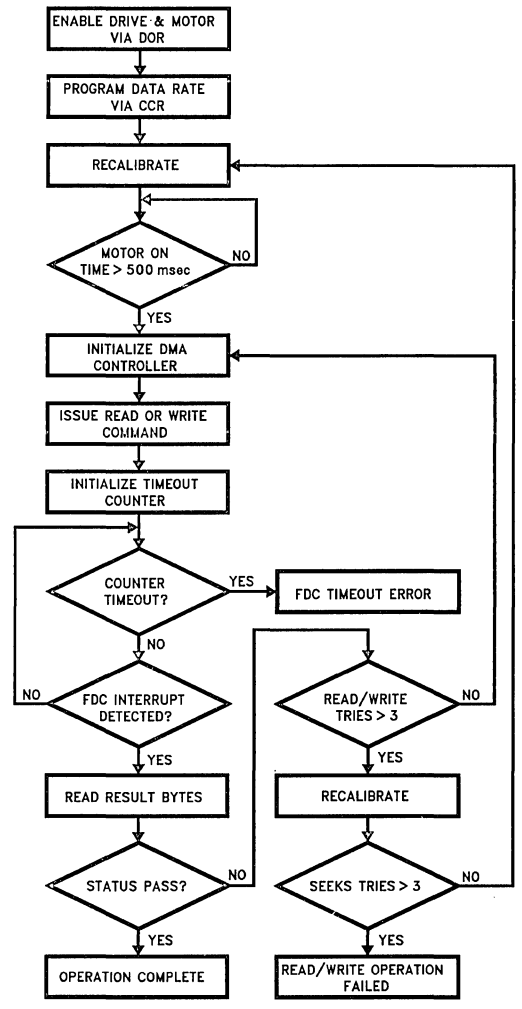


Figure 9-5. Read/Write Operation

is designed to allow a different data rate to be programmed arbitrarily without disrupting the integrity of the device. In some applications, it is required to automatically determine the recorded data rate of the inserted media. One technique for doing this is to perform a READ ID operation at each available data rate until a successful status is returned in the result phase.

If implied seeks are not enabled, the disk drive head must be positioned over the correct cylinder by executing a SEEK command. After the seek is complete, a head settling time needs to be asserted before the read or write operation begins. For most drives, this delay should be a minimum of 15 ms. When using implied seeks, the minimum head settling time can be enforced by the head load time (HLT) parameter designated in the SPECIFY command. For example, a HLT value of 8 will yield an effective head settling time of 16 ms for a programmed data rate of 500 Kbps. Of course if the head is already positioned over the correct cylinder, the head settling time does not need to be enforced.

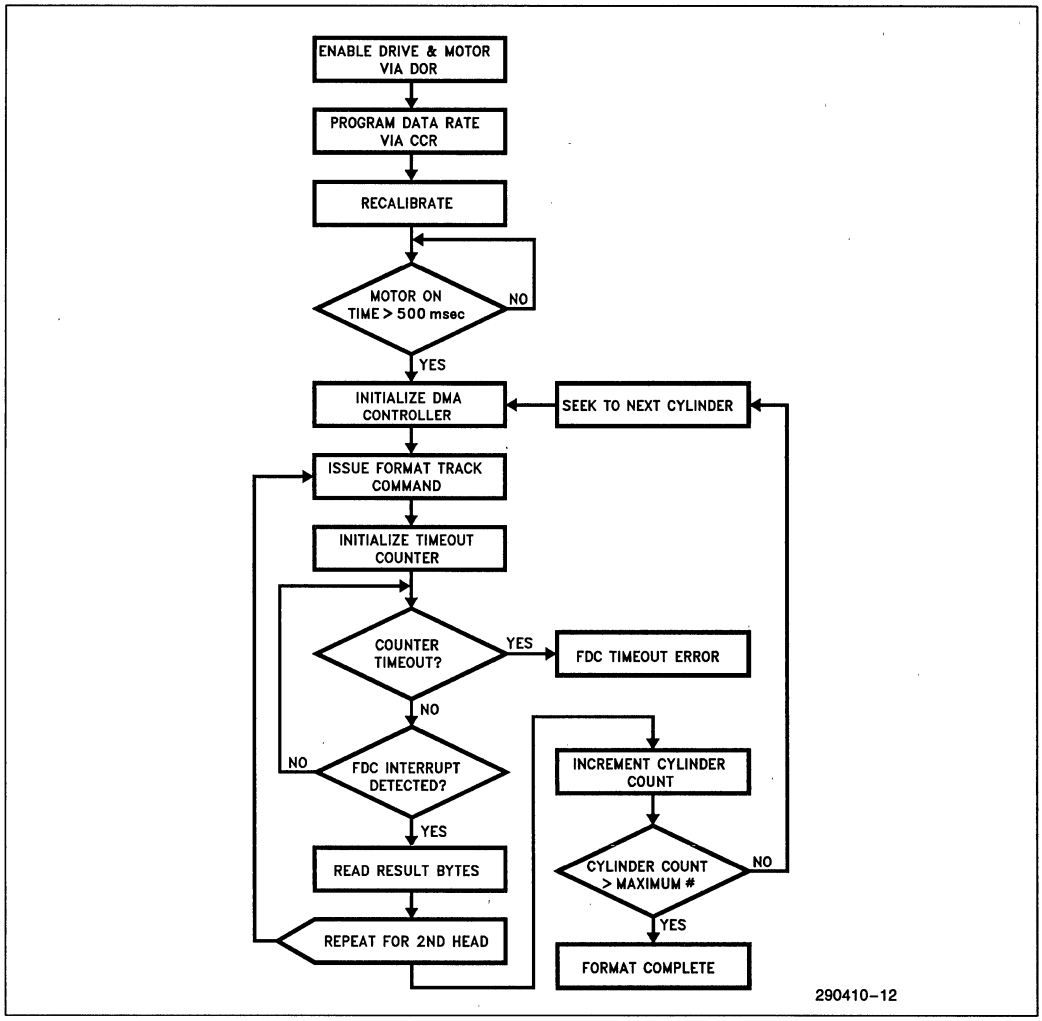


Figure 9-6 Formatting

The DMA controller is then initialized for the data transfer and the read or write command is executed. Typically the DMA controller will assert Terminal Count (TC) when the data transfer is complete. The 82077SL will then complete the current data transfer and assert the INT signal signifying it has entered the result phase. The result phase can also be entered by the 82077SL if an error is encountered or the last sector number equals the End of Track (EOT) parameter.

Based on the algorithm in Figure 9-5, if an error is encountered after reading the result bytes, two more retries are performed by reinitializing the DMA controller and re-issuing the read or write data command. A persisting failure could indicate the seek operation did not achieve proper alignment between the head and the track. The disk head should then be recalibrated and the seek repeated for a maximum of two more tries. Unsuccessful operation after this point should be reported as a disk failure to the operating system.

9.5 Formatting

The disk formatting procedure involves positioning the head on each track and creating a fixed format field used for organizing the data fields. The flowchart in Figure 9-6 highlights the typical format procedure.

After the motor has been turned on and the correct data rate programmed, the disk head is recalibrated to track 0. The disk is then allowed to come up to speed via a 500 ms delay. It is important the disk speed has stabilized before the actual formatting to avoid any data rate frequency variations. Since the format fields contain critical information used by the data separator of the disk controller for synchronization purposes, frequency stability of the data stream is imperative for media interchangeability among different systems.

The ID field data created on the disk during the format process is provided by the DMA controller during the execution phase. The DMA controller is initialized to send the C, H, R and N values for each sector ID field. For example, to format cylinder 7, on head 1, with 9 sectors, and a sector size of 2 (512 bytes), the DMA controller should be programmed to transfer 36 bytes (9 sectors x 4 bytes per sector) with the following data field: 7,1,1,2, 7,1,2,2, 7,1,3,2, ... 7,1,9,2. Since the values provided to the 82077SL during the execution phase of the format command are directly recorded as the ID fields on the disk, the data contents can be arbitrary. Some forms of copy protection have been implemented by taking advantage of this capability.

After each head for a cylinder has been formatted, a seek operation to the next cylinder is performed and the format process is repeated. Since the FORMAT TRACK command does not have implied seek capability, the SEEK command must be used. Also, as discussed in Section 9-2, the head settling time needs to be adhered to after each seek operation.

9.6 Verifies

In some applications, the sector data needs to be verified immediately after each write operation. The verify technique historically used with the 8272A or 82072 disk controller involved reinitializing the DMA controller to perform a read transfer or verify transfer (DACK# is asserted but not RD#) immediately after each write operation. A read command is then to be issued to the disk controller and the resulting status indicates if the CRC validated the previously written data. This technique has the drawback of requiring additional software intervention by having to reprogram the DMA controller between each sector write operation. The 82077SL supports this older verify technique but also provides a new VERIFY command that does not require the use of the DMA controller. This is also available in 82077AA.

To verify a write data transfer or format track operation using the VERIFY command, the software simply issues the command with the same format as a READ DATA command but without the support of the DMA controller. The 82077SL will then perform a disk read operation without a host data transfer. The CRC will be calculated for each sector read and compared against the value stored on the disk. When the VERIFY command is complete, the status register will report any detected CRC errors.

9.7 Powerdown State and Recovery

The two power management modes coupled with the internal oscillator power management forms an important consideration for programming the 82077SL. The recovery of 82077SL and the time it takes to achieve complete recovery depends on how 82077SL is powered down and how it is awakened. The following sections describe all the programming concerns and subtleties involved in using power management features of the 82077SL.

2

9.7.1 OSCILLATOR POWER MANAGEMENT

Section 4.1 covers the power management scheme involved in powering down of both an internal and an external oscillator. Both types of oscillators face drop out effects and require recovery times on the order of tens of milliseconds (this may be objectionable to some application software). This means that if the oscillator is powered down then it is imperative for the software to assure enough time for the oscillator to recover to a stable state. Oscillator power management must be controlled by the system software especially to maintain software transparency. In cases where the system goes into a standby mode (by user request or system timeout), the power management software can turn off the oscillator to conserve power. Complete recovery from an oscillator powerdown state requires the software to turn on the oscillator sufficiently ahead of awakening the 82077SL.

9.7.2 PART POWER MANAGEMENT

The part powerdown and wake up modes are covered in Section 4.2 in detail. This section is meant to address the programming concerns for the part (excluding the oscillator) during these modes.

9.7.2.a Powerdown Modes

For both types of powerdown modes—DSR powerdown and auto powerdown, if reset is used to exit the part from powerdown then the internal microcontroller will go through a standard sequence: register initialization followed after some delay by an interrupt.

Software transparency in auto powerdown mode is preserved by MSR retaining the value of 80H which indicates that the part is ready to receive a command. This feature allows the part to powerdown while maintaining its responsiveness to any application software.

9.7.2.b Wake Up Modes

Wake up from DSR powerdown results in the part being internally reset and all present status being lost. During DSR powerdown the RQM bit in the MSR is set. A software or hardware reset will wake up the part.

The case for wake up from auto powerdown is different. The BIOS and application software are very sensitive to delays involved in writing the first command bytes to the 82077SL. Most programs have short error timeouts in these cases. Such programs would not tolerate any floppy disk controller that was unable to receive the first byte of a command at any time. The following describes how 82077SL uniquely sustains its software transparency during wake up sequences.

Prior to writing a command to 82077SL, it is first necessary to read the MSR to ensure that the 82077SL is ready (RQM bit must be set) to receive the command. When the part detects a MSR read, it assumes that another command will follow and begins the wake up process. While the part is waking up it does not change the state of the MSR (MSR = 80H) and is able to receive the command in the FIFO. At this point one of the two following scenarios can occur.

— No other command is sent subsequent to the MSR read. The part wakes up and initializes the minimum power up timer. Upon the expiration of this timer the part is once again put in powerdown state.

— Another command follows the MSR read. If the command is send during the part's recovery from powerdown, the part remembers the command, clears the RQM bit (to prevent further bytes being written) and acts on the command once it is fully awake.

If the MSR was not checked prior to writing of a command, the part will proceed as stated above with the RQM bit cleared and the command byte held until the internal microcontroller is ready. Writing the motor enable bits in DOR active will initiate the wake up sequence with RQM set high, ready to receive any command.

As it is clear from the above discussion, the immediate access to the floppy disk controller for the first command byte is vital to software transparency. The recovery of the part from powerdown may involve a delay after the first command byte has been issued. However, all programs have tolerance for the delay after the first command byte is issued. In a powered up chip, it is possible for the microcontroller to be in its "polling loop". As a result the tolerance for this delay provides an excellent window for recovery of the part.

10.0 DESIGN APPLICATIONS

10.1 PC/AT Floppy Disk Controller

This section presents a design application of a PC/AT compatible floppy disk controller. With an 82077SL, a 24 MHz crystal, a resistor package, and a device chip select, a complete floppy disk controller can be built. The 82077SL integrates all the necessary building blocks for a reliable and low cost solution. But before we discuss the design application using the 82077SL, it is helpful to describe the architecture of the original IBM PC/AT floppy disk controller design that uses the 8272A.

10.1.1 PC/AT FLOPPY DISK CONTROLLER ARCHITECTURE

The standard IBM PC/AT floppy disk controller using the 8272A requires 34 devices for a complete solution. The block diagram in Figure 10-1 illustrates the complexity of the disk controller. A major portion of this logic involves the design of the data separator. The reliability of the disk controller is primarily dictated by the performance and stability of the data separator. Discrete board level analog phase lock loops generally offer good bit jitter margins but suffer from instability and tuning problems in the manufacturing stage if not carefully designed. While digital data separator designs offer stability and generally a lower chip count, they suffer from poor performance in the recovery of data.

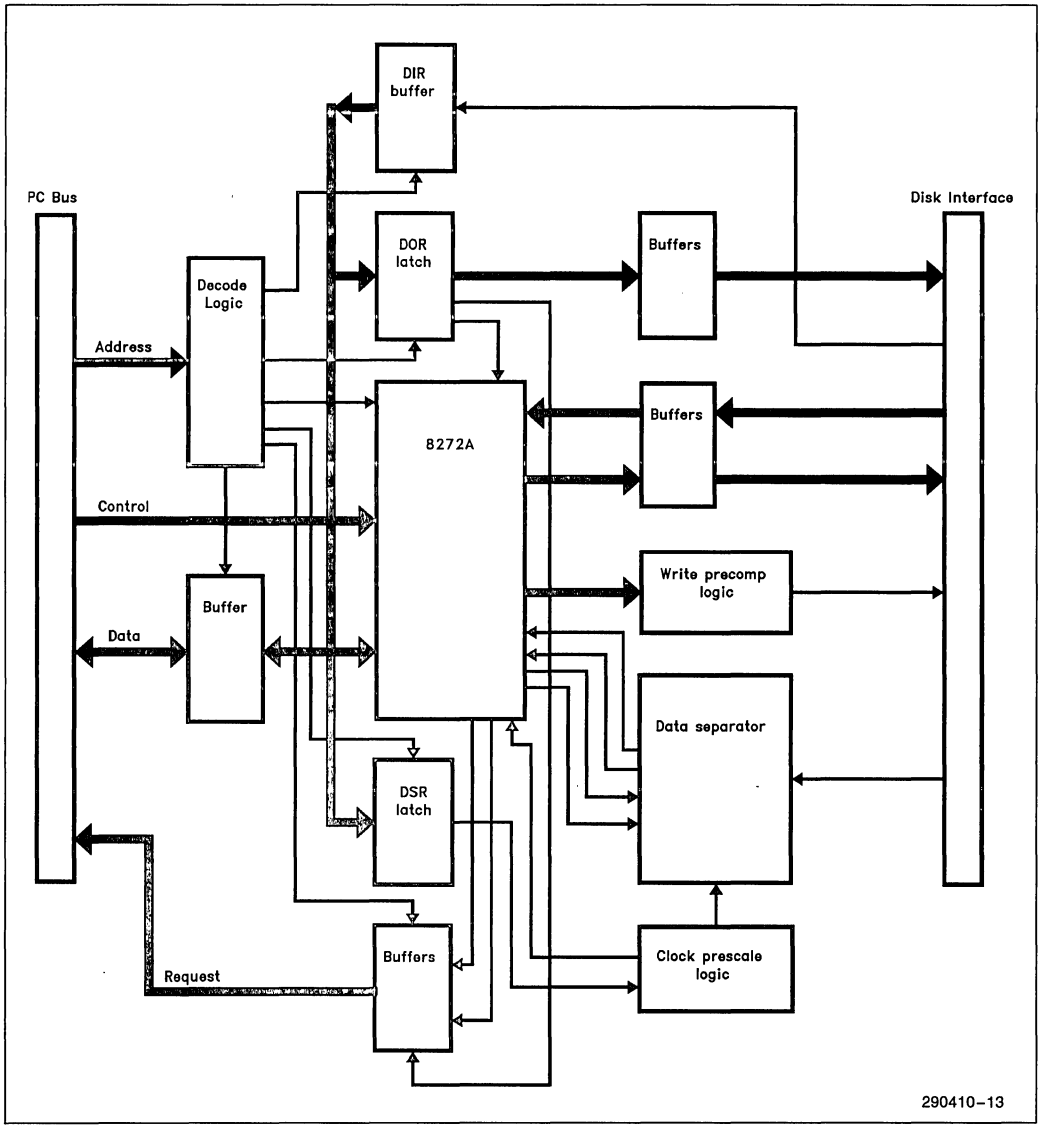


Figure 10-1. Standard IBM PC/AT Floppy Disk Controller

Table 10-1 indicates the drive and media types the IBM PC/AT disk controller can support. This requires the data separator to operate at three different data rates: 250 Kbps, 300 Kbps and 500 Kbps. Clocks to the data separator and disk controller need to be prescaled correspondingly to accommodate each of these data rates. The clock prescaling is controlled by the Data rate Select Register (DSR). Supporting all three data rates can compromise the performance of the phase lock loop (PLL) if steps are not taken in the design to adjust the performance parameters of the PLL with the data rate.

Table 10-1. Standard PC/AT Drives and Media Formats

Capacity	Drive Speed	Data Rate	Sectors	Cylinders
360 Kbyte	300 RPM	250 Kbps	9	40
*360 Kbyte	360 RPM	300 Kbps	9	40
1.2 Mbyte	360 RPM	500 Kbps	15	80

*360 Kbyte diskette in a 1.2 Mbyte drive.

The PC/AT disk controller provides direct control of the drive selects and motors via the Digital Output Register (DOR). As a result, drive selects on the 8272A are not utilized. This places drive selection and motor speed-up control responsibility with the software. The DOR is also used to perform a software reset of the disk controller and tristate the DRQ2 and IRQ6 output signals on the PC bus.

The design of the disk controller also requires address decode logic for the disk controller and register set, buffering for both the disk interface and PC bus, support for write precompensation and monitoring of the disk change signal via a separate read only register (DIR). An I/O address map of the complete register set for the PC/AT floppy disk controller is shown in Table 10-2.

Table 10-2. I/O Address Map for the PC/AT

I/O Address	Access Type	Description
3F0H	—	Unused
3F1H	—	Unused
3F2H	Write	Digital Output Register
3F3H	—	Unused
3F4H	Read	Main Status Register
3F5H	Read/Write	Data Register
3F6H	—	Unused
3F7H	Write	Data Rate Select Register
3F7H	Read	Digital Input Register

10.1.2 82077SL PC/AT SOLUTION

The 82077SL integrates the entire PC/AT controller design with the exception of the address decode on a single chip. The schematic for this solution is shown in Figure 10-2. The chip select for the 82077SL is generated by a 16L8 PAL that is programmed to decode addresses 03F0H thru 03F7H when AEN (Address Enable) is low. The programming equation for the PAL is shown in a ABEL file format in Figure 10-3. An alternative address decode solution could be provided by using a 74LS133 13 input NAND gate and 74LS04 inverter to decode A3–A14 and AEN. Although the PC/AT allows for a 64K I/O address space, decoding down to a 32K I/O address space is sufficient with the existing base of add-in cards.

A direct connection between the disk interface and the 82077SL is provided by on-chip output buffers with a 40 mA sink capability. Open collector outputs from the disk drive are terminated at the disk controller with a 150Ω resistor pack. The 82077SL disk interface inputs contain a schmitt trigger input structure for higher noise immunity. The host interface is a similar direct connection with 12 mA sink capabilities on DB0–DB7, INT and DRQ.

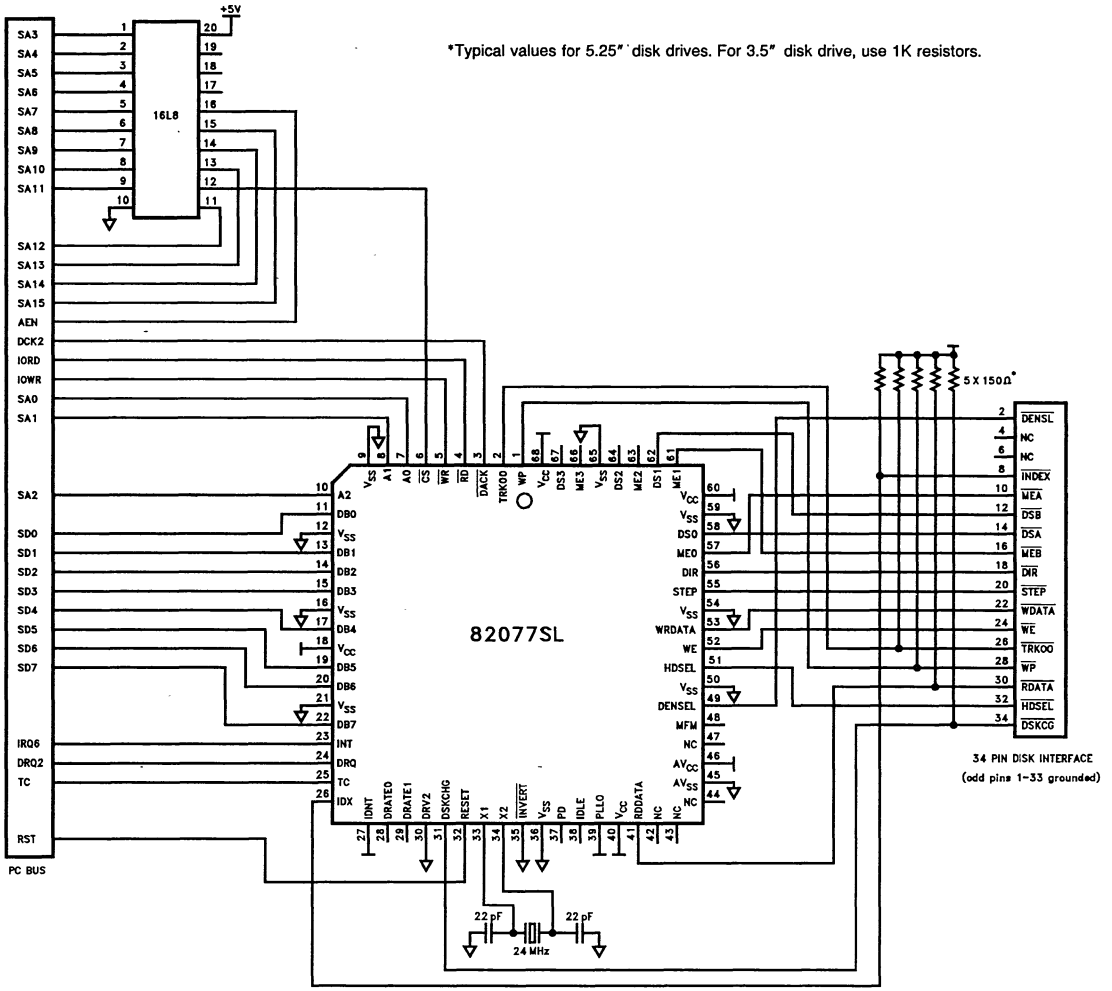


Figure 10-2. 82077SL PC/AT Floppy Disk Controller

```
MODULE PCAT077_LOGIC;

TITLE "82077SL PC/AT FLOPPY DISK CONTROLLER';
PCAT077 DEVICE 'P16L8';

GND,VCC                              PIN 10,20;
SA3,SA4,SA5,SA6,SA7,SA8,SA9,SA10     PIN 1,2,3,4,5,6,7,8;
SA11,SA12,SA13,SA14,SA15,AEN         PIN 9,11,13,14,15,16;
CS077_                               PIN 12;

EQUATIONS

"" CHIP SELECT FOR THE 82077SL (3F0H -- 3F7H)

CS077_ = !(!SA15 & !SA14 & !SA13 & !SA12 & !SA11 & !SA10 &
          SA9 & SA8 & SA7 & SA6 & SA5 & SA4 & !SA3 & !AEN);

END PCAT077_LOGIC
```

Figure 10-3. PAL Equation File for a PC/AT Compatible FDC Board

10.2 3.5″ Drive Interfacing

The 82077SL is designed to interface to both 3.5″ and 5.25″ disk drives. This is facilitated by the 82077SL by orienting IDENT to get the proper polarity of DENSEL for the disk drive being used. Typically DENSEL is active high for high (500 Kbps/ 1 Mbps) data rates on 5.25″ drives. And DENSEL is typically active low for high data rates on 3.5″ drives. A complete description of how to orient IDENT to get the proper polarity for DENSEL is given in Table 2-6.

10.2.1 3.5″ DRIVES UNDER THE AT MODE

When interfacing the 82077SL floppy disk controller with a 3.5″ disk drive in a PC/AT application, it is possible that two design changes will need to be implemented for the design discussed in Section 10.1. Most 3.5″ disk drives incorporate a totem pole interface structure as opposed to open collector. Outputs of the disk drive will drive both high or low voltage levels when the drive is selected, and float only when the drive has been deselected. These totem pole outputs generally can only sink or source 4 mA of current. As a result, it is recommended to replace the 150Ω termination resistor pack with a 4.7 KΩ package to pull floating signals inactive. Some other 3.5″ drives do have an open collector interface, but have limited sink capability. In these cases, the drive manufacturer manuals usually suggest a 1 KΩ termination.

A second possible change required under "AT mode" operation involves high capacity 3.5″ disk drives that utilize a density select signal to switch between media recorded at a 250 Kbps and 500 Kbps data rate. The polarity of this signal is typically inverted for 3.5″ drives versus 5.25″ drives. Thus, an inverter can be added between the DENSEL output of the 82077SL and the disk drive interface connector when using 3.5″ drives.

But drives that do not support both data rates or drives with an automatic density detection feature via an optical sensor do not require the use of the DENSEL signal.

Another method is to change the polarity of IDENT with a drive select signal. ORing RESET with the drive select signal (DS0–3) used for the 3.5″ disk drive will produce the proper polarity for DENSEL (assuming INVERT# is low).

10.2.2 3.5″ DRIVES UNDER THE PS/2 MODES

If IDENT is strapped to ground, the DENSEL output signal polarity will reflect a typical 3.5″ drive mode of operation. That is, DENSEL will be high for 250 Kbps or 300 Kbps and low for 500 Kbps or 1 Mbps (assuming INVERT# is low). Thus the only change from the disk interface shown in Figure 10-2 is to replace the 150Ω termination resistor pack with a value of about 10 KΩ. This will prevent excessive current consumption on the CMOS inputs of the 82077SL by pulling them inactive when the drive(s) are deselected.

10.2.3 COMBINING 5.25″ AND 3.5″ DRIVES

If 5.25″ and 3.5″ drives are to be combined in a design, then steps need to be taken to avoid contention problems on the disk interface. Since 3.5″ drives do not have a large sink capability, the 150Ω termination resistor pack required by 5.25″ drives cannot be used with the 3.5″ drive. To accommodate both drives with the same disk controller, the outputs of the 3.5″ drive should be buffered before connecting to the 82077SL disk interface inputs. The 82077SL inputs are then connected to the necessary resistive termination load for the 5.25″ interface.

The block diagram in Figure 10-4 highlights how a combined interface could be designed. In this example, the 5.25″ drive is connected to drive select 0 (DS0) and the 3.5″ drive is connected to drive select 1 (DS1). DS1 is also used to enable a 74LS244 buffer on the output signals of the 3.5″ drive. The drive select logic of the 82077SL is mutually exclusive and prevents the activation of the buffer and 5.25″ drive at the same time. Since the 74LS244 has an I_{OL} of 24 mA, the termination resistor should be increased to 220Ω. This could impact the reliability of the 5.25″ drive interface if the cable lengths are greater than 5 feet.

To accommodate the polarity reversal of the DENSEL signal for 3.5″ drives, it is routed through an inverter for the 3.5″ drive interface. A 1 KΩ pull-up should be placed on the output of the inverter to satisfy the I_{OH} requirements for the 3.5″ drive when using a 74LS04.

10.2.4 OPTIMIZING 82077SL-1 FOR TAPE DRIVE MODE

The floppy disk controller can be configured for the tape drive mode by both hardware and software. Configuring the 82077SL-1 for the tape drive mode refers to optimization of the internal data separator in order to deal with the effect of ISV which is more pronounced on a tape drive than on a floppy disk controller. Hardware selection is done by setting the PLL0 (pin 39) to 0 or GND. This optimizes the data separator for tape drives by changing the loop filter component values and loop gain. TDR selection is disabled under this mode. Software selection of the tape drive mode for the FDC is implemented via setting of the appropriate bits in the tape drive register (TDR). This selection is enabled only while PLL0 is set high. This aids the user in configuring the particular drives as tape drive even when in floppy mode.

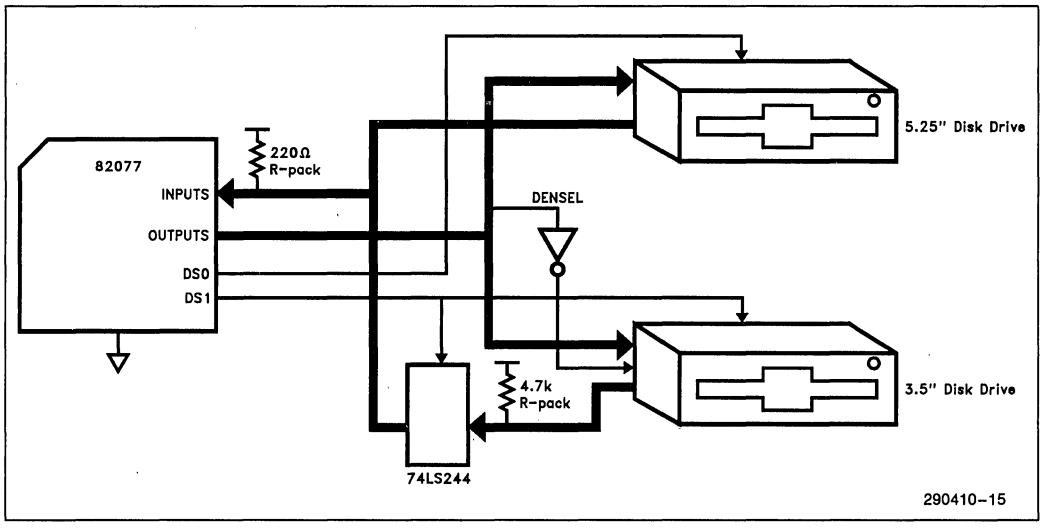


Figure 10-4. Combined 3.5″ and 5.25″ Drive Interface

As shown below the TDR contains two bits which can be utilized to assign tape support to a particular drive during initialization.

7	6	5	4	3	2	1	0
*	*	*	*	*	*	TAPE SEL1	TAPE SEL1

Hardware resets clears this register but it remains unaffected by any software reset. TDR[2:7] remain in a tristated condition and are not readable. Drive 0 is reserved for the floppy boot drive and cannot be configured for tape drives using the TDR (software mechanism). Hardware selection overrides any selection made by the software, i.e., by setting PLL0 to GND, tape drive mode will be selected regardless of the changes made to the TDR. Although the software mechanism does not allow to select drive 0 for tape drive, when PLL0 = 0 any drive can be supported for tape drive.

82077SL-1 has the capability to support up to a total of four drives. Most PC systems today have at least one floppy disk drive. This leaves the possibility of installing up to three tape drives. The following describes a way to configure the floppy disk controller in a multiple tape drive environment. This also depends on whether the system manufacturer wishes to leave certain drive slots fixed for tape drives or variable by the user.

All Tape Drives Are Variable—If the drives chosen as tape drives are variable then the configuration mechanism used is strictly software. After strapping PLL0 high, the bits TDR[0:1] can be programmed during initialization for various drives that can be selected as tape drives. It should be noted that in this case drive 0 cannot be selected as one of the tape drives.

Combination of Fixed/Variable Tape Drives—If any drive can be determined to be fixed then either the motor enable pin or the drive select pin of that particular drive can be used to drive PLL0 to GND when selected. Figure 10-5A and Figure 10-5B show two scenarios where drives that are fixed for tape drive use their motor enable or drive select signals to drive PLL0 to GND.

Figure 10-5C shows by using jumpers flexibility can be incorporated in the system and the drive/s to be fixed for tape drives can be left to the user.

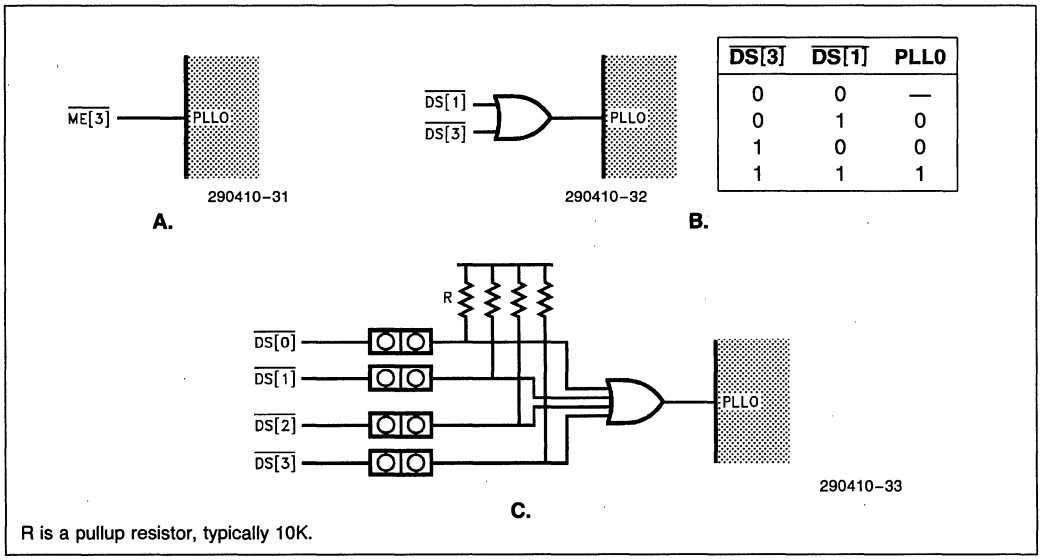


Figure 10-5. Optimizing 82077SL-1 for Tape Drive Mode

11.0 D.C. SPECIFICATIONS

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

11.1 Absolute Maximum Ratings

Storage Temperature .....−65°C to +150°C
Supply Voltage .....−0.5 to +8.0V
Voltage on Any Input .....GND − 2V to 6.5V
Voltage on Any Output..GND − 0.5V to VCC+0.5V
Power Dissipation .....1 Watt

**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.*

11.2 D.C. Characteristics

T_A = 0°C to = 70°C, V_{CC} = +5V ±10%, V_{SS} = AV_{SS} = 0V

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{ILC}	Input Low Voltage, X1	−0.5	0.8	V	
V_{IHC}	Input High Voltage, X1	3.9	V_{CC} +0.5	V	
V_{IL}	Input Low Voltage (all pins except X1)	−0.5	0.8	V	
V_{IH}	Input High Voltage (all pins except X1)	2.0	V_{CC} +0.5	V	
V_{OL}	Output Low Voltage MFM		0.4	V	I_{OL} = 2.5 mA
	DRATE0–1		0.4	V	I_{OL} = 6.0 mA
	DB0–7, INT and DRQ		0.4	V	I_{OL} = 12 mA
	ME0–3, DS0–3, DIR, STP WRDATA, WE, HDSEL and DENSEL		0.4	V	I_{OL} = 40 mA
V_{OH}	Output High Voltage MFM	3.0		V	I_{OH} = −2.5 mA
	All Other Outputs	3.0		V	I_{OH} = −4.0 mA
	All Outputs	V_{CC} − 0.4		V	I_{OH} = −100 µA
	V_{CC} Supply Current (Total)				
I_{CC1}	1 Mbps Data Rate, V_{IL} = V_{SS}, V_{IH} = V_{CC}		45	mA	(Notes 1, 2)
I_{CC2}	1 Mbps Data Rate, V_{IL} = 0.45, V_{IH} = 2.4		50	mA	(Notes 1, 2)
I_{CC3}	500 Kbps Data Rate, V_{IL} = V_{SS}, V_{IH} = V_{CC}		35	mA	(Notes 1, 2)
I_{CC4}	500 Kbps Data Rate, V_{IL} = 0.45, V_{IH} = 2.4		40	mA	(Notes 1, 2)
I_{CCSB}	I_{CC} in Powerdown		60	µA	(Note 3)
I_{IL}	Input Load Current (all input pins)		10 −10	µA µA	V_{IN} = V_{CC} V_{IN} = 0V
I_{OFL}	Data Bus Output Float Leakage		±10	µA	0.45 < V_{OUT} < V_{CC}

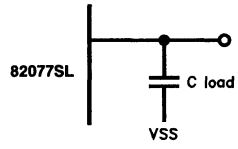
NOTES:
1. The data bus are the only inputs that may be floated.
2. Tested while reading a sync field of "00". Outputs not connected to D.C. Loads.
3. V_{IL} = V_{SS}, V_{IH} = V_{CC}; Outputs not connected to D.C. loads.

Capacitance

C_{IN}	Input Capacitance	10	pF	F = 1 MHz, T_A = 25°C
C_{IN1}	Clock Input Capacitance	20	pF	Sampled, not 100% Tested
$C_{I/O}$	Input/Output Capacitance	20	pF	

NOTE:
All pins except pins under test are tied to AC ground.

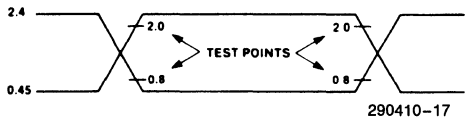
LOAD CIRCUIT



290410-16

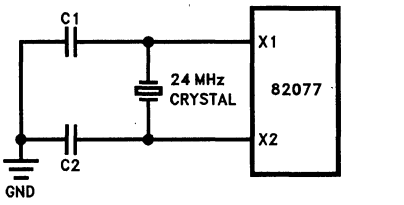
C_{load} = 50 pF for all logic outputs, 100 pF for the data bus.

A. C. TESTING INPUT, OUTPUT WAVEFORM



290410-17

11.3 Oscillator



290410-3

Figure 11-2. Crystal Oscillator Circuit

The 24 MHz clock can be supplied either by a crystal or a MOS level square wave. All internal timings are referenced to this clock or a scaled count which is data rate dependent.

The crystal oscillator must be allowed to run for 10 ms after VCC has reached 4.5V or exiting the POWERDOWN mode to guarantee that it is stable.

Crystal Specifications

Frequency:	24 MHz ±0.1%
Mode:	Parallel Resonant Fundamental Mode
Series Resistance:	Less than 40Ω
Shunt Capacitance:	Less than 5 pF

12.0 A.C. SPECIFICATIONS

T_A = 0°C to 70°C, V_{CC} = +5V ±10%, V_{SS} = AV_{SS} = 0V

Symbol	Parameter	Min	Max	Unit
CLOCK TIMINGS				
t1	Clock Rise Time		10	ns
	Clock Fall Time		10	ns
t2	Clock High Time(7)	16	26	ns
t3	Clock Low Time(7)	16	26	ns
t4	Clock Period	41.66	41.66	ns
t5	Internal Clock Period(3)			
HOST READ CYCLES				
t7	Address Setup to $\overline{RD}$	5		ns
t8	$\overline{RD}$ Pulse Width	90		ns
t9	Address Hold from RD	0		ns
t10	Data Valid from $\overline{RD}$(12)		80	ns
t11	Command Inactive	60		ns
t12	Output Float Delay		35	ns
t13	INT Delay from RD(16)		t5 + 125	ns
t14	Data Hold from $\overline{RD}$	5		ns
HOST WRITE CYCLES				
t15	Address Setup to $\overline{WR}$	5		ns
t16	$\overline{WR}$ Pulse Width	90		ns
t17	Address Hold from WR	0		ns
t18	Command Inactive	60		ns
t19	Data Setup to WR	70		ns
t20	Data Hold from WR	0		ns
t21	INT Delay from WR(16)		t5 + 125	ns
DMA CYCLES				
t22	DRQ Cycle Period(1)	6.5		µs
t23	DACK to DRQ Inactive		75	ns
t23a	DRQ to DACK Inactive	(Note 15)		ns
t24	RD to DRQ Inactive(4)		100	ns
t25	DACK Setup to $\overline{RD}$, $\overline{WR}$	5		ns
t26	DACK Hold from RD, WR	0		ns
t27	DRQ to $\overline{RD}$, $\overline{WR}$ Active(1)	0	6	µs
t28	Terminal Count Width(10)	50		ns
t29	TC to DRQ Inactive		150	ns
RESET				
t30	"Hardware" Reset Width(5)	170		t4
t30a	"Software" Reset Width(5)	(Note 11)		ns
t31	Reset to Control Inactive		2	µs

A.C. SPECIFICATIONS (Continued)

T_A = 0°C to 70°C, V_{CC} = +5V ±10%, V_{SS} = AV_{SS} = 0V

Symbol	Parameter	Min	Max	Unit
WRITE DATA TIMING				
t32	Write Data Width(6)			ns
DRIVE CONTROL				
t35	DIR Setup to STEP(14)	1.0		µs
t36	DIR Hold from STEP	10		µs
t37	STEP Active Time (High)	2.5		µs
t38	STEP Cycle Time(2)			µs
t39	INDEX Pulse Width	5		t5
t41	WE to HDSEL Change	(Note 13)		ms
READ DATA TIMING				
t40	Read Data Pulse Width	50		ns
f44	PLL Data Rate			
	82077SL-1		1M	bits/sec
	82077SL		1M	bits/sec
	82077SL-5		500K	bits/sec
t44	Data Rate Period = 1/f44			
tLOCK	Lockup Time		64	t44

NOTES:

1. This timing is for FIFO threshold = 1. When FIFO threshold is N bytes, the value should be multiplied by N and subtract 1.5 µs. The value shown is for 1 Mbps, scales linearly with data rate.
2. This value can range from 0.5 ms to 8.0 ms and is dependent upon data rate and the Specify command value.
3. Many timings are a function of the selected data rate. The nominal values for the internal clock period (t5) for the various data rates are:

1 Mbps	3 x oscillator period = 125 ns
500 Kbps	6 x oscillator period = 250 ns
300 Kbps	10 x oscillator period = 420 ns
250 Kbps	12 x oscillator period = 500 ns

4. If $\overline{DACK}$ transitions before $\overline{RD}$, then this specification is ignored. If there is no transition on $\overline{DACK}$, then this becomes the DRQ inactive delay.
5. Reset requires a stable oscillator to meet the minimum active period.
6. Based on the internal clock period (t5). For various data rates, the Write Data Width minimum values are:

1 Mbps	5 x oscillator period −50 ns = 150 ns
500 Kbps	10 x oscillator period −50 ns = 360 ns
300 Kbps	16 x oscillator period −50 ns = 615 ns
250 Kbps	19 x oscillator period −50 ns = 740 ns

7. Test points for clock high time are 3.5V. Due to transitional times, clock high time max and clock low time max cannot be met simultaneously. Clock high time min and clock low time max cannot be met simultaneously.
8. Based on internal clock period (t5).
9. Jitter tolerance is defined as: $\frac{\text{Maximum bit shift from nominal position}}{\text{1/4 period of nominal data rate}} \times 100\%$

It is a measure of the allowable bit jitter that may be present and still be correctly detected. The data separator jitter tolerance is measured under dynamic conditions that jitters the bit stream according to a reverse precompensation algorithm.

10. TC width is defined as the time that both TC and DACK are active.

A.C. SPECIFICATIONS (Continued)

NOTES: (Continued)

11. The minimum reset active period for a software reset is dependent on the data rate, after the 82077SL has been properly reset using the t30 spec. The minimum software reset period then becomes:

1 Mbps	3 x t4 = 125 ns
500 Kbps	6 x t4 = 250 ns
300 Kbps	10 x t4 = 420 ns
250 Kbps	12 x t4 = 500 ns

12. Status Register's status bits which are not latched may be updated during a Host read operation.
13. The minimum MFM values for WE to HDSEL change (t41) for the various data rates are:

1 Mbps	0.5 ms + [8 x GPL]
500 Kbps	1.0 ms + [16 x GPL]
300 Kbps	1.6 ms + [26.66 x GPL]
250 Kbps	2.0 ms + [32 x GPL]

GPL is the size of gap 3 defined in the sixth byte of a Write Command.

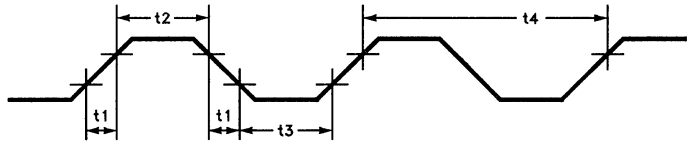
14. This timing is a function of the selected data rate as follows:

1 Mbps	1.0 μs Min
500 Kbps	2.0 μs Min
300 Kbps	3.3 μs Min
250 Kbps	4.0 μs Min

15. This timing is a function of the internal clock period (t5) and is given as (2/3) t5. The values of t5 are shown in Note 3.
16. The timings t13 and t21 are specified for INT signal in the polling mode only. These timings in case of the result phase of the read and write commands are microcode dependent.
17.

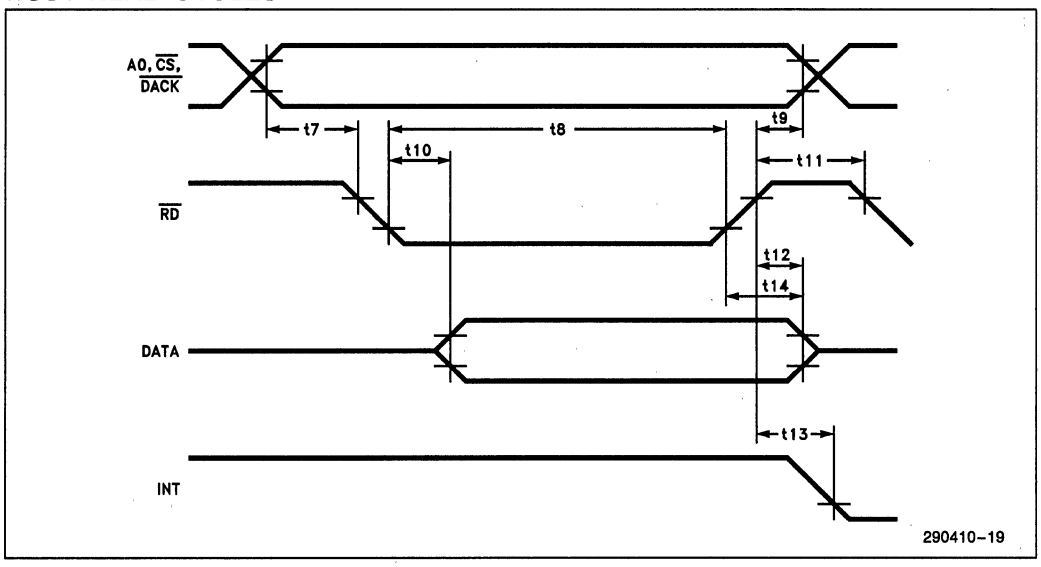
Part Specification	Supported Feature	
	Tape Drive Mode	Perpendicular Mode
82077SL-1	Yes	Yes
82077SL		Yes
82077SL-5		

CLOCK TIMINGS

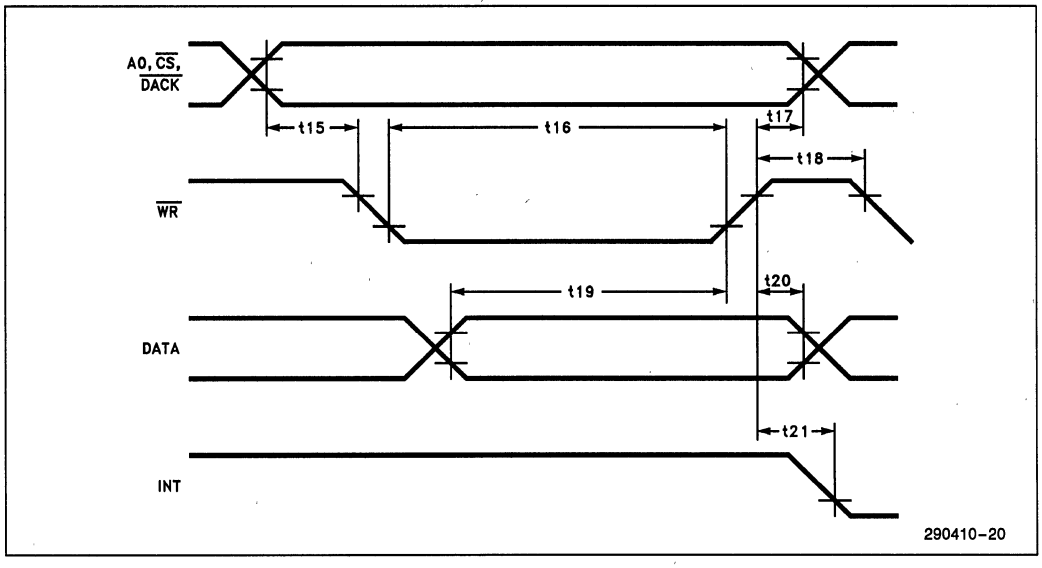


290410-18

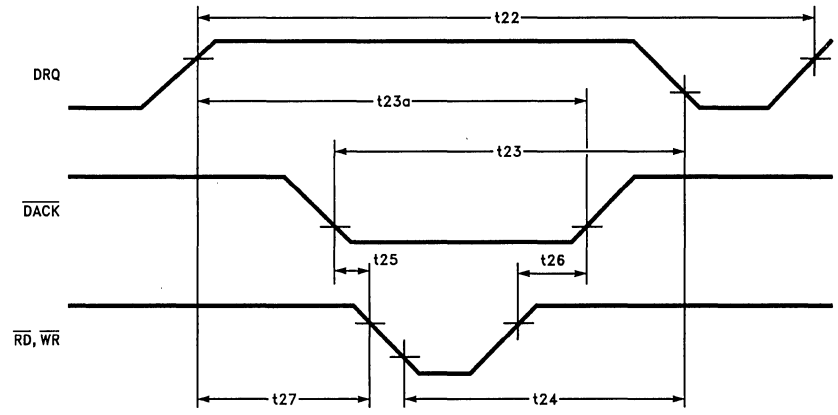
HOST READ CYCLES



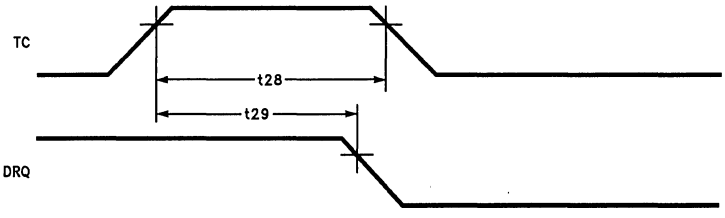
HOST WRITE CYCLES



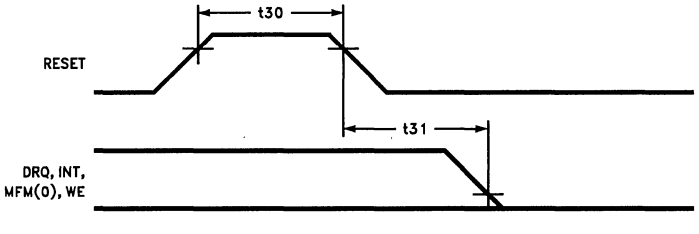
DMA CYCLES



TERMINAL COUNT

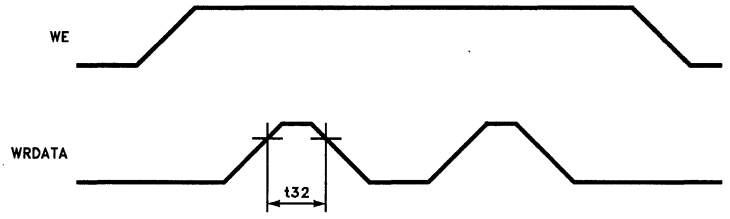


RESET



NOTE:
MFM(0) refers to the MFM pin left open during hardware reset.

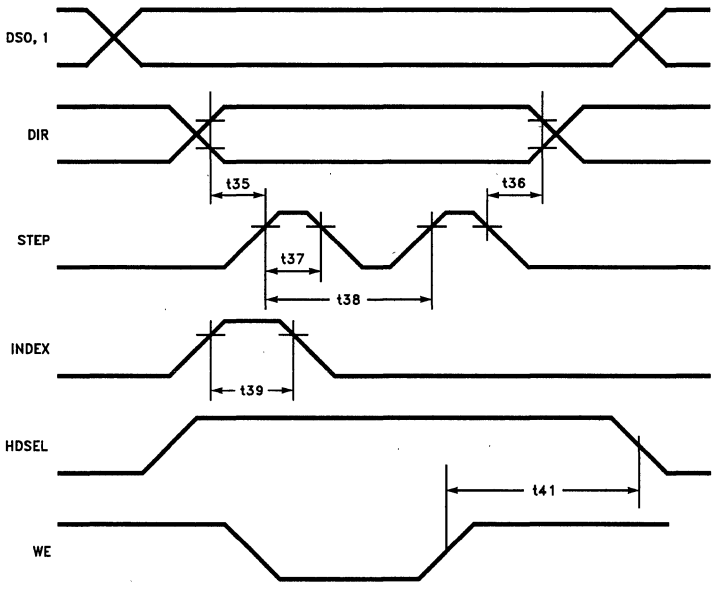
WRITE DATA TIMING



290410-24

NOTE:
Invert high.

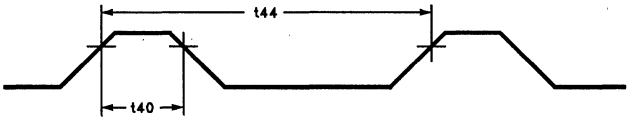
DRIVE CONTROL



290410-25

NOTE:
For overlapped seeks, only one step pulse per drive selection is issued. Non-overlapped seeks will issue all programmed step pulses. Invert high.

INTERNAL PLL



290410-26

NOTE:
Invert high.

13.0 DATA SEPARATOR CHARACTERISTICS FOR FLOPPY DISK MODE

Figure 13-1. Typical Jitter Tolerance vs Data Rate (Capture Range) (250 Kbps)

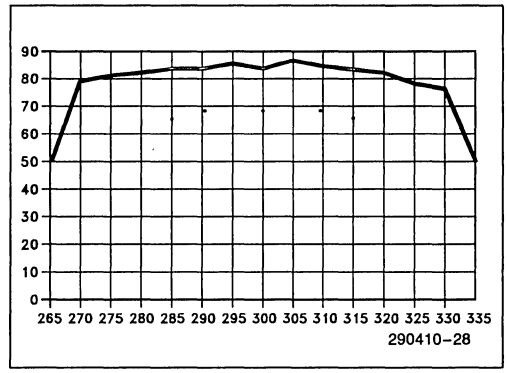


Figure 13-2. Typical Jitter Tolerance vs Data Rate (Capture Range) (300 Kbps)

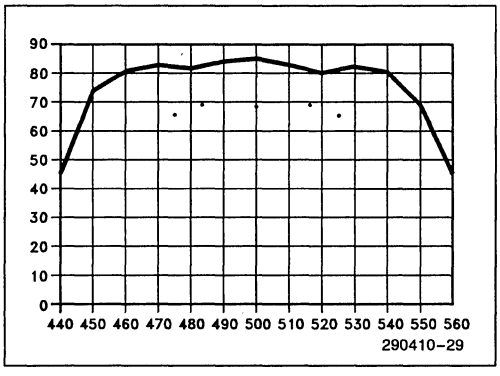


Figure 13-3. Typical Jitter Tolerance vs Data Rate (Capture Range) (500 Kbps)

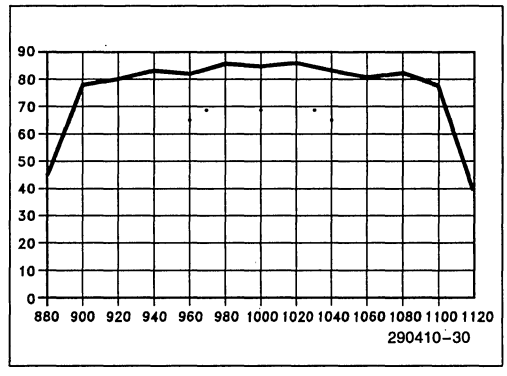


Figure 13-4. Typical Jitter Tolerance vs Data Rate (Capture Range) (1 Mbps), 82077SL-1

Jitter Tolerance measured in percent. See datasheet — Section 3.2.1 capture range expressed as a percent of data rate, i.e., ±3%.

- = Test Points:
  250, 300, 500 Kbps are center, ±3% @ 68% jitter, ±5% @ 65% jitter
  1 Mbps are center, ±3% @ 68% jitter, ±4% @ 63% jitter

Test points are tested at temperature and V_{CC} limits. Refer to the datasheet. Typical conditions are: room temperature, nominal V_{CC}.

14.0 DATA SEPARATOR CHARACTERISTICS FOR TAPE DRIVE MODE

Figure 14-1. Typical Jitter Tolerance vs Data Rate (Capture Range) (±0% ISV, 500 Kbps)

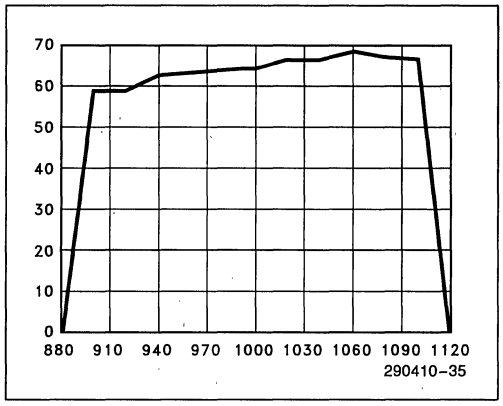


Figure 14-2. Typical Jitter Tolerance vs Data Rate (Capture Range) (±0% ISV, 1 Mbps)

Figure 14-3. Typical Jitter Tolerance vs Data Rate (Capture Range) (±3% ISV, 500 Kbps)

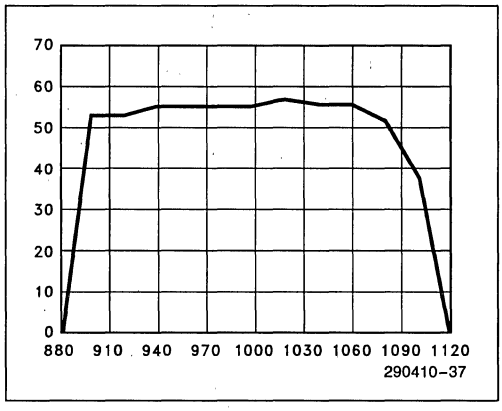


Figure 14-4. Typical Jitter Tolerance vs Data Rate (Capture Range) (±3% ISV, 1 Mbps)

NOTES:
1. Jitter Tolerance measured in percent. See datasheet — Section 3.2.1 capture range expressed as a percent of data rate, i.e., ±5%.
2. Typical conditions are: room temperature, nominal V_{CC}.

15.0 82077SL 68-LEAD PLCC PACKAGE THERMAL CHARACTERISTICS

T_A Ambient Temp. (°C)	Typical Values				θ_{ja} (°C/W)	θ_{jc} (°C/W)
	T_c (°C)	T_j (°C)	I_{cc} (mA)	V_{cc} (V)		
70	75	75	30	5.0	36	5

NOTES:
Case Temperature Formula:
$T_c = T_a + P [\theta_{ja} - \theta_{jc}]$
Junction Temperature Formula:
$T_j = T_c + p [\theta_{jc}]$
P = Power dissipated
θ_{jc} = thermal resistance from the junction to the case.
θ_{ja} = thermal resistance from the junction to the ambient.

82077SL Revision Summary

2

The following changes have been made since revision 004:

All references to the FM Mode have been removed. The 82077SL does not support the FM Mode.

The following changes have been made since revision 003:

1. The 82077SL does not support the FM Mode. All references to the FM Mode have been removed. The 82077SL does not support the FM Mode.

The following changes have been made since revision 002:

Title Page — Second paragraph, last two sentences deleted and replaced with:

The 82077SL is available in three versions—82077SL-5, 82077SL and 82077SL-1. 82077SL-1 has all features listed in this data sheet. It supports both tape drives and 4 MB floppy drives. The 82077SL supports 4 MB floppy drives and is capable of operation at all data rates through 1 Mbps. The 82077SL-5 supports 500/300/250 Kbps data rates for high and low density floppy drives.

Section 2.1.8a — Bit 7 has been changed from DSK to DSKCHG.

Section 2.3 — New sentence added to end of paragraph. This sentence reads, "$\overline{CS}$ can be held inactive during DMA transfers".

Section 6.0 — Addition of Scan Equal, Scan Low or Equal, and Scan High or Equal to Table 6-1.

Section 6.1.8 — New section added, titled "Scan Commands".

Section 12.0 — Timing diagram, DMA Cycles corrected. Symbol t26 corrected on signal $\overline{DACK}$.

82078 CHMOS SINGLE-CHIP FLOPPY DISK CONTROLLER

- **Small Footprint and Low Height Packages**
- **Supports Standard 5.0V as Well as Low Voltage 3.3V Platforms**
  - **— Selectable 3.3V and 5.0V Configuration**
  - **— 5.0V Tolerant Drive Interface**
- **Enhanced Power Management**
  - **— Application Software Transparency**
  - **— Programmable Powerdown Command**
  - **— Save and Restore Commands for 0V Powerdown**
  - **— Auto Powerdown and Wakeup Modes**
  - **— Two External Power Management Pins**
  - **— Consumes No Power While in Powerdown**
- **Programmable Internal Oscillator**
- **Floppy Drive Support Features**
  - **— Drive Specification Command**
  - **— Media ID Capability Provides Media Recognition**
  - **— Drive ID Capability Allows the User to Recognize the Type of Drive**
  - **— Selectable Boot Drive**
  - **— Standard IBM and ISO Format Features**
  - **— Format with Write Command for High Performance in Mass Floppy Duplication**
- **Integrated Host/Disk Interface Drivers**
- **Integrated Analog Data Separator**
  - **— 250 Kbits/sec**
  - **— 300 Kbits/ sec**
  - **— 500 Kbits/ sec**
  - **— 1 Mbits/sec**
  - **— 2 Mbits/sec**
- **Integrated Tape Drive Support**
  - **— Standard 1 Mbps/500 Kbps/ 250 Kbps Tape Drives**
  - **— New 2 Mbps Tape Drive Mode**
- **Perpendicular Recording Support for 4 MB Drives**
- **Fully Decoded Drive Select and Motor Signals**
- **Programmable Write Precompensation Delays**
- **Addresses 256 Tracks Directly, Supports Unlimited Tracks**
- **16 Byte FIFO**
- **Single-Chip Floppy Disk Controller Solution for Portables and Desktops**
  - **— 100% PC-AT* Compatible**
  - **— 100% PS/2* Compatible**
  - **— 100% PS/2 Model 30 Compatible**
  - **— Fully Compatible with Intel's 386SL Microprocessor SuperSet**
  - **— Integrated Drive and Data Bus Buffers**
- **Available in 64 Pin QFP and 44 Pin QFP Package**
  (See Package Specification Order Number 240800, Package Type S)

The 82078 Product Family brings a set of enhanced floppy disk controllers. These include several features that allow for easy implementation in both the portable and desktop market. The current family includes a 64 pin and a 44 pin part in the smaller form factor QFP package. The 3.3V version of the 64 pin part provides an ideal solution for the rapidly emerging 3.3V platforms. It also allows for a 5.0V tolerant floppy drive interface that lets the users retain their normal 5.0V drives. Another version of the 64 pin part provides support for 2 Mbps data rate tape drives.

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

The complete document for this product is available from Intel's Literature Center at 1-800-548-4725.

February 1994
Order Number: 290468-003

Table 1-0. 64 Pin Part Versions

	3.3V	5.0V	2 Mbps Data Rate
82078SL	X	X	
82078-1		X	X

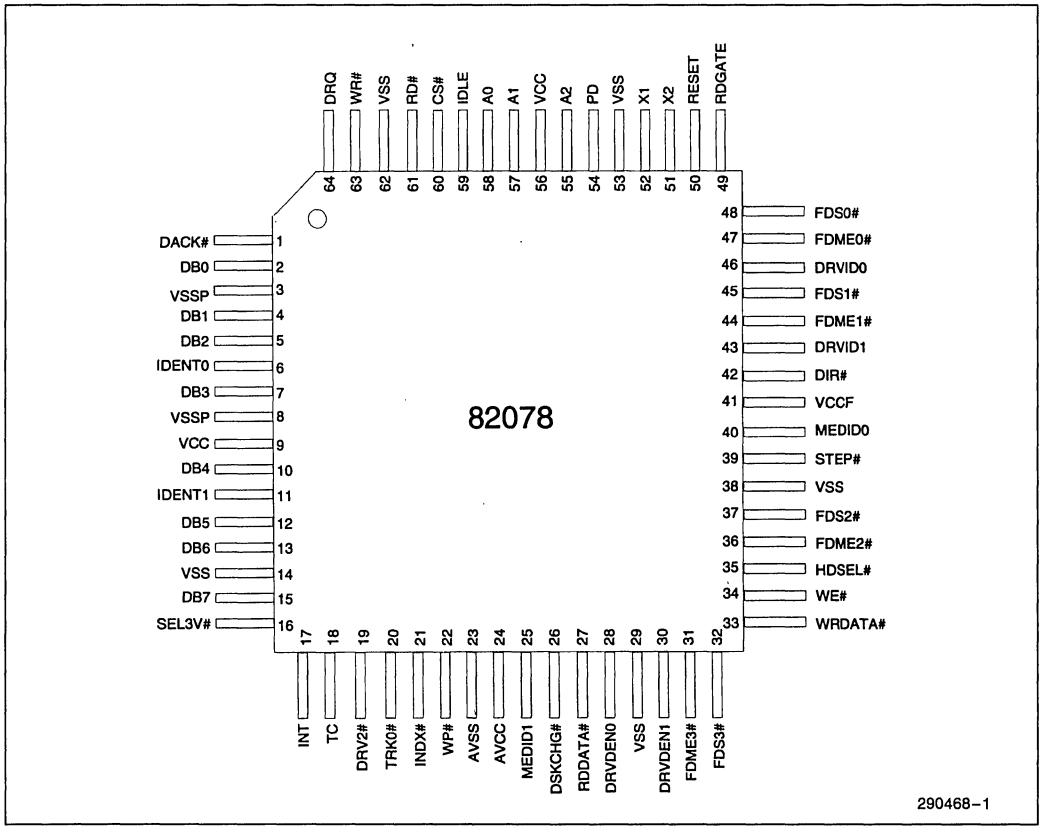
The 44 pin is targeted for platforms that are operated at 3.3V or 5.0V and do not require more than two drive support. The 82078-5 is designed for price sensitive 5.0V designs which do not include 4 MB drive support.

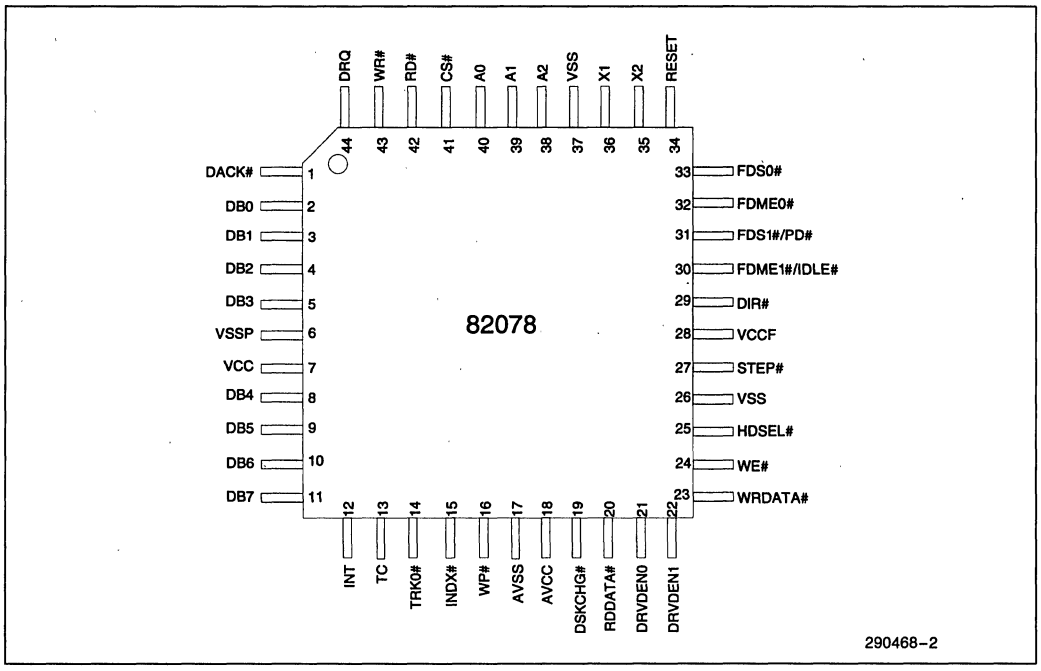
Table 2-0. 44 Pin Part Versions

	3.3V	5.0V	1 Mbps Data Rate
82078		X	X
82078-5		X	
82078-3	X		X

Both parts can be operated at 1 Mbps/500 Kbps/300 Kbps/250 Kbps. Additionally, one version of the 64 pin part provides 2 Mbps data rate operation specific for the new tape drives.

The 82078 is fabricated with Intel's advanced CHMOS III technology.




82078
DRQ 44
WR# 43
RD# 42
CS# 41
A0 40
A1 39
A2 38
VSS 37
X1 36
X2 35
RESET 34
DACK# 1
DB0 2
DB1 3
DB2 4
DB3 5
VSSP 6
VCC 7
DB4 8
DB5 9
DB6 10
DB7 11
33 FDS0#
32 FDME0#
31 FDS1#/PD#
30 FDME1#/IDLE#
29 DIR#
28 VCCF
27 STEP#
26 VSS
25 HDSEL#
24 WE#
23 WRDATA#
12 INT
13 TC
14 TRK0#
15 INDX#
16 WP#
17 AVSS
18 AVCC
19 DSKCHG#
20 RDDATA#
21 DRVDEN0
22 DRVDEN1
290468-2

82078 44 PIN
CHMOS SINGLE-CHIP FLOPPY DISK CONTROLLER

- Small Footprint and Low Height Package
- Enhanced Power Management
  - Application Software Transparency
  - Programmable Powerdown Command
  - Save and Restore Commands for Zero-Volt Powerdown
  - Auto Powerdown and Wakeup Modes
  - Two External Power Management Pins
  - Consumes No Power While in Powerdown
- Integrated Analog Data Separator
  - 250 Kbps
  - 300 Kbps
  - 500 Kbps
  - 1 Mbps
- Programmable Internal Oscillator
- Floppy Drive Support Features
  - Drive Specification Command
  - Selectable Boot Drive
  - Standard IBM and ISO Format Features
  - Format with Write Command for High Performance in Mass Floppy Duplication
- Integrated Tape Drive Support
  - Standard 1 Mbps/500 Kbps/ 250 Kbps Tape Drives
- Perpendicular Recording Support for 4 MB Drives
- Integrated Host/Disk Interface Drivers
- Fully Decoded Drive Select and Motor Signals
- Programmable Write Precompensation Delays
- Addresses 256 Tracks Directly, Supports Unlimited Tracks
- 16 Byte FIFO
- Single-Chip Floppy Disk Controller Solution for Portables and Desktops
  - 100% PC/AT* Compatible
  - Fully Compatible with Intel386™ SL
  - Integrated Drive and Data Bus Buffers
- Separate 5.0V and 3.3V Versions of the 44 Pin part are Available
- Available in a 44 Pin QFP Package

2

The 82078, a 24 MHz crystal, a resistor package, and a device chip select implements a complete solution. All programmable options default to 82078 compatible values. The dual PLL data separator has better performance than most board level/discrete PLL implementations. The FIFO allows better system performance in multi-master (e.g., Microchannel, EISA).

The 82078 maintains complete software compatibility with the 82077SL/82077AA/8272A floppy disk controllers. It contains programmable power management features while integrating all of the logic required for floppy disk control. The power management features are transparent to any application software.

The 82078 is fabricated with Intel's advanced CHMOS III technology and is also available in a 64-lead QFP package.

---

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

October 1995
Order Number: 290474-003

82078 44 Pin CHMOS Single-Chip Floppy Disk Controller

CONTENTS

CONTENTS

CONTENTS

PAGE

CONTENTS

PAGE

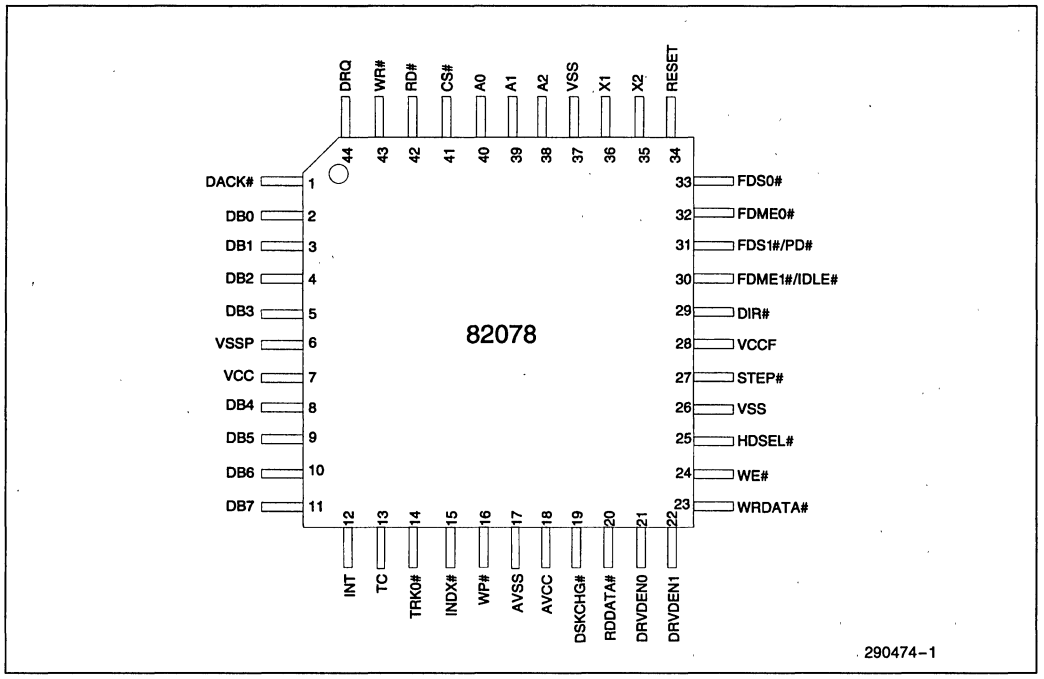


Figure 1-0. 82078 44 Pin Pinout

Table 1.0. 82078 (44 Pin) Description

Symbol	Pin #	I/O	@ H/W Reset	Description
HOST INTERFACE				
RESET	34	I	N/A	**RESET:** A high level places the 82078 in a known idle state. All registers are cleared except those set by the Specify command.
A0 A1 A2	40 39 38	I	N/A	**ADDRESS:** Selects one of the host interface registers: **A2 A1 A0 Access Register** 0 0 0 R Reserved 0 0 1 R/W Status Register B SRB 0 1 0 R/W Digital Output Register DOR 0 1 1 R/W Tape Drive Register TDR 1 0 0 R Main Status Register MSR 1 0 0 W Data Rate Select Register DSR 1 0 1 R/W Data Register (FIFO) FIFO 1 1 0 Reserved 1 1 1 R Digital Input Register DIR 1 1 1 W Configuration Control Register CCR
CS#	41	I	N/A	**CHIP SELECT:** Decodes the base address range and qualifies RD# and WR#.
RD#	42	I	N/A	**READ:** Read control signal for data transfers from the floppy drive to the system.

Table 1.0 82078 (44 Pin) Description (Continued)

Symbol	Pin #	I/O	@ H/W Reset	Description
HOST INTERFACE (Continued)				
WR#	43	I	N/A	**WRITE:** Write control signal for data transfers to the floppy drive from the system.
DRQ	44	O		**DMA REQUEST:** Requests service from a DMA controller. Normally active high, but will go to high impedance in AT and Model 30 modes when the appropriate bit is set in the DOR.
DACK#	1	I	N/A	**DMA ACKNOWLEDGE:** Control input that qualifies the RD#, WR# inputs in DMA cycles. Normally active low, but is disabled in AT and Model 30 modes when the appropriate bit is set in the DOR.
DB0 DB1 DB2 DB3 DB4 DB5 DB6 DB7	2 3 4 5 8 9 10 11	I/O		**DATA BUS:** 12 mA data bus.
INT	12	O		**INTERRUPT:** Signals a data transfer in non-DMA mode and when status is valid. Normally active high, but goes to high impedance when the appropriate bit is set in the DOR.
TC	13	I	N/A	**TERMINAL COUNT:** Control line from a DMA controller that terminates the current disk transfer. TC is effective only when qualified by DACK#. This input is active high.
X1 X2	36 35		N/A	**EXTERNAL CLOCK OR CRYSTAL:** Connection for a 24 MHz fundamental mode parallel resonant crystal. X1 can also be driven by an external clock (external oscillator) which can be either at 48 MHz or 24 MHz. If external oscillator is used then the PDOSC bit can be set to turn off the internal oscillator. Also, if a 48 MHz external oscillator is used then the CLK48 bit must be set in the enhanced CONFIGURE command.
PLL SECTION				
RDDATA#	20	I	N/A	**READ DATA:** Serial data from the floppy disk.
DISK CONTROL				
TRK0#	14	I	N/A	**TRACK0:** This is an active low signal that indicates that the head on track 0.
INDX#	15	I	N/A	**INDEX:** This is an active low signal that indicates the beginning of the track.
WP#	16	I	N/A	**WRITE PROTECT:** This is an active low signal that indicates whether the floppy disk in the drive is write protected.
DSKCHG#	19	I	N/A	**DISK CHANGE:** This is an input from the floppy drive reflected in the DIR.
DRVDEN0, DRVDEN1	21 22	O		**DRIVE DENSITY:** These signals are used by the floppy drive to configure the drive for the appropriate media.
WRDATA#	23	O		**WRITE DATA:** MFM serial data to the drive. Precompensation value is selectable through software.

Table 1.0 82078 (44 Pin) Description (Continued)

Symbol	Pin #	I/O	@ H/W Reset	Description
DISK CONTROL (Continued)				
WE#	24	O		**WRITE ENABLE:** Floppy drive control signal that enables the head to write onto the floppy disk.
STEP#	27	O		**STEP:** Supplies step pulses to the floppy drive to move the head between tracks.
DIR#	29	O		**DIRECTION:** It is an active low signal which controls the direction the head moves when a step signal is present. The head moves inwards towards the center if this signal is active.
HDSEL#	25	O		**HEAD SELECT:** Selects which side of the floppy disk is to be used for the corresponding data transfer. It is active low and an active level selects head 1, otherwise it defaults to head 0.
FDME0#	32	O		**FLOPPY DRIVE MOTOR ENABLE 0:** Decoded motor enable for drive 0. The motor enable pins are directly controlled via the DOR and are a function of the mapping based on BOOTSEL bits in the TDR.
FDME1#/IDLE#	30	O		**FLOPPY DRIVE MOTOR ENABLE** or **IDLE:** One of these is selected based on the level of the 44PDEN bit in the auto powerdown command. **FLOPPY DRIVE MOTOR ENABLE 1:** Decoded motor enable for drive 1. The motor enable pins are directly controlled via the DOR and are a function of the mapping based on BOOTSEL bits in the TDR. **IDLE:** This pin indicates that the part is in the IDLE state and can be powered down. IDLE state is defined as MSR = 80H, INT = 0, and the head being "unloaded" (as defined in the section describing powerdown). Whenever the part is in this state, IDLE pin is active low. If the part is powered down by the Auto Powerdown Mode, IDLE pin is set low. If the part is powered down by setting the DSR POWERDOWN bit, IDLE pin is set high.
FDS0#	33	O		**FLOPPY DRIVE SELECT 0:** Decoded floppy drive select for drive 0. These outputs are decoded from the select bits in the DOR and are a function of the mapping based on BOOTSEL bits in the TDR.
FDS1#/PD#	31	O		**FLOPPY DRIVE MOTOR ENABLE** or **PD:** One of these is selected based on the level of the 44PDEN bit in the auto powerdown command. **FLOPPY DRIVE SELECT 1:** Decoded floppy drive select for drive 1. These outputs are decoded from the select bits in the DOR and are a function of the mapping based on BOOTSEL bits in the TDR. **POWERDOWN:** This pin is active low whenever the part is in powerdown state, either via DSR POWERDOWN bit or via the Auto Powerdown Mode. This pin can be used to disable an external oscillator's output.

Table 1.0. 82078 (44 Pin) Description (Continued)

Symbol	Pin #	I/O	@ H/W Reset	Description
POWER AND GROUND SIGNALS				
V_{CC}	7		N/A	**Power Supply***
V_{SSP}	6		N/A	**GROUND:** 0V
V_{SS}	26 37		N/A	**GROUND:** 0V
AV_{CC}	18		N/A	**ANALOG VOLTAGE**
V_{CCF}	28		N/A	**VOLTAGE:** +5V for a 5V floppy drive, +3.3V for a 3.3V drive.
AV_{SS}	17		N/A	**ANALOG GROUND**

NOTE:
*The digital power supply V_{CC} and the analog power supply AV_{CC} should either be the same or regulated to be within 0.1V of either.

1.0 INTRODUCTION

The 82078 (44 pin) enhanced floppy disk controller incorporates several new features allowing for easy implementation in both the portable and desktop markets. It provides a low cost, small form factor solution targeted for 5.0V and 3.3V platforms that do not require more than two drive support.

The 82078 (44 pin) implements these new features while remaining functionally compatible with 82077SL/82077AA/8272A floppy disk controllers.

Together with a 24 MHz crystal, a resistor package and a device chip select, these devices allow for the most integrated solution available. The integrated analog PLL data separator has better performance than most board level discrete PLL implementations and can be operated at 1 Mbps/500 Kbps/ 300 Kbps/250 Kbps. A 16-byte FIFO substantially improves system performance especially in multi-master systems (e.g. Microchannel, EISA).

Figure 1-1 is a block diagram of the 82078.

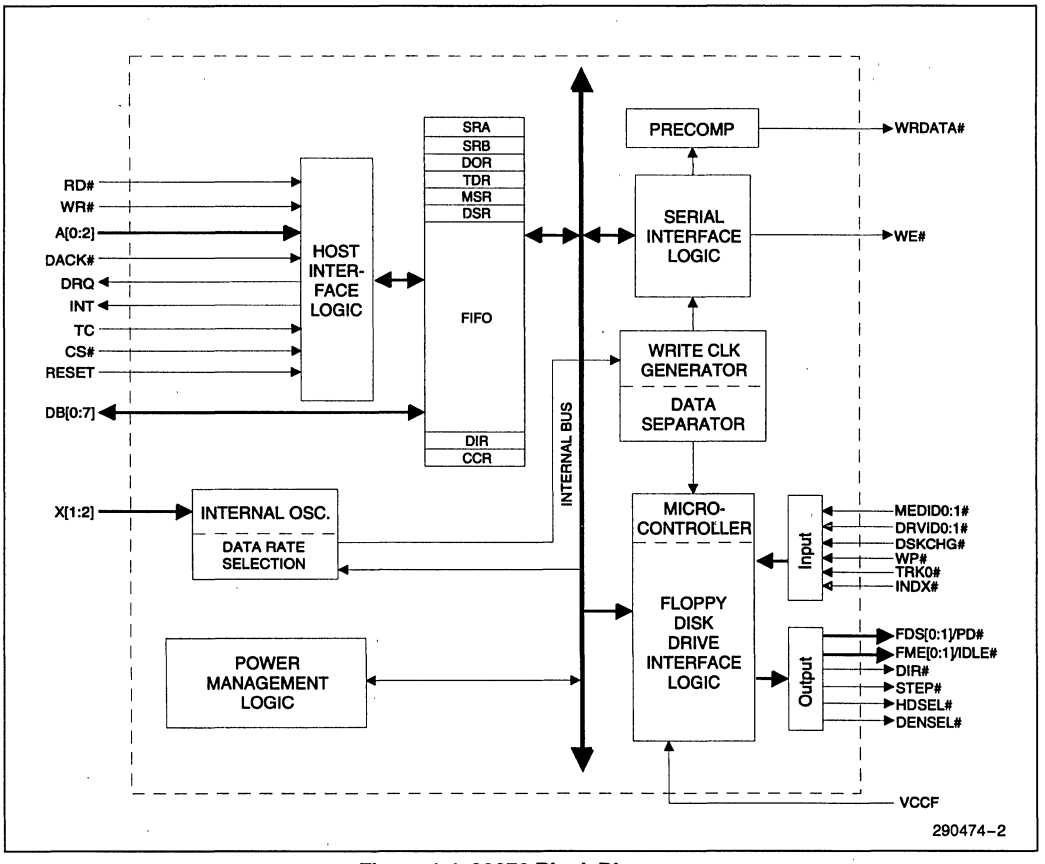


Figure 1-1. 82078 Block Diagram

2.0 MICROPROCESSOR INTERFACE

The interface consists of the standard asynchronous signals: RD#, WR#, CS#, A0–A2, INT, DMA control and a data bus. The address lines select between configuration registers, the FIFO and control/status registers.

2.1 Status, Data, and Control Registers

As shown below, the base address range is supplied via the CS# pin. For PC-AT or PS/2 designs, the primary and secondary address ranges are 3F0 Hex to 3F7 Hex and 370 Hex to 377 Hex respectively.

A2	A1	A0	Access Type	Register	
0	0	0		Reserved	
0	0	1	R/W	Status Register B	SRB
0	1	0	R/W	Digital Output Register	DOR
0	1	1	R/W	Tape Drive Register	TDR
1	0	0	R	Main Status Register	MSR
1	0	0	W	Data Rate Select Register	DSR
1	0	1	R/W	Data (First In First Out)	FIFO
1	1	0		Reserved	
1	1	1	R	Digital Input Register	DIR
1	1	1	W	Configuration Control Register	CCR

In the following sections, the various registers are shown in their powerdown state. The "UC" notation stands for a value that is returned without change from the active mode. The notation "*" means that the value is reflecting the required status (for powerdown). "N/A" means not applicable. "X" indicates that the value is undefined.

2.1.1 STATUS REGISTER B (SRB, EREG EN = 1)

In the AT/EISA mode the SRB is made available whenever the EREG EN bit in the auto powerdown command is set. The register functionality is defined as follows (bits 7 through 3 are reserved):

SRB								
R/W	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R	**RSVD**	**RSVD**	**RSVD**	**RSVD**	**RSVD**	**IDLEMSK**	**PD**	**IDLE**
H/W Reset	X	X	X	X	X	0	PD	IDLE
Auto PD	X	X	X	X	X	UC	UC	UC
W	0	0	0	0	0	IDLEMSK	RSVD	RSVD
H/W Reset	N/A	N/A	N/A	N/A	N/A	0	N/A	N/A
Auto PD	N/A	N/A	N/A	N/A	N/A	UC	N/A	N/A

PD and IDLE reflect the inverted values on the corresponding pins when 44PD EN = 1 (these pins are muxed with FDS1 and FDME1). The signal on the IDLE# pin can be masked by setting IDLEMSK bit high in this register. The IDLE bit will remain unaffected. Since some systems will use the IDLE# pin to provide interrupt to the SMM power management, its disabling allows less external interrupt logic and reduction in board space. Only hardware reset will clear the IDLEMSK bit to zero. When the IDLEMSK bit is set, there is no way to distinguish between autopowerdown and DSR powerdown.

NOTE:
The 44 pin versions of the 82078 are designed to support *either* PD# and IDLE# or FDME1# and FDS1#, but not both simultaneously.

IDLEMSK	IDLE# (pin)
0	unmasked
1	masked

2.1.2 DIGITAL OUTPUT REGISTER (DOR)

The Digital Output Register contains the drive select and motor enable bits, a reset bit and a DMAGATE# bit.

Bits	7	6	5	4	3	2	1	0
Function	RSVD	RSVD	MOT EN1	MOT EN0	DMA GATE#	RESET#	RSVD	DRIVE SEL
H/W Reset State	0	0	0	0	0	0	0	0
Auto PD State	0	0	0*	0*	UC	1*	UC	UC

The MOT ENx bits directly control their respective motor enable pins (FDME0–1). The DRIVE SEL bit is decoded to provide four drive select lines and only one may be active at a time. Standard programming practice is to set both MOT ENx and DRIVE SELx bits at the same time.

NOTE:
The 44 pin versions of the 82078 are designed to support *either* PD# and IDLE# or FDME1# and FDS1#, but not both simultaneously.

Table 2-1 lists a set of DOR values to activate the drive select and motor enable for each drive.

Table 2-1. Drive Activation Value

Drive	DOR Value
0	1CH
1	2DH

The DMAGATE# bit is enabled only in PC-AT. If DMAGATE# is set low, the INT and DRQ outputs are tri-stated and the DACK# and TC inputs are disabled. DMAGATE# set high will enable INT, DRQ, TC, and DACK# to the system.

The DOR reset bit and the Motor Enable bits have to be inactive when the 82078 is in powerdown. The DMAGATE# and DRIVE SEL bits are unchanged. During powerdown, writing to the DOR does not awaken the 82078 with the exception of activating any of the motor enable bits. Setting the motor enable bits active (high) will wake up the part.

This RESET# bit clears the basic core of the 82078 and the FIFO circuits when the LOCK bit is set to "0" (see Section 5.3.2 for LOCK bit definitions). Once set, it remains set until the user clears this bit. This bit is set by a chip reset and the 82078 is held in a reset state until the user clears this bit. The RESET# bit has no effect upon the register.

2.1.3 ENHANCED TAPE DRIVE REGISTER (TDR)

TDR								
R/W	**7***	**6***	**5***	**4***	**3***	**2***	**1**	**0**
R	**RSVD**	**RSVD**	**RSVD**	**RSVD**	**RSVD**	**BOOTSEL**	**TAPESEL1**	**TAPESEL0**
H/W Reset	N/A	N/A	N/A	N/A	N/A	0	0	0
Auto PD	N/A	N/A	N/A	N/A	N/A	UC	UC	UC
W	0	0	0	0	0	BOOTSEL	TAPESEL1	TAPESEL0
H/W Reset	N/A	N/A	N/A	N/A	N/A	0	0	0
Auto PD	N/A	N/A	N/A	N/A	N/A	UC	UC	UC

NOTE:
*These bits are only available when EREG EN = 1, otherwise the bits are tri-stated.

This register allows the user to assign tape support to a particular drive during initialization. Any future references to that drive number automatically invokes tape support. Hardware reset clears this register; software resets have no effect. The tape select bits are hardware RESET to zeros, making Drive 0 **not** available for tape support. Drive 0 is reserved for the floppy boot drive.

The BOOTSEL bit in the 44 pin part is used to remap the drive selects and motor enables. The functionality is as described below:

44PD EN	BOOTSEL(TDR)	Mapping
0	0	Default → DS0 → FDS0, ME0 → FDME0 DS1 → FDS1, ME1 → FDME1
0	1	DS0 → FDS1, ME0 → FDME1 DS1 → FDS0, ME1 → FDME0
1	X	DS0 → FDS0, ME0 → FDME0 DS1 → PD, ME1 → IDLE

The 44PD EN bit in the Auto Powerdown command has precedence over the BOOTSEL bit mapping as shown above.

2.1.4 DATARATE SELECT REGISTER (DSR)

Bits	7	6	5	4	3	2	1	0
Function	S/W RESET	POWER DOWN	PDOSC	PRE-COMP2	PRE COMP1	PRE COMP0	DRATE SEL1	DRATE SEL0
H/W Reset State	0	0	0	0	0	0	1	0
Auto PD State	S/W RESET	POWER DOWN	PDOSC	PRE-COMP2	PRE-COMP1	PRE-COMP0	DRATE SEL1	DRATE SEL0

This register ensures backward compatibility with the 82072 floppy controller and is write-only. Changing the data rate changes the timings of the drive control signals. To ensure that drive timings are not violated when changing data rates, choose a drive timing such that the fastest data rate will not violate the timing.

The PDOSC bit is used to implement crystal oscillator power management. The internal oscillator in the 82078 can be programmed to be either powered on or off via PDOSC. This capability is independent of the chip's powerdown state. Auto powerdown mode and powerdown via the POWERDOWN bit have no effect over the power state of the oscillator.

In the default state the PDOSC bit is low and the oscillator is powered up. When this bit is programmed to a one, the oscillator is shut off. Hardware reset clears this bit to a zero. Neither of the software resets (via DOR or DSR) have any effect on this bit. Note, PDOSC should only be set high when the part is in the powerdown state, otherwise the part will not function correctly and must be hardware reset once the oscillator has turned back on and stabilized. Setting the PDOSC bit has no effect on the clock input to the 82078 (the X1 pin). The clock input is separately disabled when the part is powered down. The SAVE command checks the status of PDOSC, however the RESTORE command will not restore the bit high.

S/W RESET behaves the same as DOR RESET except that this reset is self cleaning.

POWERDOWN bit implements direct powerdown. Setting this bit high will put the 82078 into the powerdown state regardless of the state of the part. The part is internally reset and then put into powerdown. No status is saved and any operation in progress is aborted. A hardware or software reset will exit the 82078 from this powerdown state.

PRECOMP 0–2 adjusts the WRDATA output to the disk to compensate for magnetic media phenomena known as bit shifting. The data patterns that are susceptible to bit shifting are well understood and the 82078 compensates the data pattern as it is written to the disk. The amount of pre-compensation is dependent upon the drive and media but in most cases the default value is acceptable.

Table 2-2. Precompensation Delays

PRECOMP	Precompensation Delays
DSR[4,3,2]	x1 @ 24 MHz
111	0.00 ns – Disabled
001	41.67
010	83.34
011	125.00
100	166.67
101	208.33
110	250.00
000	DEFAULT

Table 2-3. Default Precompensation Delays

Data Rate	Precompensation Delays (ns)
1 Mbps	41.67
0.5 Mbps	125
0.3 Mbps	125
0.25 Mbps	125

The 82078 starts pre-compensating the data pattern starting on Track 0. The CONFIGURE command can change the track that pre-compensating starts on. Table 2-2 lists the pre-compensation values that can be selected and Table 2-3 lists the default pre-compensation values. The default value is selected if the three bits are zeroes.

DRATE 0–1 select one of the four data rates as listed in Table 2-4. The default value is 250 Kbps upon after a "Hardware" reset. Other "Software" Resets do not affect the DRATE or PRECOMP bits.

Table 2-4. Data Rates

DRATESEL0	DRATESEL1	DATA RATE
1	1	1 Mbps
0	0	500 Kbps
0	1	300 Kbps
1	0	250 Kbps

2.1.5 MAIN STATUS REGISTER (MSR)

Bits	7	6	5	4	3	2	1	0
Function	RQM	DIO	NON DMA	CMD BSY	RSVD	RSVD	DRV1 BUSY	DRV0 BUSY
H/W Reset State	0	X	X	X	X	X	X	X
Auto PD State	1	0	0	0	0	0	0	0

The Main Status Register is a read-only register and is used for controlling command input and result output for all commands.

RQM—Indicates that the host can transfer data if set to 1. No access is permitted if set to a 0.

DIO—Indicates the direction of a data transfer once RQM is set. A 1 indicates a read and a 0 indicates a write is required.

NON-DMA—This mode is selected in the SPECIFY command and will be set to a 1 during the execution phase of a command. This is for polled data transfers and helps differentiate between the data transfer phase and the reading of result bytes.

COMMAND BUSY—This bit is set to a one when a command is in progress. It goes active after the command byte has been accepted and goes inactive at the end of the results phase. If there is no result phase (SEEK, RECALIBRATE commands), the bit returns to a 0 after the last command byte.

DRV x BUSY—These bits are set to ones when a drive is in the seek portion of a command, including seeks and recalibrates.

Some example values of the MSR are:

- MSR = 80H; The controller is ready to receive a command.
- MSR = 90H; executing a command or waiting for the host to read status bytes (assume DMA mode).
- MSR = D0H; waiting for the host to write status bytes.

2.1.6 FIFO (DATA)

All command parameter information and disk data transfers go through the FIFO. The FIFO is 16 bytes in size and has programmable threshold values. Data transfers are governed by the RQM and DIO bits in the Main Status Register.

The FIFO defaults to an 8272A compatible mode after a "Hardware" reset (Reset via pin 32). "Software" Resets (Reset via DOR or DSR register) can also place the 82078 into 8272A compatible mode if the LOCK bit is set to "0" (See the definition of the LOCK bit), maintaining PC-AT hardware compatibility. The default values can be changed through the CONFIGURE command (enable full FIFO operation with threshold control). The advantage of the FIFO is that it allows the system a larger DMA latency without causing a disk error. Table 2-5 gives several examples of the delays with a FIFO. The data is based upon the following formula:

$$\text{Threshold\#} \times 1/\text{DATA RATE} \times 8 - 1.5\ \mu s = \text{DELAY}$$

Table 2-5. Delay Servicing Time

FIFO Threshold Examples	Maximum Delay to Servicing at 1 Mbps Data Rate*
1 byte	$1 \times 8\ \mu s - 1.5\ \mu s = 6.5\ \mu s$
2 bytes	$2 \times 8\ \mu s - 1.5\ \mu s = 14.5\ \mu s$
8 bytes	$8 \times 8\ \mu s - 1.5\ \mu s = 62.5\ \mu s$
15 bytes	$15 \times 8\ \mu s - 1.5\ \mu s = 118.5\ \mu s$

NOTE:
*Not available on the 82078-5.

FIFO Threshold Examples	Maximum Delay to Servicing at 500 Kbps Data Rate*
1 byte	$1 \times 16\ \mu s - 1.5\ \mu s = 14.5\ \mu s$
2 bytes	$2 \times 16\ \mu s - 1.5\ \mu s = 30.5\ \mu s$
8 bytes	$8 \times 16\ \mu s - 1.5\ \mu s = 126.5\ \mu s$
15 bytes	$15 \times 16\ \mu s - 1.5\ \mu s = 238.5\ \mu s$

At the start of a command, the FIFO action is always disabled and command parameters must be sent based upon the RQM and DIO bit settings. As the 82078 enters the command execution phase, it clears the FIFO of any data to ensure that invalid data is not transferred. An overrun or underrun will terminate the current command and the transfer of data. Disk writes will complete the current sector by generating a 00 pattern and valid CRC.

2.1.7 DIGITAL INPUT REGISTER (DIR)

Only bit 7 is driven, all other bits remain tri-stated.

Bits	7	6	5	4	3	2	1	0
Function	DSK CHG#	—	—	—	—	—	—	—
H/W Reset State	DSK CHG#	—	—	—	—	—	—	—
Auto PD State	0	—	—	—	—	—	—	—

NOTE:
(—) means these bits are tri-stated.

DSKCHG# monitors the pin of the same name and reflects the opposite value seen on the disk cable. The DSKCHG# bit is forced inactive along with all the inputs from the floppy disk drive. All the other bits remain tri-stated.

2.2 Reset

There are three sources of reset on the 82078; the RESET pin, a reset generated via a bit in the DOR and a reset generated via a bit in the DSR. All resets take the 82078 out of the powerdown state.

In entering the reset state, all operations are terminated and the 82078 enters an idle state. Activating reset while a disk write activity is in progress will corrupt the data and CRC.

On exiting the reset state, various internal registers are cleared, and the 82078 waits for a new command. Drive polling will start unless disabled by a new CONFIGURE command.

2.2.1 RESET PIN ("HARDWARE") RESET

The RESET pin is a global reset and clears all registers except those programmed by the SPECIFY command. The DOR Reset bit is enabled and must be cleared by the host to exit the reset state.

2.2.2 DOR RESET vs DSR RESET ("SOFTWARE" RESET)

These two resets are functionally the same. The DSR Reset is included to maintain 82072 compatibility. Both will reset the 8272 core which affects drive status information. The FIFO circuits will also be reset if the LOCK bit is a "0" (see definition of the LOCK bit). The DSR Reset clears itself automatically while the DOR Reset requires the host to manually clear it. DOR Reset has precedence over the DSR Reset. The DOR Reset is set automatically upon a pin RESET. The user must manually clear this reset bit in the DOR to exit the reset state.

The t30a specification in the A.C. Specifications gives the minimum amount of time that the DOR reset must be held active. This amount of time that the DOR reset must be held active is dependent upon the data rate. 82078 requires that the DOR reset bit must be held active for at least 0.5 μs at 250 Kbps. This is less than a typical ISA I/O cycle time.

2.3 DMA Transfers

DMA transfers are enabled with the SPECIFY command and are initiated by the 82078 by activating the DRQ pin during a data transfer command. The FIFO is enabled directly by asserting DACK# and addresses need not be valid (CS# can be held inactive during DMA transfers).

3.0 DRIVE INTERFACE

The 82078 has integrated all of the logic needed to interface to a floppy disk or a tape drive which use floppy interface. All drive outputs have 12 mA drive capability and all inputs use a receive buffer with hysteresis. The internal analog data separator requires no external components, yet allows for an extremely wide capture range with high levels of read-data jitter, and ISV. The designer needs only to run the 82078 disk drive signals to the disk or tape drive connector.

3.1 Cable Interface

Generally, 5.25" drive uses open collector drivers and 3.5" drives use totem-pole drivers. The output buffers on the 82078 do not change between open collector or totem-pole, they are always totem-pole.

DRVDEN0 and DRVDEN1 connect to pins 2 and 6 or 33 (on most disk drives) to select the data rate sent from the drive to the 82078. The polarity of DRVDEN0 and DRVDEN1 can be programmed through the Drive Specification command (see the command description for more information).

3.2 Host and FDD Interface Drivers

The chart below shows the drive capabilities of the 82078.

Drive Requirement	3.3V (I_{OL}/I_{OH})	5.0V (I_{OL}/I_{OH})
82078 Drivers	FDD = 6 mA/−2 mA SYS = 6 mA/−2 mA	FDD = 12 mA/−4 mA SYS = 12 mA/−4 mA

Today's floppy disk drives have reduced the output buffer's drive requirements on the floppy drive interface to 6 mA per drive at 5.0V. To support 2 drives, the drive output buffer drive capability needs to be 12 mA (at 5.0V). This is a reduction from 40 mA needed on the 82077SL. At 3.3V the 82078 halves the drive capability to 6 mA (3 mA per drive).

The slew rate control on the output buffers of the 82078 has been changed to reduce noise. The di/dt of the output drivers has been controlled such that the noise on the signal is minimized. The transition times are illustrated in the table below:

Signal Edge	Transition Time (ns)
t_{HL}	>5 ns
t_{LH}	>5 ns

NOTE:
*At 5.6V, 0°C, 50 pF load, 10% V_{CC} to 90% V_{CC}.

3.3 Data Separator

The function of the data separator is to lock onto the incoming serial read data. When lock is achieved the serial front end logic of the chip is provided with a clock which is synchronized to the read data. The synchronized clock, called Data Window, is used to internally sample the serial data. One state of Data Window is used to sample the data portion of the bit cell, and the alternate state samples the clock portion. Serial to parallel conversion logic separates the read data into clock and data bytes.

To support reliable disk reads the data separator must track fluctuations in the read data frequency. Frequency errors primarily arise from two sources: motor rotation speed variation and instantaneous speed variation (ISV). A second condition, and one that opposes the ability to track frequency shifts is the response to bit jitter.

The internal data separator consists of two analog phase lock loops (PLLs) as shown in Figure 3-1. The two PLLs are referred to as the reference PLL and the data PLL. The reference PLL (the master PLL) is used to bias the data PLL (the slave PLL). The reference PLL adjusts the data PLL's operating point as a function of process, junction temperature and supply voltage. Using this architecture it was possible to eliminate the need for external trim components.

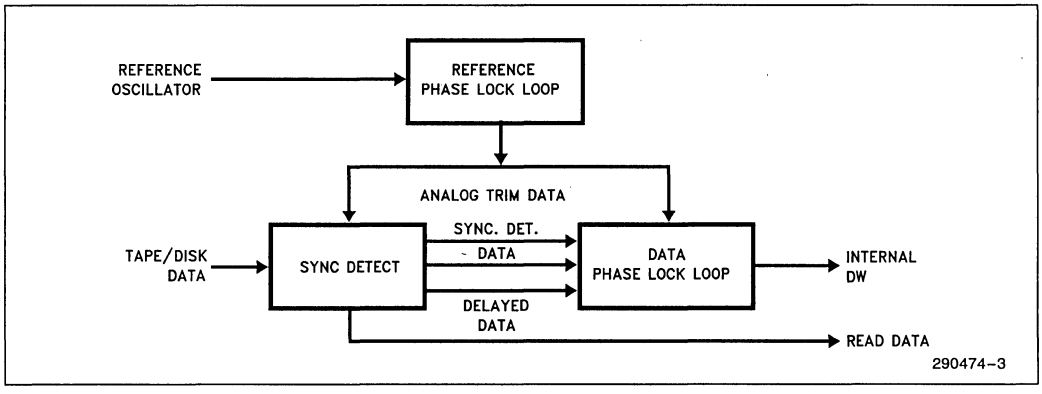


Figure 3-1. Data Separator Block Diagram

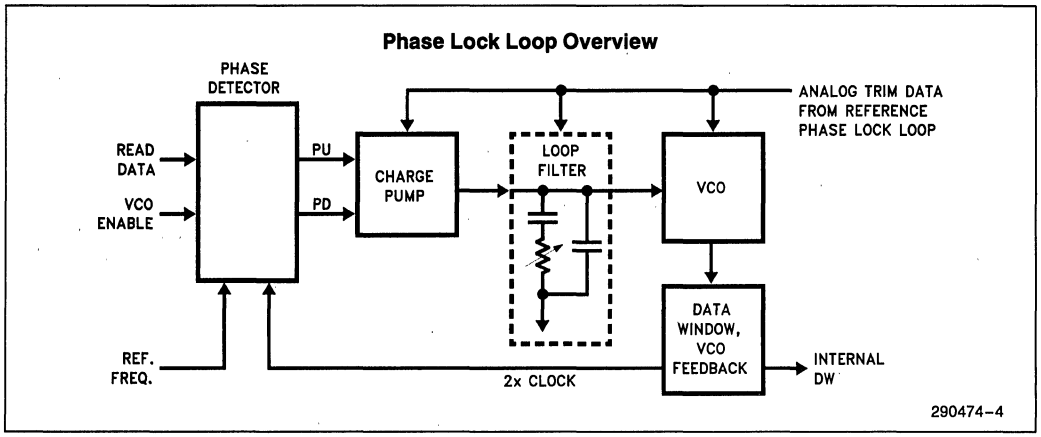


Figure 3-2. Data PLL

Figure 3-2 shows the data PLL. The reference PLL has control over the loop gain by its influence on the charge pump and the VCO. In addition, the reference PLL controls the loop filter time constant. As a result, the closed loop transfer function of the data PLL is controlled, and immune to the first order, to environmental factors and process variation.

Systems with analog PLLs are often very sensitive to noise. In the design of this data separator, many steps were taken to avoid noise sensitivity problems. The analog section of the chip has a separate V_{SS} pin (AV_{SS}) which should be connected externally to a noise free ground. This provides a clean basis for V_{SS} referenced signals. In addition, many analog circuit features were employed to make the overall system as insensitive to noise as possible.

3.3.1 JITTER TOLERANCE

The jitter immunity of the system is dominated by the data PLL's response to phase impulses. This is measured as a percentage of the theoretical data window by dividing the maximum readable bit shift by a ¼ bitcell distance. For instance, if the maximum allowable bit shift is 300 ns for a 500 Kbps data stream, the jitter tolerance is 60%.

3.3.2 LOCKTIME (t_{LOCK})

The lock, or settling time of the data PLL is designed to be 64-bit times (8 sync bytes). The value assumes that the sync field jitter is 5% the bit cell or less. This level of jitter is realistic for a constant bit pattern. Intersymbol interference should be equal, thus nearly eliminating random bit shifting.

3.3.3 CAPTURE RANGE

Capture Range is the maximum frequency range over which the data separator will acquire phase lock with the incoming RDDATA signal. In a floppy disk environment, this frequency variation is composed of two components: drive motor speed error and ISV. Frequency is a factor which may determine the maximum level of the ISV (Instantaneous Speed Variation) component. In general, as frequency increases the allowed magnitude of the ISV component will decrease. When determining the capture range requirements, the designer should take the maximum amount of frequency error for the disk drive and double it to account for media switching between drives.

3.4 Write Precompensation

The write precompensation logic is used to minimize bit shifts in the RDDATA stream from the disk drive. The shifting of bits is a known phenomena of magnetic media and is dependent upon the disk media AND the floppy drive.

The 82078 monitors the bit stream that is being sent to the drive. The data patterns that require precompensation are well known. Depending upon the pattern, the bit is shifted either early or late (or not at all) relative to the surrounding bits. Figure 3-3 is a block diagram of the internal circuit.

The top block is a 13-bit shift register with the no delay tap being in the center. This allows 6 levels of early and late shifting with respect to nominal. The shift register is clocked at the main clock rate (24 MHz). The output is fed into 2 multiplexors, one for early and one for late. A final stage of multiplexors combines the early, late and normal data stream back into one which is the WRDATA output.

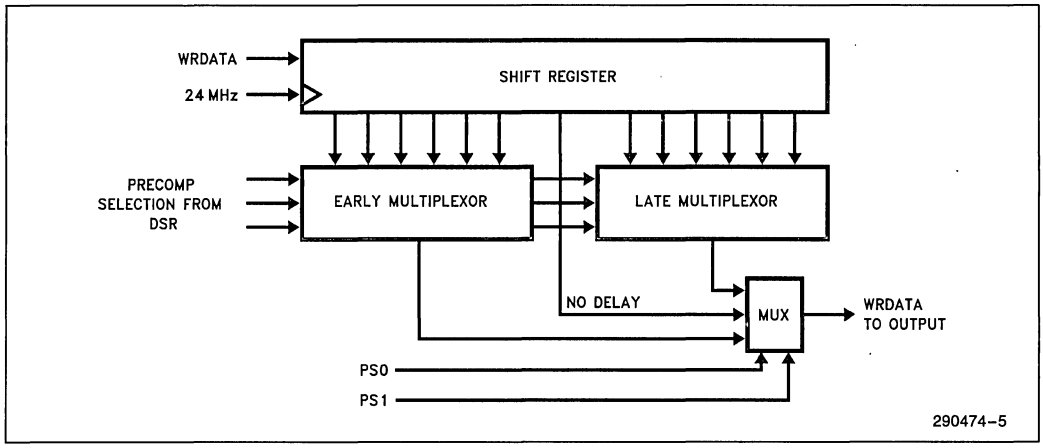


Figure 3-3. Precompensation Block Diagram

4.0 POWER MANAGEMENT FEATURES

The 82078 contains power management features that makes it ideal for design of portable personal computers. These features can be classified into power management of the part and that of the internal oscillator. The powerdown of the part is done independently of the internal oscillator in the 82078.

4.1 Power Management Scheme

The portable market share of the personal computing market has increased significantly. To improve power conservation on portable platforms, designs are migrating from 5.0V to 3.3V. Intel's 82078-3 allows designers to incorporate 3.3V floppy disk controller support in their systems.

The 82078 supports two powerdown modes, direct powerdown and automatic powerdown. Direct powerdown refers to direct action by the software to powerdown without dependence on external factors. Automatic powerdown results from 82078's monitoring of the current conditions according to a previously programmed mode. Any hardware reset disables the automatic POWERDOWN command, however software resets have no effect on the command. The 82078 also supports powerdown of its internal crystal oscillator independent of the powerdown modes described above. By setting bit 5 (PDOSC) in the DSR register, the internal oscillator is turned off. This bit has sole control of the oscillator powerdown, allowing the internal clock to be turned off when an external oscillator is used.

4.2 Oscillator Power Management

The 82078 supports a built-in crystal oscillator that can be programmed to be either powered down or active, independent of the power state of the chip. This capability is implemented by the PDOSC bit in the DSR. When PDOSC is set low, the internal oscillator is on. When PDOSC is set high, the internal oscillator is off. Note, a DSR powerdown does not turn off the oscillator.

When the external oscillator is used, power can be saved by turning off the internal oscillator. If the internal oscillator is used, the oscillator may be powered up (even when the rest of the chip is powered off) allowing the chip to wake up quickly and in a stable state. It is recommended to keep the internal oscillator on even when in the powerdown state. The main reason for this is that the recovery time of the oscillator during wake up may take tens of milliseconds under the worst case, which may create problems with any sensitive application software. In a typical application the internal oscillator should be on unless the system goes into a power saving or standby mode (such a mode request would be made by a system time out or by a user). In this case, the system software would take over and must turn on the oscillator sufficiently ahead of awakening the part.

In the case of the external oscillators, the power up characteristics are similar. If the external source remains active during the time the 82078 is powered down, then the recovery time effect is minimized. The PD# pin can be used to turn off the external source. While the PD# pin is active 82078 does not require a clock source. However, when the PD# pin is inactive, the clocking source, once it starts oscillating, must be completely stable to ensure that the 82078 operates properly.

4.3 Part Power Management

This section deals with the power management of the rest of the chip excluding the oscillator. This section explains powerdown modes and wake up modes.

4.3.1 DIRECT POWERDOWN

Direct powerdown is conducted via the POWERDOWN bit in the DSR register (bit 6). Programming this bit high will powerdown 82078. All status is lost if this type of powerdown mode is used. The part can exit powerdown from this mode via any hardware or software reset. This type of powerdown overrides the automatic powerdown. When the part is in automatic powerdown and the DSR powerdown is issued, the previous status of the part is lost and the 82078 resets to software default values.

4.3.2 AUTO POWERDOWN

Automatic powerdown is conducted via a "Powerdown Mode" command. There are four conditions required before the part will enter powerdown. All of these conditions must be true for the part to initiate the powerdown sequence. These conditions follow:

1. The motor enable pins FDME[0:1] must be inactive.
2. The part must be idle; this is indicated by MSR = 80H and INT = 0 (INT may be high even if MSR = 80H due to polling interrupt).
3. The Head Unload Timer (HUT, explained in the SPECIFY command) must have expired.
4. The auto powerdown timer must have timed out.

The command can be used to enable powerdown by setting the AUTO PD bit in the command to high. The command also provides a capability of programming a minimum power up time via the MIN DLY bit in the command. The minimum power up time refers to a minimum amount of time the part will remain powered up after being awakened or reset. An internal timer is initiated as soon as the auto powerdown command is enabled. The part is then powered down provided all the remaining conditions are met. Any software reset will reinitialize the timer. Changing of data rate extends the auto powerdown timer by up to 10 ms, but only if the data rate is changed during the countdown.

Disabling the auto powerdown mode cancels the timers and holds the 82078 out of auto powerdown.

The IDLE# pin can be masked via the IDLEMSK bit in Status Register B (EREG EN = 1).

4.3.3 WAKE UP MODES

This section describes the conditions for awakening the part from both direct and automatic powerdown. Power conservation or extension of battery life is the main reason power management is required. This means that the 82078 must be kept in powerdown state as long as possible and should be powered up as late as possible without compromising software transparency.

To keep the part in powerdown mode as late as possible implies that the part should wake up as fast as possible. However, some amount of time is required for the part to exit powerdown state and prepare the internal microcontroller to accept commands. Application software is very sensitive to such a delay and in order to maintain software transparency, the recovery time of the wake up process must be carefully controlled by the system software.

4.3.3.1 Wake Up from DSR Powerdown

If the 82078 enters the powerdown through the DSR powerdown bit, it must be reset to exit. Any form of software or hardware reset will serve, although DSR is recommended. No other register access will awaken the part, including writing to the DOR's motor enable (FDME[0:1]) bits.

If DSR powerdown is used when the part is in auto powerdown, the DSR powerdown will override the auto powerdown. However, when the part is awakened by a software reset, the auto powerdown command (including the minimum delay timer) will once again become effective as previously programmed. If the part is awakened via a hardware reset, the auto powerdown is disabled.

After reset, the part will go through a normal sequence. The drive status will be initialized. The FIFO mode will be set to default mode on a hardware reset or on a software reset if the LOCK command has not blocked it. Finally, after a delay, the polling interrupt will be issued.

4.3.3.2 Wake Up from Auto Powerdown

If the part enters the powerdown state through the auto powerdown mode, then the part can be awakened by reset or by appropriate access to certain registers.

If a hardware or software reset is used then the part will go through the normal reset sequence. If the access is through the selected registers, then the 82078 resumes operation as though it was never in powerdown. Besides activating the RESET pin or

one of the software reset bits in the DOR or DSR, the following register accesses will wake up the part:

1. Enabling any one of the motor enable bits in the DOR register (reading the DOR does not awaken the part)
2. A read from the MSR register
3. A read or write to the FIFO register

Any of these actions will wake up the part. Once awake, 82078 will reinitiate the auto powerdown timer for 10 ms or 0.5s (depending on the MIN DLY bit the auto powerdown command). The part will powerdown again when all the auto powerdown conditions are satisfied.

4.4 Register Behavior

The register descriptions and their values in the powerdown state are listed in the Microprocessor Interface section. Table 4-1 reiterates the configuration registers available. It also shows the type of access permitted. In order to maintain software transparency, access to all the registers must be maintained. As Table 4-1 shows, two sets of registers are distinguished based on whether their access results in the part remaining in powerdown state or exiting it.

Table 4-1. 82078 Register Behavior

Address	Available Registers	Access
Access to these registers DOES NOT wake up the part		
000	—	
001	SRB (EREG EN = 1)	R/W
010	DOR*	R/W
011	TDR	R/W
100	DSR*	W
110	—	—
111	DIR	R
111	CCR	W
Access to these registers wakes up the part		
100	MSR	R
101	FIFO	R/W

NOTE:
*Writing to the DOR or DSR does not wake up the part, however, writing any of the motor enable bits or doing a software reset (either via DOR or DSR reset bits) will wake up the part.

Access to all other registers is possible without awakening the part. These registers can be accessed during powerdown without changing the status of the part. A read from these registers will reflect the true status as shown in the register description in Section 2.1. A write to the part will result in the part retaining the data and subsequently reflecting it when the part awakens. Accessing the part during powerdown may cause an increase in the power consumption by the part. The part will revert back to its low power mode when the access has been completed. None of the extended registers effect the behavior of the powerdown mode.

4.5 Pin Behavior

The 82078 is specifically designed for the portable PC systems in which the power conservation is a primary concern. This makes the behavior of the pins during powerdown very important.

The pins of 82078 can be divided into two major categories; system interface and floppy disk drive interface. The floppy disk drive pins are disabled such that no power will be drawn through the 82078 as a result of any voltage applied to the pin within the 82078's power supply range. The floppy disk drive interface pins are configurable by the FDI TRI bit in the auto powerdown command. When the bit is set the output pins of the floppy disk drive retain their original state. All other pins are either disabled or unchanged as depicted in Table 4-4. Most of the system interface pins are left active to monitor system accesses that may wake up the part.

4.5.1 System Interface Pins

Table 4-2 gives the state of the system interface pins in the powerdown state. Pins unaffected by powerdown are labeled "UC". Input pins are "DISABLED" to prevent them from causing currents internal to the 82078 when they have indeterminate input values.

Table 4-2. System Interface Pins

System Pins	State In Power Down	System Pins	State In Power Down
Input Pins		Output Pins	
CS#	UC	DRQ	UC (Low)
RD#	UC	INT	UC (Low)
WR#	UC	PD#*	HIGH
A[0:2]	UC	IDLE#*	High (Auto PD) Low (DSR PD)
DB[0:7]	UC	DB[0:7]	UC
RESET	UC		
DACK#	Disabled		
TC	Disabled		
X[1:2]	Programmable		

NOTE:
*These pins are muxed with FDS1 and FDME1 and are only available when 44PD EN = 1.

Two pins which can be used to indicate the status of the part are IDLE# and PD#. Table 4-3 shows how these pins reflect the 82078 status. Note that these pins are only enabled when 44PD EN = 1.

Table 4-3. 82078 Status Pins

PD	IDLE	MSR	Part Status
1	1	80H	Auto Powerdown
1	0	RQM = 1; MSR[6:0] = X	DSR Powerdown
0	1	80H	Idle
0	0	—	Busy

The IDLE# pin indicates when the part is in idle state and can be powered down. It is a combination of MSR equaling 80H, the head being unloaded and the INT pin being low. As shown in the table, the IDLE# pin will be low when the part is in DSR powerdown state. The PD# pin is active whenever the part is in the powerdown state. It is active for either mode of powerdown. The PD# pin can be used to turn off an external oscillator of other floppy disk drive interface hardware.

4.5.2 FDD INTERFACE PINS

The FDD interface "input" pins during powerdown are disabled or unchanged as shown in Table 4-4. The floppy disk drive "output" pins are programmable by the FDI TRI bit in the auto powerdown command. Setting of the FDI TRI bit in the auto power down command results in the interface retaining its normal state. When this bit is low (default state) all output pins in the FDD interface to the floppy disk drive itself are tri-stated. Pins used for local logic control or part programming are unaffected. Table 4-4 depicts the state of the floppy disk interface pins in the powerdown state (FDI TRI is low).

Table 4-4. 82078 FDD Interface Pins

FDD Pins	State In Powerdown	FDD Pins	State In Powerdown
Input Pins		Output Pins (FDI TRI = 0)	
RDDATA#	Disabled	FDME[0:1]#	Tristated
WP#	Disabled	FDS[0:1]#	Tristated
TRK0#	Disabled	DIR#	Tristated
INDX#	Disabled	STEP#	Tristated
DSKCHG#	Disabled	WRDATA#	Tristated
		WE#	Tristated
		HDSEL#	Tristated
		DRVDEN[0:1]	Tristated

5.0 CONTROLLER PHASES

For simplicity, command handling in the 82078 can be divided into three phases: Command, Execution and Result. Each phase is described in the following sections.

When there is no command in progress, the 82078 can be in idle, drive polling or powerdown state.

5.1 Command Phase

After a reset, the 82078 enters the command phase and is ready to accept a command from the host. For each of the commands, a defined set of command code bytes and parameter bytes has to be written to the 82078 before the command phase is complete (Please refer to Section 6.0 for the command descriptions). These bytes of data must be transferred in the order prescribed.

Before writing to the 82078, the host must examine the RQM and DIO bits of the Main Status Register. RQM, DIO must be equal to "1" and "0" respectively before command bytes may be written. RQM is set false by the 82078 after each write cycle until the received byte is processed. The 82078 asserts RQM again to request each parameter byte of the command, unless an illegal command condition is detected. After the last parameter byte is received, RQM remains "0", and the 82078 automatically enters the next phase as defined by the command definition.

The FIFO is disabled during the command phase to retain compatibility with the 8272A, and to provide for the proper handling of the "Invalid Command" condition.

5.2 Execution Phase

All data transfers to or from the 82078 occur during the execution phase, which can proceed in DMA or non-DMA mode as indicated in the SPECIFY command.

Each data byte is transferred by an INT or DRQ depending on the DMA mode. The CONFIGURE command can enable the FIFO and set the FIFO threshold value.

The following paragraphs detail the operation of the FIFO flow control. In these descriptions, (threshold) is defined as the number of bytes available to the 82078 when service is requested from the host, and ranges from 1 to 16. The parameter FIFOTHR which the user programs is one less, and ranges from 0 to 15.

A low threshold value (i.e. 2) results in longer periods of time between service requests, but requires faster servicing of the request, for both read and write cases. The host reads (writes) from (to) the FIFO until empty (full), then the transfer request goes inactive. The host must be very responsive to the service request. This is the desired case for use with a "fast" system.

A high value of threshold (i.e. 12) is used with a "sluggish" system by affording a long latency period after a service request, but results in more frequent service requests.

5.2.1 NON-DMA MODE, TRANSFERS FROM THE FIFO TO THE HOST

The INT pin and RQM bits in the Main Status Register are activated when the FIFO contains 16 (or set threshold) bytes, or the last bytes of a full sector transfer have been placed in the FIFO. The INT pin can be used for interrupt driven systems and RQM can be used for polled sytems. The host must respond to the request by reading data from the FIFO. This process is repeated until the last byte is transferred out of the FIFO, then 82078 deactivates the INT pin and RQM bit.

5.2.2 NON-DMA MODE, TRANSFERS FROM THE HOST TO THE FIFO

The INT pin and RQM bit in the Main Status Register are activated upon entering the execution phase of data transfer commands. The host must respond to the request by writing data into the FIFO. The INT pin and RQM bit remain true until the FIFO becomes full. They are set true again when the FIFO has (threshold) bytes remaining in the FIFO. The INT pin will also be deactivated if TC and DACK# both go inactive. The 82078 enters the result phase after the last byte is taken by the 82078 from the FIFO (i.e. FIFO empty condition).

5.2.3 DMA MODE, TRANSFERS FROM THE FIFO TO THE HOST

The 82078 activates the DRQ pin when the FIFO contains 16 (or set threshold) bytes, or the last byte of a full sector transfer has been placed in the FIFO. The DMA controller must respond to the request by reading data from the FIFO. The 82078 will deactivate the DRQ pin when the FIFO becomes empty. DRQ goes inactive after DACK# goes active for the last byte of a data transfer (or on the active edge of RD#, on the last byte, if no edge is present on DACK#) Note that DACK# and TC must overlap for at least 50 ns for proper functionality.

5.2.4 DMA MODE, TRANSFERS FROM THE HOST TO THE FIFO

The 82078 activates the DRQ pin when entering the execution phase of the data transfer commands. The DMA controller must respond by activating the DACK# and WR# pins and placing data in the FIFO. DRQ remains active until the FIFO becomes full. DRQ is again set true when the FIFO has (threshold) bytes remaining in the FIFO. The 82078 will also deactivate the DRQ pin when TC becomes true (qualified by DACK# by overlapping by 50 ns), indicating that no more data is required. DRQ goes inactive after DACK# goes active for the last byte of a data transfer (or on the active edge of WR# of the last byte, if no edge is present on DACK#).

5.2.5 DATA TRANSFER TERMINATION

The 82078 supports terminal count explicitly through the TC pin and implicitly through the underrun/overrun and end-of-track (EOT) functions. For full sector transfers, the EOT parameter can define the last sector to be transferred in a single or multisector transfer. If the last sector to be transferred is a partial sector, the host can stop transferring the data in mid-sector, and the 82078 will continue to complete the sector as if a hardware TC was received. The only difference between these implicit functions and TC is that they return "abnormal termination" result status. Such status indications can be ignored if they were expected.

Note that when the host is sending data to the FIFO of the 82078, the internal sector count will be complete when 82078 reads the last byte from its side of the FIFO. There may be a delay in the removal of the transfer request signal of up to the time taken for the 82078 to read the last 16 bytes from the FIFO. The host must tolerate this delay.

5.3 Result Phase

The generation of INT determines the beginning of the result phase. For each of the commands, a defined set of result bytes has to be read from the 82078 before the result phase is complete. (Refer to Section 6.0 on command descriptions.) These bytes of data must be read out for another command to start.

RQM and DIO must both equal "1" before the result bytes may be read from the FIFO. After all the result bytes have been read, the RQM and DIO bits switch to "1" and "0" respectively, and the CB bit is cleared. This indicates that the 82078 is ready to accept the next command.

6.0 COMMAND SET/DESCRIPTIONS

Commands can be written whenever the 82078 is in the command phase. Each command has a unique set of needed parameters and status results. The 82078 checks to see that the first byte is a valid command and, if valid, proceeds with the command. If it was invalid, the next time the RQM bit in the MSR register is a "1" the DIO and CB bits will also be "1", indicating the FIFO must be read. A result byte of 80H will be read out of the FIFO, indicating an invalid command was issued. After reading the result byte from the FIFO the 82078 will return to the command phase. Table 6-1 is a summary of the Command set.

Table 6-1. 82078 Command Set

Phase	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
READ DATA										
Command	W	MT	MFM	SK	0	0	1	1	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W					C				Sector ID Information Prior to Command Execution
	W					H				
	W					R				
	W					N				
	W					EOT				
	W					GPL				
	W					DTL				
Execution										Data Transfer Between the FDD and System
Result	R					ST 0				Status Information After Command Execution
	R					ST 1				
	R					ST 2				
	R					C				
	R					H				Sector ID Information After Command Execution
	R					R				
	R					N				
READ DELETED DATA										
Command	W	MT	MFM	SK	0	1	1	0	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W					C				Sector ID Information Prior to Command Execution
	W					H				
	W					R				
	W					N				
	W					EOT				
	W					GPL				
	W					DTL				
Execution										Data Transfer Between the FDD and System
Result	R					ST 0				Status Information After Command Execution
	R					ST 1				
	R					ST 2				
	R					C				
	R					H				Sector ID Information After Command Execution
	R					R				
	R					N				
WRITE DATA										
Command	W	MT	MFM	0	0	0	1	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W					C				Sector ID Information Prior to Command Execution
	W					H				
	W					R				
	W					N				
	W					EOT				
	W					GPL				
	W					DTL				
Execution										Data Transfer Between the FDD and System
Result	R					ST 0				Status Information After Command Execution
	R					ST 1				
	R					ST 2				
	R					C				
	R					H				Sector ID Information After Command Execution
	R					R				
	R					N				

Table 6-1. 82078 Command Set (Continued)

Phase	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
WRITE DELETED DATA										
Command	W	MT	MFM	0	0	1	0	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W					C				Sector ID Information Prior
	W					H				to Command Execution
	W					R				
	W					N				
	W					EOT				
	W					GPL				
	W					DTL				
Execution										Data Transfer Between the FDD and System
Result	R					ST 0				Status Information After
	R					ST 1				Command Execution
	R					ST 2				
	R					C				
	R					H				Sector ID Information After
	R					R				Command Execution
	R					N				
READ TRACK										
Command	W	0	MFM	0	0	0	0	1	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W					C				Sector ID Information Prior
	W					H				to Command Execution
	W					R				
	W					N				
	W					EOT				
	W					GPL				
	W					DTL				
Execution										Data Transfer Between the FDD and System. FDC Reads All Sectors from Index Hole to EOT
Result	R					ST 0				Status Information After
	R					ST 1				Command Execution
	R					ST 2				
	R					C				
	R					H				Sector ID Information After
	R					R				Command Execution
	R					N				
VERIFY										
Command	W	MT	MFM	SK	1	0	1	1	0	Command Codes
	W	EC	0	0	0	0	HDS	DS1	DS0	
	W					C				Sector ID Information Prior
	W					H				to Command Execution
	W					R				
	W					N				
	W					EOT				
	W					GPL				
	W					DTL/SC				
Execution										No Data Transfer Takes Place
Result	R					ST 0				Status Information After
	R					ST 1				Command Execution
	R					ST 2				
	R					C				
	R					H				Sector ID Information After
	R					R				Command Execution
	R					N				
VERSION										
Command	W	0	0	0	1	0	0	0	0	Command Code
Result	R	1	0	0	1	0	0	0	0	Enhanced Controller

Table 6-1. 82078 Command Set (Continued)

Phase	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
FORMAT TRACK										
Command	W	0	MFM	0	0	1	1	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				N					Bytes/Sector
	W				SC					Sectors/Cylinder
	W				GPL					Gap3
	W				D					Filler Byte
Execution For Each Sector Repeat:	W				C					Input Sector Parameters
	W				H					
	W				R					
	W				N					
										82078 Formats an Entire Cylinder
Result	R				ST 0					Status Information After Command Execution
	R				ST 1					
	R				ST 2					
	R				Undefined					
	R				Undefined					
	R				Undefined					
	R				Undefined					
SCAN EQUAL										
Command	W	MT	MFM	SK	1	0	0	0	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				C					Sector ID Information Prior to Command Execution
	W				H					
	W				R					
	W				N					
	W				EOT					
	W				GPL					
	W				STP					
Execution										Data Compared Between the FDO and Main-System
Result	R				ST 0					Status Information After Command Execution
	R				ST 1					
	R				ST 2					
	R				C					Sector ID Information After Command Execution
	R				H					
	R				R					
	R				N					

2

Table 6-1. 82078 Command Set (Continued)

Phase	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
SCAN LOW OR EQUAL										
Command	W	MT	MFM	SK	1	1	0	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				C					Sector ID Information
	W				H					Prior to Command
	W				R					Execution
	W				N					
	W				EOT					
	W				GPL					
	W				STP					
Execution										Data Compared Between the FDO and Main-System
Result	R				ST 0					Status Information
	R				ST 1					After Command
	R				ST 2					Execution
	R				C					
	R				H					Sector ID Information
	R				R					After Command
	R				N					Execution
SCAN HIGH OR EQUAL										
Command	W	MT	MFM	SK	1	1	1	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				C					Sector ID Information
	W				H					Prior to Command
	W				R					Execution
	W				N					
	W				EOT					
	W				GPL					
	W				STP					
Execution										Data Compared Between the FDO and Main-System
Result	R				ST 0					Status Information
	R				ST 1					After Command
	R				ST 2					Execution
	R				C					
	R				H					Sector ID Information
	R				R					After Command
	R				N					Execution

Table 6-1. 82078 Command Set (Continued)

Phase	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
RECALIBRATE										
Command	W	0	0	0	0	0	1	1	1	Command Codes
	W	0	0	0	0	0	0	DS0	DS1	Enhanced Controller
Execution										Head Retracted to Track 0 Interrupt
SENSE INTERRUPT STATUS										
Command	W	0	0	0	0	1	0	0	0	Command Codes
Result	R				ST 0					Status Information at the End of Each Seek Operation
	R				PCN					
SPECIFY										
Command	W	0	0	0	0	0	0	1	1	Command Codes
	W	SRT						HUT		
	W	HLT							ND	
SENSE DRIVE STATUS										
Command	W	0	0	0	0	0	1	0	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
Result	R				ST 3					Status Information About FDD
DRIVE SPECIFICATION COMMAND										
Command	W	1	0	0	0	1	1	1	0	Command Codes
Phase	W	0	FD1	FD0	PTS	DRT1	DRT0	DT1	DT0	
	:	:	:	:	:	:	:	:	:	0–4 Bytes Issued
	W	DN	NRP	0	0	0	0	0	0	
Result	R	0	0	0	PTS	DRT1	DRT0	DT1	DT0	Drive 0
Phase	R	0	0	0	PTS	DRT1	DRT0	DT1	DT0	Drive 1
	R	0	0	0	0	0	0	0	0	RSVD
	R	0	0	0	0	0	0	0	0	RSVD
SEEK										
Command	W	0	0	0	0	1	1	1	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				NCN					
Execution										Head is Positioned Over Proper Cylinder on Diskette
CONFIGURE										
Command	W	0	0	0	1	0	0	1	1	Command Code
	W	0	0	0	0	0	0	0	0	
	W	0	EIS	EFIFO	POLL			FIFOTHR		
	W				PRETRK					
RELATIVE SEEK										
Command	W	1	DIR	0	0	1	1	1	1	
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				RCN					

2

Table 6-1. 82078 Command Set (Continued)

Phase	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
DUMPREG										
Command Execution	W	0	0	0	0	1	1	1	0	*Note Registers Placed in FIFO
Result	R				PCN-Drive 0					
	R				PCN-Drive 1					
	R				RSVD					
	R				RSVD					
	R			SRT				HUT		
	R				HLT				ND	
	R				SC/EOT					
	R	LOCK	0	0	0	D_1	D_0	GAP	WGATE	
	R	0	EIS	EFIFO	POLL		FIFOTHR			
	R				PRETRK					
READ ID										
Command	W	0	MFM	0	0	1	0	1	0	Commands
	W	0	0	0	0	0	HDS	DS1	DS0	
Execution										The First Correct ID Information on the Cylinder is Stored in Data Register
Result	R				ST 0					Status Information After Command Execution
	R				ST 1					
	R				ST 2					
	R				C					
	R				H					Disk Status After the Command has Completed
	R				R					
	R				N					
PERPENDICULAR MODE										
Command	W	0	0	0	1	0	0	1	0	Command Codes
	W	OW	0	0	0	D_1	D0	GAP	WGATE	
LOCK										
Command	W	LOCK	0	0	1	0	1	0	0	Command Codes
Result	R	0	0	0	LOCK	0	0	0	0	
PART ID										
Command	W	0	0	0	1	1	0	0	0	Command Code
Result	R	0	1	0		—STEPPING—			1	Part ID Number
POWERDOWN MODE										
Command	W	0	0	0	1	0	1	1	1	Command Code
	W	0	0	EREG EN	44PD EN	0	FDI TRI	MIN DLY	AUTO PD	
Result	R	0	0	EREG EN	44PD EN	0	FDI TRI	MIN DLY	AUTO PD	
OPTION										
Command	W	0	0	1	1	0	0	1	1	Command Code
	W				—RSVD—				ISO	

Table 6-1. 82078 Command Set (Continued)

Phase	R/W	DATA BUS								Remarks
		D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
SAVE										
Command Phase	W	0	0	1	0	1	1	1	0	Command Code
Result Phase	R	RSVD	SEL 3V#*	PD OSC	PC2	PC1	PC0	DRATE1	DRATE0	Save Info to Reprogram the FDC
	R	0	0	0	0	0	0	0	ISO	
	R				PCN-Drive 0					
	R				PCN-Drive 1					
	R				RSVD					
	R				RSVD					
	R		SRT					HUT		
	R				HLT				ND	
	R				SC/EOT					
	R	LOCK	0	0	0	D_1	D_0	GAP	WGATE	
	R	0	EIS	EFIFO	POLL			FIFOTHR		
	R				PRETRK					
	R	0	0	EREG EN	44PD EN	RSVD	FDI TRI	MIN DLY	AUTO PD	
	R				DISK/STATUS					
	R				RSVD					
RESTORE										
Command Phase	W	0	1	0	0	1	1	1	0	Command Code
	W	0	SEL 3V#*	0	PC2	PC1	PC0	DRATE1	DRATE0	Restore Original
	W	0	0	0	0	0	0	0	ISO	Register Status
	W				PCN-Drive 0					
	W				PCN-Drive 1					
	W				RSVD					
	W				RSVD					
	W		SRT					HUT		
	W				HLT				ND	
	W				SC/EOT					
	W	LOCK	0	0	0	D_1	D_0	GAP	WGATE	
	W	0	EIS	EFIFO	POLL			FIFOTHR		
	W				PRETRK					
	W	0	0	EREG EN	44PD EN	RSVD	FDI TRI	MIN DLY	AUTO PD	
	W				DISK/STATUS					
	W				RSVD					
	W				RSVD					

NOTE:
*For the 82078, 82078-5, SEL3V# = 1. For the 82078-3, SEL3V# = 0.

Table 6-1. 82078 Command Set (Continued)

Phase	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
FORMAT AND WRITE										
Command	W	1	MFM	1	0	1	1	0	1	Command Code
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				N					
	W				SC					
	W				GPL					
	W				D					
Execution	W				C					
Repeated	W				H					Input
for each	W				R					Sector
Sector	W				N					Parameters
	W	Data Transfer of N Bytes								82078 Formats and Writes Entire Track
Result	R				ST 0					
Phase	R				ST 1					
	R				ST 2					
	R				Undefined					
	R				Undefined					
	R				Undefined					
	R				Undefined					
INVALID										
Command	W				Invalid Codes					Invalid Command Codes (NoOp — 82078 Goes into Standby State)
Result	R				ST 0					ST 0 = 80H

Parameter Abbreviations

Symbol **Description**

44PD EN — Powerdown pin status. This bit allows the PD and IDLE pins to be available at FDS1 and FDME1 instead of the DS1 and ME1 pins. The BOOTSEL bit in the 44 pin part remaps the drive selects and motor enables when this bit is low. See the table below for functionality:

44PD EN	BOOTSEL(TDR)	Mapping
0	0	Default → DS0 → FDS0, ME0 → FDME0 DS1 → FDS1, ME1 → FDME1
0	1	DS0 → FDS1, ME0 → FDME1 DS1 → FDS0, ME1 → FDME0
1	X	DS0 → FDS0, ME0 → FDME0 DS1 → PD, ME1 → IDLE

AUTO PD — Auto powerdown control. If this bit is 0, then the automatic powerdown is disabled. If it is set to 1, then the automatic powerdown is enabled.

C — Cylinder address. The currently selected cylinder address, 0 to 255.

D0, D1 — Drive Select 0–3. Designates which drives are Perpendicular drives, a "1" indicating Perpendicular drive.

D — Data pattern. The pattern to be written in each sector data field during formatting.

DN — Done. This bit indicates that this is the last byte of the drive specification command. The 82078 checks to see if this bit is high or low. If it is low, it expects more bytes.

DN = 0 82078 expects more subsequent bytes.

DN = 1 Terminates the command phase and jumps to the results phase. An additional benefit is that by setting this bit high, a direct check of the current drive specifications can be done.

DIR — Direction control. If this bit is 0, then the head will step out from the spindle during a relative seek. If set to a 1, the head will step in toward the spindle.

DS0, DS1 — Disk Drive Select.

DS1	DS0	
0	0	drive 0
0	1	drive 1
1	0	RSVD
1	1	RSVD

DTL — Special sector size. By setting N to zero (00), DTL may be used to control the number of bytes transferred in disk read/write commands. The sector size (N = 0) is set to 128. If the actual sector (on the diskette) is larger than DTL, the remainder of the actual sector is read but is not passed to the host during read commands; during write commands, the remainder of the actual sector is written with all zero bytes. The CRC check code is calculated with the actual sector. When N is not zero, DTL has no meaning and should be set to FF HEX.

DRATE[0:1] — Data rate values from the DSR register.

DRT0, DRT1 Data rate table select. These two bits select between the different data rate tables. The default is the conventional table. These also provide mapping of the data rates selected in the DSR and CCR. The table below shows this.

Bits in DSR/CCR					
DRT0	**DRT1**	**DRATE1**	**DRATE0**	**Data Rate**	**Operation**
0	0	1	1	1 Mbps	Default
		0	0	500 Kbps	
		0	1	300 Kbps	
		1	0	250 Kbps	
0	1	RSVD	RSVD	RSVD	RSVD
1	0	RSVD	RSVD	RSVD	RSVD
1	1	1	1	1 Mbps	Perpendicular mode FDDs
		0	0	500 Kbps	
		0	1	RSVD	
		1	0	250 Kbps	

DT0, DT1 Drive density select type. These bits select the outputs on DRVDEN0 and DRVDEN1 based on mode of operation that was selected via the IDENT1 and IDENT0 pins. More information is available in the Design Applications section.

EC Enable Count. When this bit is "1" the "DTL" parameter of the Verify Command becomes SC (Number of sectors per track).

EFIFO Enable FIFO. When this bit is 0, the FIFO is enabled. A "1" puts the 82078 in the 8272A compatible mode where the FIFO is disabled.

EIS Enable implied seek. When set, a seek operation will be performed before executing any read or write command that requires the C parameter in the command phase. A "0" disables the implied seek.

EOT End of track. The final sector number of the current track.

EREG EN Enhanced Register Enable.

EREG EN = 1 The TDR register is extended and SRB is made visible to the user.

EREG EN = 0 Standard registers are used.

FDI TRI Floppy Drive Interface Tristate: If this bit is 0, then the output pins of the floppy disk drive interface are tri-stated. This is also the default state. If it is set to 1, then the floppy disk drive interface remains unchanged.

FD0, FD1 Floppy drive select. These two bits select which physical drive is being specified. The FDn corresponds to FDSn and FDMEn on the floppy drive interface. The drive is selected independent of the BOOTSELn bits. Please refer to Section 2.1.1 which explains the distinction between physical drives and their virtual mapping as defined by the BOOTSEL1 and BOOTSEL0 bits.

FD1	**FD0**	**Drive Slot**
0	0	drive 0
0	1	drive 1
1	0	RSVD
1	1	RSVD

GAP Alters Gap2 length when using Perpendicular Mode.

GPL Gap length. The Gap3 size. (Gap3 is the space between sectors excluding the VCO synchronization field).

HDS Head address. Selected head: 0 or 1 (disk side 0 or 1) as encoded in the sector ID field.

HLT Head load time. The time interval that 82078 waits after loading the head and before initiating a read or write operation. Refer to the SPECIFY command for actual delays.

HUT Head unload time. The time interval from the end of the execution phase (of a read or write command) until the head is unloaded. Refer to the SPECIFY command for actual delays.

ISO ISO Format: If this bit is set high the ISO format is used for all data transfer commands. When this bit is set low the normal IBM system 34 and perpendicular is used. The default is ISO = 0.

Lock — Lock defines whether EFIFO, FIFOTHR, and PRETRK parameters of the CONFIGURE command can be reset to their default values by a "Software Reset" (Reset made by setting the proper bit in the DSR or DOR registers).

MFM — MFM mode. A one selects the double density (MFM) mode. A zero is reserved.

MIN DLY — Minimum power up time control. This bit is active only if AUTO PD bit is enabled. Setting this bit to a 0, assigns a 10 ms minimum power up time and setting this bit to a 1, assigns a 0.5s minimum power up time.

MT — Multi-track selector. When set, this flag selects the multi-track operating mode. In this mode, the 82078 treats a complete cylinder, under head 0 and 1, as a single track. The 82078 operates as if this expanded track started at the first sector under head 0 and ended at the last sector under head 1. With this flag set, a multitrack read or write operation will automatically continue to the first sector under head 1 when the 82078 finishes operating on the last sector under head 0.

N — Sector size code. This specifies the number of bytes in a sector. If this parameter is "00", then the sector size is 128 bytes. The number of bytes transferred is determined by the DTL parameter. Otherwise the sector size is (2 raised to the "N'th" power) times 128. All values up to "07" hex are allowable. "07"h would equal a sector size of 16K. It is the users responsibility to not select combinations that are not possible with the drive.

N	Sector Size
00	128 bytes
01	256 bytes
02	512 bytes
03	1024
..	...
07	16 Kbytes

NCN — New cylinder number. The desired cylinder number.

ND — Non-DMA mode flag. When set to 1, indicates that the 82078 is to operate in the non-DMA mode. In this mode, the host is interrupted for each data transfer. When set to 0, the 82078 operates in DMA mode, interfacing to a DMA controller by means of the DRQ and DACK# signals.

NRP — No Results phase. When this bit is set high the result phase is skipped. When this bit is low the result phase will be generated.

OW — The bits denoted D0, D1, D2, and D3 of the PERPENDICULAR MODE command can only be overwritten when the OW bit is set to "1".

PCN — Present cylinder number. The current position of the head at the completion of SENSE INTERRUPT STATUS command.

PC2, PC1, PC0 — Precompensation values from the DSR register.

PDOSC — When this bit is set, the internal oscillator is turned off.

PTS — Precompensation table select. This bit selects whether to enable the precompensation value programmed in the DSR or not. In the default state, the value programmed in DSR will be used.

PTS = 0 DSR programmed precompensation delays

PTS = 1 No precompensation delay is selected for the corresponding drive.

POLL — Polling disable. When set, the internal polling routine is disabled. When clear, polling is enabled.

PRETRK — Precompensation start track number. Programmable from track 00 to FFH.

R — Sector address. The sector number to be read or written. In multi-sector transfers, this parameter specifies the sector number of the first sector to be read or written.

RCN — Relative cylinder number. Relative cylinder offset from present cylinder as used by the RELATIVE SEEK command.

SC — Number of sectors. The number of sectors to be initialized by the FORMAT command. The number of sectors to be verified during a Verify Command, when EC is set.

SK — Skip flag. When set to 1, sectors containing a deleted data address mark will automatically be skipped during the execution of READ DATA. If READ DELETED is executed, only sectors with a deleted address mark will be accessed. When set to "0", the sector is read or written the same as the read and write commands.

SRT — Step rate interval. The time interval between step pulses issued by the 82078. Programmable from 0.5 ms to 8 ms, in increments of 0.5 ms at the 1 Mbit data rate. Refer to the SPECIFY command for actual delays.

ST0–3 — Status registers 0–3. Registers within the 82078 that store status information after a command has been executed. This status information is available to the host during the result phase after command execution.

STEPPING — These bits identify the stepping of the 82078.

WGATE — Write gate alters timing of WE, to allow for pre-erase loads in perpendicular drives.

6.1 Data Transfer Commands

All of the READ DATA, WRITE DATA and VERIFY type commands use the same parameter bytes and return the same results information. The only difference being the coding of bits 0–4 in the first byte.

An implied seek will be executed if the feature was enabled by the CONFIGURE command. This seek is completely transparent to the user. The Drive Busy bit for the drive will go active in the Main Status Register during the seek portion of the command. If the seek portion fails, it will be reflected in the results status normally returned for a READ/WRITE DATA command. Status Register 0 (ST0) would contain the error code and C would contain the cylinder on which the seek failed.

6.1.1 READ DATA

A set of nine (9) bytes is required to place the 82078 into the Read Data Mode. After the READ DATA command has been issued, the 82078 loads the head (if it is in the unloaded state), waits the specified head settling time (defined in the SPECIFY command), and begins reading ID Address Marks and ID fields. When the sector address read off the diskette matches with the sector address specified in the command, the 82078 reads the sector's data field and transfers the data to the FIFO.

After completion of the read operation from the current sector, the sector address is incremented by one, and the data from the next logical sector is read and output via the FIFO. This continuous read function is called "Multi-Sector Read Operation". Upon receipt of TC, or an implied TC (FIFO overrun/underrun), the 82078 stops sending data, but will continue to read data from the current sector, check the CRC bytes, and at the end of the sector terminate the READ DATA Command.

N determines the number of bytes per sector (see Table 6-2). If N is set to zero, the sector size is set to 128. The DTL value determines the number of bytes to be transferred. If DTL is less than 128, the 82078 transfers the specified number of bytes to the host. For reads, it continues to read the entire 128 byte sector and checks for CRC errors. For writes it completes the 128 byte sector by filling in zeroes. If N is not set to 00 Hex, DTL should be set to FF Hex, and has no impact on the number of bytes transferred.

Table 6-2. Sector Sizes

N	Sector Size
00	128 bytes
01	256 bytes
02	512 bytes
03	1024 bytes
—	—
07	16 Kbytes

The amount of data which can be handled with a single command to the 82078 depends upon MT (multi-track) and N (Number of bytes/sector).

Table 6-3. Effects of MT and N Bits

MT	N	Max. Transfer Capacity	Final Sector Read from Disk
0	1	256 × 26 = 6656	26 at side 0 or 1
1	1	256 × 52 = 13312	26 at side 1
0	2	512 × 15 = 7680	15 at side 0 or 1
1	2	512 × 30 = 15360	15 at side 1
0	3	1024 × 8 = 8192	8 at side 0 or 1
1	3	1024 × 16 = 16384	16 at side 1

The Multi-Track function (MT) allows the 82078 to read data from both sides of the diskette. For a particular cylinder, data will be transferred starting at Sector 1, Side 0 and completing at the last sector of the same track at Side 1.

If the host terminates a read or write operation in the 82078, then the ID information in the result phase is dependent upon the state of the MT bit and EOT byte. Refer to Table 6-6. The termination must be normal.

At the completion of the READ DATA Command, the head is not unloaded until after the Head Unload Time Interval (specified in the SPECIFY command) has elapsed. If the host issues another command before the head unloads then the head settling time may be saved between subsequent reads.

If the 82078 detects a pulse on the INDX# pin twice without finding the specified sector (meaning that the diskette's index hole passes through index detect logic in the drive twice), the 82078 sets the IC code in Status Register 0 to "01" (Abnormal termination), and sets the ND bit in Status Register 1 to "1" indicating a sector not found, and terminates the READ DATA Command.

After reading the ID and Data Fields in each sector, the 82078 checks the CRC bytes. If a CRC error occurs in the ID or data field, the 82078 sets the IC code in Status Register 0 to "01" (Abnormal termination), sets the DE bit flag in Status Register 1 to "1", sets the DD bit in Status Register 2 to "1" if CRC is incorrect in the ID field, and terminates the READ DATA Command.

Table 6-4 below describes the affect of the SK bit on the READ DATA command execution and results.

Table 6-4. Skip Bit vs READ DATA Command

SK Bit Value	Data Address Mark Type Encountered	Results		
		Sector Read?	CM Bit of ST2 Set?	Description of Results
0	Normal Data	Yes	No	Normal Termination.
0	Deleted Data	Yes	Yes	Address Not Incremented. Next Sector Not Searched For.
1	Normal Data	Yes	No	Normal Termination.
1	Deleted Data	No	Yes	Normal Termination Sector Not Read ("Skipped").

Except where noted in Table 6-4, the C or R value of the sector address is automatically incremented (see Table 6-6).

6.1.2 READ DELETED DATA

This command is the same as the READ DATA command, only it operates on sectors that contain a Deleted Data Address Mark at the beginning of a Data Field.

Table 6-5 describes the affect of the SK bit on the READ DELETED DATA command execution and results.

Table 6-5. Skip Bit vs READ DELETED DATA Command

SK Bit Value	Data Address Mark Type Encountered	Results		
		Sector Read?	CM Bit of ST2 Set?	Description of Results
0	Normal Data	Yes	Yes	Normal Termination.
0	Deleted Data	Yes	No	Address Not Incremented. Next Sector Not Searched For.
1	Normal Data	No	Yes	Normal Termination Sector Not Read ("Skipped").
1	Deleted Data	Yes	No	Normal Termination.

Except where noted in Table 6-5 above, the C or R value of the sector address is automatically incremented (see Table 6-6).

6.1.3 READ TRACK

This command is similar to the READ DATA command except that the entire data field is read continuously from each of the sectors of a track. Immediately after encountering a pulse on the INDX# pin, the 82078 starts to read all data fields on the track as continuous blocks of data without regard to logical sector numbers. If the 82078 finds an error in the ID or DATA CRC check bytes, it continues to read data from the track and sets the appropriate error bits at the end of the command. The 82078 compares the ID information read from each sector with the specified value in the command, and sets the ND flag of Status Register 1 to a "1" if there is no comparison.

Multi-track or skip operations are not allowed with this command. The MT and SK bits (Bits D7 and D5 of the first command byte respectively) should always be set to "0".

This command terminates when the EOT specified number of sectors have been read. If the 82078 does not find an ID Address Mark on the diskette after the second occurrence of a pulse on the INDX# pin, then it sets the IC code in Status Register 0 to "01" (Abnormal termination), sets the MA bit in Status Register 1 to "1", and terminates the command.

Table 6-6. Result Phase Table

MT	Head	Final Sector Transferred to Host	ID Information at Result Phase			
			C	H	R	N
0	0	Less than EOT	NC	NC	R+1	NC
		Equal to EOT	C+1	NC	01	NC
	1	Less than EOT	NC	NC	R+1	NC
		Equal to EOT	C+1	NC	01	NC
1	0	Less than EOT	NC	NC	R+1	NC
		Equal to EOT	NC	LSB	01	NC
	1	Less than EOT	NC	NC	R+1	NC
		Equal to EOT	C+1	LSB	01	NC

NOTES:
NC: No Change, the same value as the one at the beginning of command execution.
LSB: Least Significant Bit, the LSB of H is complemented.

6.1.4 WRITE DATA

After the WRITE DATA command has been issued, the 82078 loads the head (if it is in the unloaded state), waits the specified head load time if unloaded (defined in the SPECIFY command), and begins reading ID Fields. When the sector address read from the diskette matches the sector address specified in the command, the 82078 reads the data from the host via the FIFO, and writes it to the sector's data field.

After writing data into the current sector, the 82078 computes the CRC value and writes it into the CRC field at the end of the sector transfer. The Sector Number stored in "R" is incremented by one, and the 82078 continues writing to the next data field. The 82078 continues this "Multi-Sector Write Operation". Upon receipt of a terminal count signal or if a FIFO over/under run occurs while a data field is being written, then the remainder of the data field is filled with zeros.

The 82078 reads the ID field of each sector and checks the CRC bytes. If it detects a CRC error in one of the ID Fields, it sets the IC code in Status Register 0 to "01" (Abnormal termination), sets the DE bit of Status Register 1 to "1", and terminates the WRITE DATA command.

The WRITE DATA command operates in much the same manner as the READ DATA command. The following items are the same. Please refer to the READ DATA Command for details:

- Transfer Capacity
- EN (End of Cylinder) bit
- ND (No Data) bit
- Head Load, Unload Time Interval
- ID information when the host terminates the command.
- Definition of DTL when N = 0 and when N does not = 0.

6.1.5 WRITE DELETED DATA

This command is almost the same as the WRITE DATA command except that a Deleted Data Address Mark is written at the beginning of the Data Field instead of the normal Data Address Mark. This command is typically used to mark a bad sector containing an error on the floppy disk.

6.1.6 VERIFY

The VERIFY command is used to verify the data stored on a disk. This command acts exactly like a READ DATA command except that no data is transferred to the host. Data is read from the disk, CRC computed and checked against the previously stored value.

Because no data is transferred to the host, TC (pin-25) cannot be used to terminate this command. By setting the EC bit to "1" an implicit TC will be issued to the 82078. This implicit TC will occur when the SC value has decrement to 0 (an SC value of 0 will verify 256 sectors). This command can also be terminated by setting the EC bit to "0" and the EOT value equal to the final sector to be checked. If EC is set to "0" DTL/SC should be programmed to 0FFH. Refer to Table 6-6 and Table 6-7 for information concerning the values of MT and EC versus SC and EOT value.

Definitions:

Sectors Per Side = Number of formatted sectors per each side of the disk.

Sectors Remaining = Number of formatted sectors left which can be read, including side 1 of the disk if MT is set to "1".

Table 6-7. Verify Command Result Phase Table

MT	EC	SC/EOT Value	Termination Result
0	0	SC = DTL EOT ≤ # Sectors Per Side	Successful Termination Result Phase Valid
0	0	SC = DTL EOT > # Sectors Per Side	Unsuccessful Termination Result Phase Invalid
0	1	SC ≤ # Sectors Remaining AND EOT ≤ # Sectors Per Side	Successful Termination Result Phase Valid
0	1	SC > # Sectors Remaining OR EOT > # Sectors Per Side	Unsuccessful Termination Result Phase Invalid
1	0	SC = DTL EOT ≤ # Sectors Per Side	Successful Termination Result Phase Valid
1	0	SC = DTL EOT > # Sectors Per Side	Unsuccessful Termination Result Phase Invalid
1	1	SC ≤ # Sectors Remaining AND EOT ≤ # Sectors Per Side	Successful Termination Result Phase Valid
1	1	SC > # Sectors Remaining OR EOT > # Sectors Per Side	Unsuccessful Termination Result Phase Invalid

NOTE:
If MT is set to "1" and the SC value is greater than the number of remaining formatted sectors on Side 0, verifying will continue on Side 1 of the disk.

6.1.7 FORMAT TRACK

The FORMAT command allows an entire track to be formatted. After a pulse from the INDX# pin is detected, the 82078 starts writing data on the disk including Gaps, Address Marks, ID Fields and Data Fields, per the IBM System 34 (MFM). The particular values that will be written to the gap and data field are controlled by the values programmed into N, SC, GPL, and D which are specified by the host during the command phase. The data field of the sector is filled with the data byte specified by D. The ID Field for each sector is supplied by the host; that is, four data bytes per sector are needed by the 82078 for C, H, R, and N (cylinder, head, sector number and sector size respectively).

After formatting each sector, the host must send new values for C, H, R and N to the 82078 for the next sector on the track. The R value (sector number) is the only value that must be changed by the host after each sector is formatted. This allows the disk to be formatted with nonsequential sector addresses (interleaving). This incrementing and formatting continues for the whole track until the 82078 encounters a pulse on the INDX# pin again and it terminates the command.

Table 6-8 contains typical values for gap fields which are dependent upon the size of the sector and the number of sectors on each track. Actual values can vary due to drive electronics.

Table 6-8. Typical PC-AT Values for Formatting

Drive Form	MEDIA	Sector Size	N	SC	GPL1	GPL2
5.25″	1.2M	512	02	0F	2A	50
	360K	512	02	09	2A	50
3.5″	2.88M	512	02	24	38	53
	1.44M	512	02	18	1B	54
	720K	512	02	09	1B	54

NOTE:
All values except Sector Size are in Hex.

Gap3 is programmable during reads, writes, and formats.

GPL1 = suggested Gap3 values in read and write commands to avoid splice point between data field and ID field of contiguous sections.

GPL2 = suggested Gap3 value in FORMAT TRACK command.

6.1.7.1 Format Fields

Table 6-9. System 34 Format Double Density

GAP 4a 80x 4E	SYNC 12x 00	IAM 3x C2	IAM FC	GAP1 50x 4E	SYNC 12x 00	IDAM 3x A1	IDAM FE	CYL	HD	SEC	NO	CRC	GAP2 22x 4E	SYNC 12x 00	DATA AM 3x A1	DATA AM FB F8	DATA	CRC	GAP3	GAP 4b

Table 6-10. ISO Format

GAP1 32x 4E	SYNC 12x 00	IDAM 3x A1	IDAM FE	CYL	HD	SEC	NO	CRC	GAP2 22x 4E	SYNC 12x 00	DATA AM 3x A1	DATA AM FB F8	DATA	CRC	GAP3	GAP 4b

Table 6-11. Perpendicular Format

GAP 4a 80x 4E	SYNC 12x 00	IAM 3x C2	IAM FC	GAP1 50x 4E	SYNC 12x 00	IDAM 3x A1	IDAM FE	CYL	HD	SEC	NO	CRC	GAP2 41x 4E	SYNC 12x 00	DATA AM 3x A1	DATA AM FB F8	DATA	CRC	GAP3	GAP 4b

6.2 Scan Commands

The SCAN Commands allow data which is being read from the diskette to be compared against data which is being supplied from the main system (Processor in NON-DMA mode, and DMA Controller in DMA mode). The FDC compares the data on a byte-by-byte basis, and looks for a sector of data which meets the conditions of $D_{FDD} = D_{Processor}$, $D_{FDD} \leq D_{Processor}$, or $D_{FDD} \geq D_{Processor}$. Ones complement arithmetic is used for comparison (FF = largest number, 00 = smallest number). After a whole sector of data is compared, if the conditions are not met, the sector number is incremented (R + STP → R), and the scan operation is continued. The scan operation continues until one of the following conditions occur, the conditions for scan are met (equal, low, or high), the last sector on the track is reached (EOT), or the terminal count signal is received.

Table 6-12. Scan Status Codes

Command	Status Register 2		Comments
	Bit 2 = SN	Bit 3 = SH	
Scan Equal	0	1	$D_{FDD} = D_{Processor}$
	1	0	$D_{FDD} \neq D_{Processor}$
Scan Low or Equal	0	1	$D_{FDD} = D_{Processor}$
	0	0	$D_{FDD} < D_{Processor}$
	1	0	$D_{FDD} \not> D_{Processor}$
Scan High or Equal	0	1	$D_{FDD} = D_{Processor}$
	0	0	$D_{FDD} > D_{Processor}$
	1	0	$D_{FDD} \not< D_{Proceeeor}$

If the conditions for scan are met then the FDC sets the SH (Scan Hit) flag of Status Register 2 to a 1 (high), and terminates the Scan Command, If the conditions for scan are not met between the starting sector (as specified by R) and the last sector on the cylinder (EOT), then the FDC sets the SN (Scan Not Satisfied) flag of Status Register 2 to a 1 (high), and terminates the Scan Command. The receipt of a TERMINAL COUNT signal from the Processor or DMA Controller during the scan operation will cause the FDC to complete the comparison of the particular byte which is in process, and then to terminate the command. Table 6-12 shows the status of bits SH and SN under various conditions of SCAN.

If the FDC encounters a Deleted Data Address Mark on one of the sectors (and SK = 0), then it regards the sector as the last sector on the cylinder, sets CM (Control Mark) flag of Status Register 2 to a 1 (high) and terminates the command. If SK = 1, the FDC skips the sector with the Deleted Address Mark, and reads the next sector. In the second case (SK = 1), the FDC sets the CM (Control Mark) flag of Status Register 2 to a 1 (high) in order to show that a Deleted Sector had been encountered.

When either the STP (contiguous sectors STP = 01, or alternate sectors STP = 02 sectors are read) or the MT (Multi-Track) are programmed, it is necessary to remember that the last sector on the track must be read. For example, if STP = 02, MT = 0, the sectors are numbered sequentially 1 through 26, and we start the Scan Command at sector 21; the following will happen. Sectors 21, 23, and 25 will be read, then the next sector (26) will be skipped and the index Hole will be encountered before the EOT value of 26 can be read. This will result in an abnormal termination of the command. If the EOT had been set at 25 or the scanning started at sector 20, then the Scan Command would be completed in a normal manner.

During the Scan Command data is supplied by either the processor or DMA Controller for comparison against the data read from the diskette. In order to avoid having the OR (Over Run) flag set in Status Register 1, it is necessary to have the data available in less than 13 μs. If an Overrun occurs the FDC terminates the command.

6.3 Control Commands

Control commands differ from the other commands in that no data transfer takes place. Three commands generate an interrupt when complete; READ ID, RECALIBRATE and SEEK. The other control commands do not generate an interrupt.

6.3.1 READ ID

The READ ID command is used to find the present position of the recording heads. The 82078 stores the values from the first ID Field it is able to read into its registers. If the 82078 does not find an ID Address Mark on the diskette after the second occurrence of a pulse on the INDX# pin, it then sets the IC code in Status Register 0 to "01" (Abnormal termination), sets the MA bit in Status Register 1 to "1", and terminates the command.

The following commands will generate an interrupt upon completion. They do not return any result bytes. It is highly recommended that control commands be followed by the SENSE INTERRUPT STATUS command. Otherwise, valuable interrupt status information will be lost.

6.3.2 RECALIBRATE

This command causes the read/write head within the 82078 to retract to the track 0 position. The 82078 clears the contents of the PCN counter, and checks the status of the TRK0 pin from the FDD. As long as the TRK0 pin is low, the DIR pin remains 0 and step pulses are issued. When the TRK0 pin goes high, the SE bit in Status Register 0 is set to "1", and the command is terminated. If the TRK0 pin is still low after 79 step pulses the command is terminated. Disks capable of handling more than 80 tracks per side may require more than one RECALIBRATE command to return the head back to physical Track 0.

The RECALIBRATE command does not have a result phase. SENSE INTERRUPT STATUS command must be issued after the RECALIBRATE command to effectively terminate it and to provide verification of the head position (PCN). During the command phase of the recalibrate operation, the 82078 is in the BUSY state, but during the execution phase it is in a NON BUSY state. At this time another RECALIBRATE command may be issued, and in this manner, parallel RECALIBRATE operations may be done on up to 2 drives at once.

Upon power up, the software must issue a RECALIBRATE command to properly initialize all drives and the controller.

6.3.3 DRIVE SPECIFICATION COMMAND

The 82078 uses two pins, DRVDEN0 and DRVDEN1 to select the density for modern drives. These signals inform the drive of the type of diskette in the drive. The Drive Specification command specifies the polarity of the DRVDEN0 and DRVDEN1 pins. It also enables or disables DSR programmed precompensation.

This command removes the need for a hardware workaround to accommodate differing specifications among drives. By programming this command during BIOS's POST routine, the floppy disk controller will internally configure the correct values for DRVDEN0 and DRVDEN1 with corresponding precompensation value and data rate table enabled for the particular type of drive.

This command is protected from software resets. After executing the DRIVE SPEC command, subsequent software resets will not clear the programmed parameters. Only another DRIVE SPEC command or H/W reset can reset it to default values. The 6 LSBs of the last byte of this command are reserved for future use.

The DRATE0 and DRATE1 are values as programmed in the DSR register. The DENSEL is high for high data rates (1 Mbps and 500 Kbps) and low for low data rates (300 Kbps and 250 Kbps).

The following table describes the drives that are supported with the DT0, DT1 bits of the Drive Specification command:

DRVDENn Polarities

DT0	DT1	Data Rate	DRVDEN0	DRVDEN1
0*	0*	1 Mbps	1	1
		500 Kbps	1	0
		300 Kbps	0	1
		250 Kbps	0	0
0	1	1 Mbps	1	1
		500 Kbps	0	0
		300 Kbps	0	1
		250 Kbps	1	0
1	0	1 Mbps	0	1
		500 Kbps	0	0
		300 Kbps	1	1
		250 Kbps	1	0
1	1	1 Mbps	1	1
		500 Kbps	0	0
		300 Kbps	1	0
		250 Kbps	0	1

NOTE:
(*) Denotes the default setting.

6.3.4 SEEK

The read/write head within the drive is moved from track to track under the control of the SEEK command. The 82078 compares the PCN which is the current head position with the NCN and performs the following operation if there is a difference:

PCN < NCN: Direction signal to drive set to "1" (step in), and issues step pulses.

PCN > NCN: Direction signal to drive set to "0" (step out), and issues step pulses.

The rate at which step pulses are issued is controlled by SRT (Stepping Rate Time) in the SPECIFY command. After each step pulse is issued, NCN is compared against PCN, and when NCN = PCN, then the SE bit in Status Register 0 is set to "1", and the command is terminated.

During the command phase of the seek or recalibrate operation, the 82078 is in the BUSY state, but during the execution phase it is in the NON BUSY state.

Note that if implied seek is not enabled, the read and write commands should be preceded by:

1. SEEK command; Step to the proper track
2. SENSE INTERRUPT STATUS command; Terminate the Seek command
3. READ ID. Verify head is on proper track
4. Issue READ/WRITE command.

The SEEK command does not have a result phase. Therefore, it is highly recommended that the SENSE INTERRUPT STATUS command be issued after the SEEK command to terminate it and to provide verification of the head position (PCN). The H bit (Head Address) in ST0 will always return a "0". When exiting DSR POWERDOWN mode, the 82078 clears the PCN value and the status information to zero. Prior to issuing the DSR POWERDOWN command, it is highly recommended that the user service all pending interrupts through the SENSE INTERRUPT STATUS command.

6.3.5 SENSE INTERRUPT STATUS

An interrupt signal on INT pin is generated by the 82078 for one of the following reasons:

1. Upon entering the Result Phase of:
   a. READ DATA Command
   b. READ TRACK Command
   c. READ ID Command
   d. READ DELETED DATA Command
   e. WRITE DATA Command

f. FORMAT TRACK Command
g. WRITE DELETED DATA Command
h. VERIFY Command

2. End of SEEK, RELATIVE SEEK or RECALIBRATE Command
3. 82078 requires a data transfer during the execution phase in the non-DMA Mode

The SENSE INTERRUPT STATUS command resets the interrupt signal and via the IC code and SE bit of Status Register 0, identifies the cause of the interrupt. If a SENSE INTERRUPT STATUS command is issued when no active interrupt condition is present, the status register ST0 will return a value of 80H (invalid command).

Table 6-13. Interrupt Indentification

SE	IC	Interrupt Due To
0	11	Polling
1	00	Normal Termination of SEEK or RECALIBRATE command
1	01	Abnormal Termination of SEEK or RECALIBRATE command

The SEEK, RELATIVE SEEK and the RECALIBRATE commands have no result phase. SENSE INTERRUPT STATUS command must be issued immediately after these commands to terminate them and to provide verification of the head position (PCN). The H (Head Address) bit in ST0 will always return a "0". If a SENSE INTERRUPT STATUS is not issued, the drive, will continue to be BUSY and may effect the operation of the next command.

6.3.6 SENSE DRIVE STATUS

SENSE DRIVE STATUS obtains drive status information. It has no execution phase and goes directly to the result phase from the command phase. STATUS REGISTER 3 contains the drive status information.

6.3.7 SPECIFY

The SPECIFY command sets the initial values for each of the three internal timers. The HUT (Head Unload Time) defines the time from the end of the execution phase of one of the read/write commands to the head unload state. The SRT (Step Rate Time) defines the time interval between adjacent step pulses. Note that the spacing between the first and second step pulses may be shorter than the remaining step pulses. The HLT (Head Load Time) defines the time between the command phase to the execution phase of a read/write data command. The Head Unload Time (HUT) timer starts at the end of the execution phase to the beginning of the result phase of a read/write command. The values change with the data rate speed selection and are documented in Table 6-14.

Table 6-14. Drive Control Delays (ms)

	HUT				SRT			
	1M	500K	300K	250K	1M	500K	300K	250K
0	128	256	426	512	8.0	16	26.7	32
1	8	16	26.7	32	7.5	15	25	30
—	—	—	—	—	—	—	—	—
A	80	160	267	320	3.0	6.0	10.2	12
B	88	176	294	352	2.5	5.0	8.35	10
C	96	192	320	384	2.0	4.0	6.68	8
D	104	208	346	416	1.5	3.0	5.01	6
E	112	224	373	448	1.0	2.0	3.33	4
F	120	240	400	480	0.5	1.0	1.67	2

Table 6-15. Head Load Time (ms)

	HLT			
	1M	500K	300K	250K
00	128	256	426	512
01	1	2	3.3	4
02	2	4	6.7	8
—	—	—	—	—
7E	126	252	420	504
7F	127	254	423	508

The choice of DMA or NON-DMA operations is made by the ND bit. When this bit is "1", the NON-DMA mode is selected, and when ND is "0", the DMA mode is selected. In DMA mode, data transfers are signalled by the DRQ pin. Non-DMA mode uses the RQM bit and the INT pin to signal data transfers.

6.3.8 CONFIGURE

Issue the configure command to enable features like the programmable FIFO and set the beginning track for pre-compensation. A CONFIGURE command need not be issued if the default values of the 82078 meet the system requirements.

CONFIGURE DEFAULT VALUES:

EIS No Implied Seeks
EFIFO FIFO Disabled
POLL Polling Enabled
FIFOTHR FIFO Threshold Set to 1 Byte
PRETRK Pre-Compensation Set to Track 0

EIS—Enable Implied Seek. When set to "1", the 82078 will perform a SEEK operation before executing a read or write command. Defaults to no implied seek.

EFIFO—A "1" puts the FIFO into the 8272A compatible mode where the FIFO is disabled. This means data transfers are asked for on a byte by byte basis. Defaults to "1", FIFO disabled. The threshold defaults to one.

POLL—Disable polling of the drives. Defaults to "0", polling enabled. When enabled, a single interrupt is generated after a RESET. No polling is performed while the drive head is loaded and the head unload delay has not expired.

FIFOTHR—The FIFO threshold in the execution phase of read or write commands. This is programmable from 1 to 16 bytes. Defaults to one byte. A "00" selects one byte, "0F" selects 16 bytes.

PRETRK—Pre-compensation start track number. Programmable from track 0 to 255. Defaults to track 0. A "00" selects track 0, "FF" selects 255.

6.3.9 VERSION

The VERSION command checks to see if the controller is an enhanced type (82077, 82077AA, 82077SL) or the older type (8272A/765A). A value of 90H is returned as the result byte, defining an enhanced FDD controller is in use. No interrupts are generated. Refer to the Part ID command for more identification information on the 82078.

6.3.10 RELATIVE SEEK

The command is coded the same as for SEEK, except for the MSB of the first byte and the DIR bit.

DIR Head Step Direction Control

DIR	Action
0	Step Head Out
1	Step Head In

RCN Relative Cylinder Number that determines how many tracks to step the head in or out from the current track number.

The RELATIVE SEEK command differs from the SEEK command in that it steps the head the absolute number of tracks specified in the command instead of making a comparison against an internal register. The SEEK command is good for drives that support a maximum of 256 tracks. RELATIVE SEEKs cannot be overlapped with other RELATIVE SEEKs. Only one RELATIVE SEEK can be active at a time. Bit 4 of Status Register 0 (EC) will be set if RELATIVE SEEK attempts to step outward beyond Track 0.

As an example, assume that a floppy drive has 300 useable tracks and that the host needs to read track 300 and the head is on any track (0—255). If a SEEK command was issued, the head would stop at track 255. If a RELATIVE SEEK command was issued, the 82078 would move the head the specified number of tracks, regardless of the internal cylinder position register (but would increment the register). If the head had been on track 40 (D), the maximum track that the 82078 could position the head on using RELATIVE SEEK, would be 296 (D), the initial track, +256 (D). The maximum count that the head can be moved with a single RELATIVE SEEK command is 256 (D).

The internal register, PCN, would overflow as the cylinder number crossed track 255 and would contain 40 (D). The resulting PCN value is thus (NCN + PCN) mod 256. Functionally, the 82078 starts counting from 0 again as the track number goes above 255(D). It is the users responsibility to compensate 82078 functions (precompensation track number) when accessing tracks greater than 255. The 82078 does not keep track that it is working in an "extended track area" (greater than 255). Any command issued would use the current PCN value except for the RECALIBRATE command which only looks for the TRACK0 signal. RECALIBRATE would return an error if the head was farther than 79 due to its limitation of issuing a maximum 80 step pulses. The user simply needs to issue a second RECALIBRATE command. The SEEK command and implied seeks will function correctly within the 44 (D) track (299–255) area of the "extended track area". It is the users responsibility not to issue a new track position that would exceed the maximum track that is present in the extended area.

To return to the standard floppy range (0-255) of tracks, a RELATIVE SEEK would be issued to cross the track 255 boundary.

A RELATIVE SEEK can be used instead of the normal SEEK but the host is required to calculate the difference between the current head location and the new (target) head location. This may require the host to issue a READ ID command to ensure that the head is physically on the track that software assumes it to be. Different 82078 commands will return different cylinder results which may be difficult to keep track of with software without the READ ID command.

6.3.11 DUMPREG

The DUMPREG command is designed to support system run-time diagnostics and application software development and debug. The command returns pertinent information regarding the internal status of the 82078. This can be used to verify the values initialized in the 82078.

6.3.12 PERPENDICULAR MODE COMMAND

Note, perpendicular mode functionality is not available on the 82078-5.

6.3.12.1 About Perpendicular Recording Mode

An added capability of the 82078 is the ability to interface directly to perpendicular recording floppy drives. Perpendicular recording differs from the traditional longitudinal method by orienting the magnetic bits vertically. This scheme packs in more data bits for the same area.

6.3.12.2 The Perpendicular Mode Command

The PERPENDICULAR MODE command allows the system designers to designate specific drives as Perpendicular recording drives. Data transfers between Conventional and Perpendicular drives are allowed without having to issue PERPENDICULAR MODE commands between the accesses of the two different drives, nor having to change write pre-compensation values.

With this command, the length of the Gap2 field and VCO enable timing can be altered to accommodate the unique requirements of these drives. Table 6-16 describes the effects of the WGATE and GAP bits for the PERPENDICULAR MODE command.

When both GAP and WGATE equal "0" the PERPENDICULAR MODE command will have the following effect on the 82078-1) if any of the new bits D0, D1, D2, and D3 are programmed to "1" the corresponding drive will automatically be programmed for Perpendicular mode (ie: GAP2 being written during a write operation, the programmed Data Rate will determine the length of GAP2.), and data will be written with 0 ns write pre-compensation. 2) Any of the new bits (D0–D1) that are programmed for "0", the designated drive, will be programmed for Conventional Mode and data will be written with the currently programmed write pre-compensation value. 3) Bits D0 and D1 can only be over written when the OW bit is written as a "1". The status of these bits can be determined by interpreting the eighth result byte of the DUMPREG command.

NOTE:
If either the GAP or WGATE bit is a "1", then bits D0–D1 are ignored.

"Software" and "Hardware" RESET will have the following effects on the enhanced PERPENDICULAR MODE command:

1. "Software" RESETs (Reset via DOR or DSR registers) will only clear GAP and WGATE bits to "0", D1 and D0 will retain their previously programmed values.
2. "Hardware" RESETs (Reset via pin-32) will clear all bits (GAP, WGATE, D0 and D1) to "0" (All Drives Conventional Mode).

Table 6-16. Effects of WGATE and GAP Bits

GAP	WGATE	MODE	VCO Low Time after Index Pulse	Length of Gap2 Format Field	Portion of Gap2 Written by Write Data Operation	Gap2 VCO Low Time for Read Operations
0	0	Conventional Mode	33 Bytes	22 Bytes	0 Bytes	24 Bytes
0	1	Perpendicular Mode (500 Kbps Data Rate)	33 Bytes	22 Bytes	19 Bytes	24 Bytes
1	0	Reserved (Conventional)	33 Bytes	22 Bytes	0 Bytes	24 Bytes
1	1	Perpendicular Mode (1 Mbps Data Rate)	18 Bytes	41 Bytes	38 Bytes	43 Bytes

NOTE:
When either GAP or WGATE bit is set, the current value of precompensation in the DSR is used.

6.3.13 POWERDOWN MODE COMMAND

The POWERDOWN MODE command allows the automatic power management and enables the enhanced registers (EREG EN) of the 82078. The use of the command can extend the battery life in portable PC applications. To enable auto powerdown the command may be issued during the BIOS power on self test (POST).

This command includes the ability to configure the 82078 into the enhanced mode extending the SRB and TDR registers. These extended registers accommodate bits that give more information about floppy drive interface, allow for boot drive selection, and identify the values of the PD and IDLE status.

As soon as the command is enabled, a 10 ms or a 0.5s minimum power up timer is initiated depending on whether the MIN DLY bit is set to 0 or 1. This timer is one of the required conditions that has to be satisfied before the part will enter auto powerdown. Any software reset will reinitialize the timer. The timer countdown is also extended by up to 10 ms if the data rate is changed during the timer's countdown. Without this timer 82078 would have been put to sleep immediately after 82078 is idle. The minimum delay gives software a chance to interact with 82078 without incurring an additional overhead due to recovery time.

The command also allows the output pins of floppy disk drive interface to be tri-stated or left unaltered during auto powerdown. This is done by the FDI TRI bit. In the default condition (FDI TRI = 0) the output pins of the floppy disk drive are tri-stated. Setting this bit leaves the interface unchanged from the normal state.

The results phase returns the values programmed for MIN DLY, FDI TRI and AUTO PD. The auto powerdown mode is disabled by a hardware reset. Software results have no effect on the POWERDOWN MODE command parameters.

6.3.14 PART ID COMMAND

This command can be used to identify the floppy disk controller as an enhanced controller. The first stepping of the 82078 (all 44 pin versions) will yield 0x41 in the result phase of this command. Any future enhancements on these parts will be denoted by the 5 LSBs (0x01 to 0x1F).

6.3.15 OPTION COMMAND

The standard IBM format includes an index address field consisting of 80 bytes of GAP4a, 12 bytes of the sync field, four bytes identifying the IAM and 50 bytes of GAP1. Under the ISO format, most of this preamble is not used. The ISO format allows only 32 bytes of GAP1 after the index mark. The ISO bit in this command allows the 82078 to configure the data transfer commands to recognize this format. The MSBs in this command are reserved for any other enhancements made available to the user in the future.

6.3.16 SAVE COMMAND

The first byte corresponds to the values programmed in the DSR with the exception of CLK48. The DRATE1, DRATE0 used here are unmapped. The second byte is used for configuring the bits from the OPTION command. All future enhancements to the OPTION command will be reflected in this byte as well. The next nine result bytes are explained in the Parameter Abbreviations section after the command summary. The 13th byte is the value associated with the auto powerdown command. The disk status is used internally by 82078. There are two reserved bytes at the end of this command for future use.

This command is similar to the Dumpreg command but it additionally allows the user to read back the precompensation values as well as the programmed data rate. It also allows the user to read the values programmed in the auto power down command. The precompensation values will be returned as programmed in the DSR register. This command is used in conjunction with the Restore command should prove very useful for SMM power management. This command reserves the last two bytes for future enhancements.

6.3.17 RESTORE COMMAND

Using Restore with the Save command, allows the SMM power management to restore the 82078 to its original state after a system powerdown. It also serves as a succinct way to provide most of the initialization requirements normally handled by the system. The sequence of initializing the 82078 after a reset occurred and assuming a Save command was issued follows:

- Issue the Drive Spec command (if the design utilizes this command)
- Issue the Restore command (pass the 16 bytes retrieved previously during SAVE)

The Restore command will program the data rate and precompensation value via the DSR. It then restores the values normally programmed through the Configure, Specify, and Perpendicular commands. It also enables the previously selected values for the AUTO Powerdown command. The PCN values are set restored to their previous values and the user is responsible for issuing the seek and recalibrate commands to restore the head to the proper location. There are some drives that do not recalibrate in which case the Restore command will restore the previous state completely. The PDOSC bit is retrievable using the Save command, however, the system designer must set it correctly. The software must allow at least 20μs to execute the Restore command. When using the BOOTSEL bits in the TDR, the user must restore or reinitialize these bits to their proper values.

6.3.18 FORMAT AND WRITE COMMAND

The format and write command is capable of simultaneously formatting and writing data to the diskette. It is essentially the same as the normal format command. With the exception that included in the execution for each sector is not only the C, H, R, and N but also the data transfer of N bytes. The D value is ignored. This command formats the entire track. High speed floppy diskette duplication can be done fast and efficiently with this command. The user can format the diskette and put data on it in a single pass. This is very useful for software duplication applications by reducing the time required to format and copy diskettes.

6.3.19 LOCK

The LOCK command is included to protect a system with long DMA latencies against older application software packages that can disable the 82078's FIFO. [Note: This command should only be used by the system's FDC routines, and ISVs (Independent Software Vendors) should refrain from using it. If an ISV's application calls for having the 82078 FIFO disabled, a CONFIGURE command should be used to toggle the EFIFO (Enable FIFO) bit. ISV can determine the value of the LOCK bit by interpreting the eighth result byte of an DUMPREG command.]

The LOCK command defines whether EFIFO, FIFOTHR, and PRETRK parameters of the CONFIGURE command can be RESET by the DOR and DSR registers. When the LOCK bit is set to a "1" all subsequent "software" RESETs by the DOR and DSR registers will not change the previously set parameter values in the CONFIGURE command. When the LOCK bit is set to a "0", "software" RESETs the DOR or DSR registers will return these parameters to their default values. All "hardware" Resets will set the LOCK bit to a "0" value, and will return EFIFO, FIFOTHR, and PRETRK to their default values. A Status byte is returned immediately after issuing the command byte. This Status byte reflects the value of the Lock bit set by the command byte.

NOTE:
No interrupts are generated at the end of this command.

7.0 STATUS REGISTER ENCODING

The contents of these registers are available only through a command sequence.

7.1 Status Register 0

Bit #	Symbol	Name	Description
7, 6	IC	Interrupt Code	00— Normal termination of command. The specified command was properly executed and completed without error. 01— Abnormal termination of command. Command execution was started, but was not successfully completed. 10— Invalid command. The requested command could not be executed. 11— Abnormal termination caused by Polling.
5	SE	Seek End	The 82078 completed a SEEK or RECALIBRATE command, or a READ or WRITE with implied seek command.
4	EC	Equipment Check	The TRK0 pin failed to become a "1" after: 1. 80 step pulses in the RECALIBRATE command. 2. The RELATIVE SEEK command causes the 82078 to step outward beyond Track 0.
3	—	—	Unused. This bit is always "0".
2	H	Head Address	The current head address.
1, 0	DS1, 0	Drive Select	The current selected drive.

7.2 Status Register 1

Bit #	Symbol	Name	Description
7	EN	End of Cylinder	The 82078 tried to access a sector beyond the final sector of the track (255D). Will be set if TC is not issued after Read or Write.
6	—	—	Unused. This bit is always "0".
5	DE	Data Error	The 82078 detected a CRC error in either the ID field or the data field of a sector.
4	OR	Overrun/ Underrun	Becomes set if the 82078 does not receive CPU or DMA service within the required time interval, resulting in data overrun or underrun.
3	—	—	Unused. This bit is always "0".
2	ND	No Data	Any one of the following: 1. READ DATA, READ DELETED DATA command, the 82078 did not find the specified sector. 2. READ ID command, the 82078 cannot read the ID field without an error. 3. READ TRACK command, the 82078 cannot find the proper sector sequence.
1	NW	Not Writable	WP pin became a "1" while the 82078 is executing a WRITE DATA, WRITE DELETED DATA, or FORMAT TRACK command.
0	MA	Missing Address Mark	Any one of the following: 1. The 82078 did not detect an ID address mark at the specified track after encountering the index pulse from the INDX# pin twice. 2. The 82078 cannot detect a data address mark or a deleted data address mark on the specified track.

7.3 Status Register 2

Bit #	Symbol	Name	Description
7	—	—	Unused. This bit is always "0".
6	CM	Control Mark	Any of the following: 1. READ DATA command, the 82078 encounters a deleted data address mark. 2. READ DELETED DATA command, the 82078 encountered a data address mark.
5	DD	Data Error in Data Field	The 82078 detected a CRC error in the date field.
4	WC	Wrong Cylinder	The track address from the sector ID field is different from the track address maintained inside the 82078.
3	—	—	Unused. This bit is always "0".
2	—	—	Unused. This bit is always "0".
1	BC	Bad Cylinder	The track address from the sector ID field is different from the track address maintained inside the 82078 and is equal to FF hex which indicates a bad track with a hard error according to the IBM soft-sectored format.
0	MD	Missing Data Address Mark	The 82078 cannot detect a data address mark or a deleted data address mark.

7.4 Status Register 3

Bit #	Symbol	Name	Description
7	—	—	Unused. This bit is always "0".
6	WP	Write Protected	Indicates the status of the WP pin.
5	—	—	Unused. This bit is always "1".
4	T0	TRACK 0	Indicates the status of TRK0 pin.
3	—	—	Unused. This bit is always "1".
2	HD	Head Address	Indicates the status of the HDSEL pin.
1, 0	DS1, 0	Drive Select	Indicates the status of the DS1, DS0 pins.

8.0 COMPATIBILITY

The 82078 was designed with software compatibility in mind. It is a fully backwards compatible solution with the older generation 8272A and NEC765A/B disk controllers. It is fully compatible with Intel's 386/486SL Microprocessor Superset.

8.1 Compatibility with the FIFO

The FIFO of the 82078 is designed to be transparent to non-FIFO disk controller software developed on the older generation 8272A standard. Operation of the 82078 FIFO can be broken down into two tiers of compatibility. For first tier compatibility, the FIFO is left in the default disabled condition upon a "Hardware" reset. In this mode the FIFO operates in a byte mode and provides complete compability with non-FIFO based software. For second tier compatibility, the FIFO is enabled via the CONFIGURE command. When the FIFO is enabled, it will temporarily enter a byte mode during the command and result phase of disk controller operation. This allows for compatible operation when interrogating the Main Status Register (MSR) for the purpose of transferring a byte at a time to or from the disk controller. For normal disk controller applications, the system designer can still take advantage of the FIFO for time critical data transfers during the execution phase and not create any conflicts with non-FIFO software during the command or result phase.

In some instances, use of the FIFO in any form has conflicted with certain specialized software. An example of a compatibility conflict using the FIFO is with software that monitors the progress of a data transfer during the execution phase. If the software assumed the disk controller was operating in a single byte mode and counted the number of bytes transferred to or from the disk controller to trigger some time dependent event on the disk media (i.e. head position over a specific data field), the same software will not have an identical time relationship if the FIFO is enabled. This is because the FIFO allows data to be queued up, and then burst transferred across the host bus. To accommodate software of this type, it is recommended that the FIFO be disabled.

8.2 Drive Polling

The 82078 supports the polling mode of the older generation 8272A. This mode is enabled upon a reset and can be disabled via the CONFIGURE command. This mode is supported for the sole purpose of providing backward compatibility with software that expects it's presence.

The intended purpose of drive polling dates back to 8" drives as a means to monitor any change in status for each disk drive present in the system. Each of the drives is selected for a period of time and its READY signal sampled. After a delay, the next drive is selected. Since the 82078 does not support READY in this capacity (internally tied true), the polling sequence is only simulated and does not affect the drive select lines (DS0–DS3) when it is active. If enabled, it occurs whenever the 82078 is waiting for a command or during SEEKs and RECALIBRATEs (but not IMPLIED SEEKs). Each drive is assumed to be not ready after a reset and a "ready" value for each drive is saved in an internal register as the simulated drive is polled. An interrupt will be generated on the first polling loop because of the initial "not ready" status. This interrupt must be followed with a SENSE INTERRUPT STATUS command from the host to clear the interrupt condition for each of the four logical drives.

9.0 PROGRAMMING GUIDELINES

Programming the 82078 is identical to any other 8272A compatible disk controller with the exception of some additional commands. For the new designer it is useful to provide some guidelines on how to program the 82078. A typical disk operation involves more than issuing a command and waiting for the results. The control of the floppy disk drive is a low level operation that requires software intervention at different stages. New commands and features have been added to the 82078 to reduce the complexity of this software interface.

9.1 Command and Result Phase Handshaking

Before a command or parameter byte can be issued to the 82078, the Main Status Register (MSR) must be interrogated for a ready status and proper FIFO direction. A typical floppy controller device driver should contain a subroutine for sending command or parameter bytes. For this discussion, the routine will be called "Send_byte" with the flowchart shown in Figure 9-1.

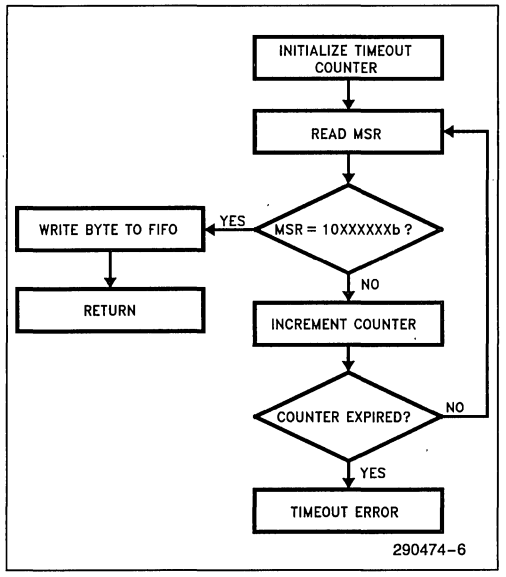


Figure 9-1. Send_Byte Routine

The routine loops until RQM is 1 and DIO is 0 indicating a ready status and FIFO direction is inward. If this condition is true, the 82078 is ready to accept a command or parameter byte. A timeout counter is used to insure software response within a reasonable amount of time in case of no response by the 82078. As a note, the programmer must be careful how the maximum delay is chosen to avoid unnecessary timeouts. For example, if a new command is issued when the 82078 is in the middle of a polling routine, the MSR will not indicate a ready status for the next parameter byte until the polling sequence completes the loop. This could cause a delay between the first and second bytes of up to 250 μs (@ 250 Kbps). If polling is disabled, this maximum delay is 175 μs. There should also be enough timeout margin to accommodate a shift of the software to a higher speed system. A timeout value that results in satisfactory operation on a 16 MHz CPU might fail when the software is moved to a system with a 25 MHz CPU. A recommended solution is to derive the timeout counter from a system hardware counter that is fixed in frequency from CPU clock to CPU clock.

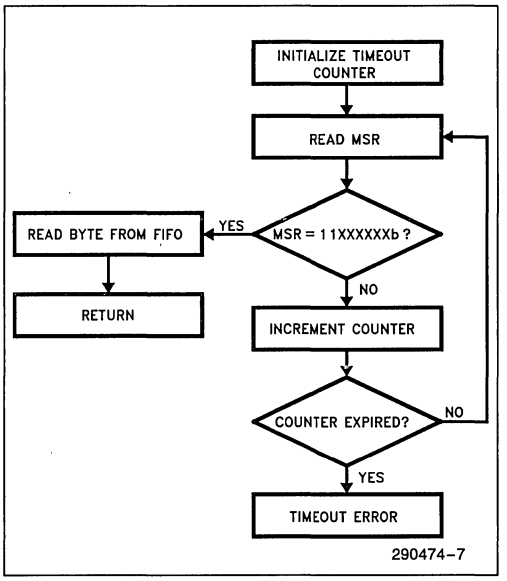


Figure 9-2. Get_Byte Routine

For reading result bytes from the 82078, a similar routine is used. Figure 9-2 illustrates the flowchart for the routine "Get_byte". The MSR is polled until RQM is 1 and DIO is 1, which indicates a ready status and outward FIFO direction. At this point, the host can read a byte from the FIFO. As in the Send_byte routine, a timout counter should be incorporated in case of a disk controller lock-up condition. For example, if a disk was not inserted into the disk drive at the time of a read operation, the controller would fail to receive the index pulse and lockup since the index pulses are required for termination of the execution phase.

9.2 Initialization

Initializing the 82078 involves setting up the appropriate configuration after a reset. Parameters set by the SPECIFY command are undefined after a system reset and will need to be reinitialized. CONFIGURE command parameters default to a known state after a system reset but will need to be reinitialized if the system requirements are different from the default settings. This can be accomplished in two ways; either issue the individual commands, or issue the Restore command (assuming the Save command was issued). The Restore command is a succinct way to initialize the 82078, this is the preferable method if the system power management powers

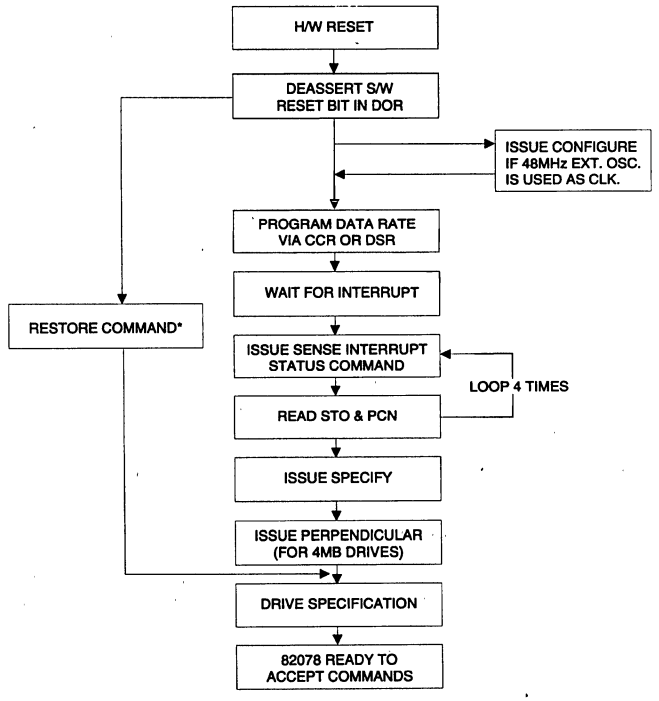
the 82078 on and off frequently. The flowchart for the recommended initialization sequence of the 82078 is shown in Figure 9-3.

Following a reset of the 82078, the Configuration Control Register (CCR) should be reinitialized for the appropriate data rate. An external reset via the RESET pin will cause the data rate and write precompensation values to default to 250 Kbps (10b) and 125 ns (000b) respectively. Since the 125 ns write precompensation value is optimal for the 5¼" and 3½" disk drive environment, most applications will not require the value to be changed in the initialization sequence. As a note, a software reset issued via the DOR or DSR will not affect the data rate or write precompensation values. But it is recommended as a safe programming practice to always program the data rate after a reset, regardless of the type.

Since polling is enabled after a reset of the 82078, four SENSE INTERRUPT STATUS commands need to be issued afterwards to clear the status flags for each drive. The flowchart in Figure 9-3 illustrates how the software clears each of the four interrupt status flags internally queued by the 82078. It should be noted that although four SENSE INTERRUPT STATUS commands are issued, the INT pin is only active until the first SENSE INTERRUPT STATUS command is executed.

As a note, if the CONFIGURE command is issued within 250 μs of the trailing edge of reset (@1 Mbps), the polling mode of the 82078 can be disabled before the polling initiated interrupt occurs. Since polling stops when the 82078 enters the command phase, it is only time critical up to the first byte of the CONFIGURE command. If disabled in time, the system software no longer needs to issue the four SENSE INTERRUPT STATUS commands to clear the internal interrupt flags normally caused by polling.

The CONFIGURE command should also be issued if the system requirements are different from the default settings. For example, the CONFIGURE command can be used to enable the FIFO, set the threshold, and enable Implied Seeks.



NOTE:
(*) A sense interrupt status may be required if the Restore command is not issued within 250 μs of the trailing edge of a H/W reset (@1 Mbps).

Figure 9-3. Initialization Flowchart

The non-DMA mode flag, step rate (SRT), head load (HLT), and head unload times (HUT) programmed by the SPECIFY command do not default to a known state after a reset. This behavior is consistent with the 8272A and has been preserved here for compatibility. Thus, it is necessary to always issue a SPECIFY command in the initialization routine.

9.3 Recalibrates and Seeks

Commands that position the disk head are different from the typical READ/WRITE/FORMAT command in the sense that there is no result phase. Once a RECALIBRATE, SEEK, or RELATIVE SEEK command has been issued, the 82078 will return a ready status in the Main Status Register (MSR) and perform the head positioning operation as a background task. When the seek is complete, the 82078 will assert the INT signal to request service. A SENSE INTERRUPT STATUS command should then be asserted to clear the interrupt and read the status of the operation. Since the drive and motor enable signals are directly controlled through the Digital Output Register (DOR) on the 82078, a write to the DOR will need to precede the RECALIBRATE or SEEK command if the drive and motor is not already enabled. Figure 9-4 shows the flow chart for this operation.

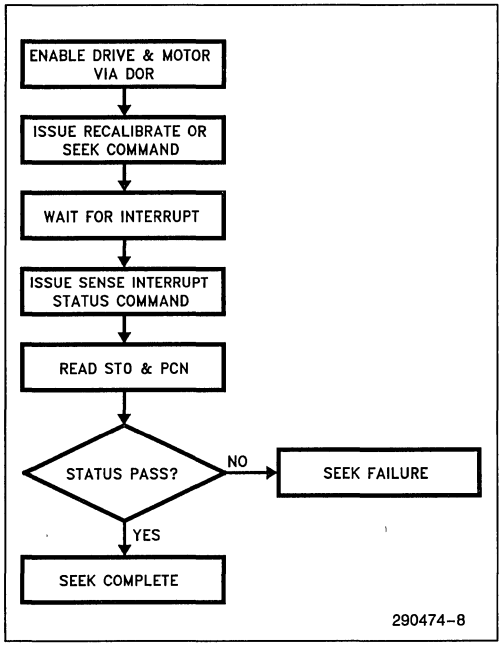


Figure 9-4. Recalibrate and Seek Operations

9.4 Read/Write Data Operations

A read or write data operation requires several steps to complete successfully. The motor needs to be turned on, the head positioned to the correct cylinder, the DMA controller initialized, the read or write command initiated, and an error recovery scheme implemented. The flowchart in Figure 9-5 highlights a recommended algorithm for performing a read or write data operation.

Before data can be transferred to or from the diskette, the disk drive motor must be brought up to speed. For most 3½" disk drives, the spin-up time is 300 ms, while the 5¼" drive usually requires about 500 ms due to the increased moment of inertia associated with the larger diameter diskette.

2

One technique for minimizing the motor spin-up delay in the read data case is to begin the read operation immediately after the motor is turned on. When the motor is not initially up to speed, the internal data separator will fail to lock onto the incoming data stream and report a failure in the status registers. The read operation is then repeated until successful status is obtained. There is no risk of a data integrity problem since the data field is CRC validated. But, it is not recommended to use this technique for the write data operation even though it requires successful reading of the ID field before the write takes place. The data separator performance of the 82078 is such that locking to the data stream could take place while the motor speed variation is still significant. This could result in errors when an attempt is made to read the disk media by other disk controllers that have a narrower incoming data stream frequency bandwidth.

After the motor has been turned on, the matching data rate for the media inserted into the disk drive should then be programmed to the 82078 via the Configuration Control Register (CCR). The 82078 is designed to allow a different data rate to be programmed arbitrarily without disrupting the integrity of the device. In some applications, it is required to automatically determine the recorded data rate of the inserted media. One technique for doing this is to perform a READ ID operation at each available data rate until a successful status is returned in the result phase.

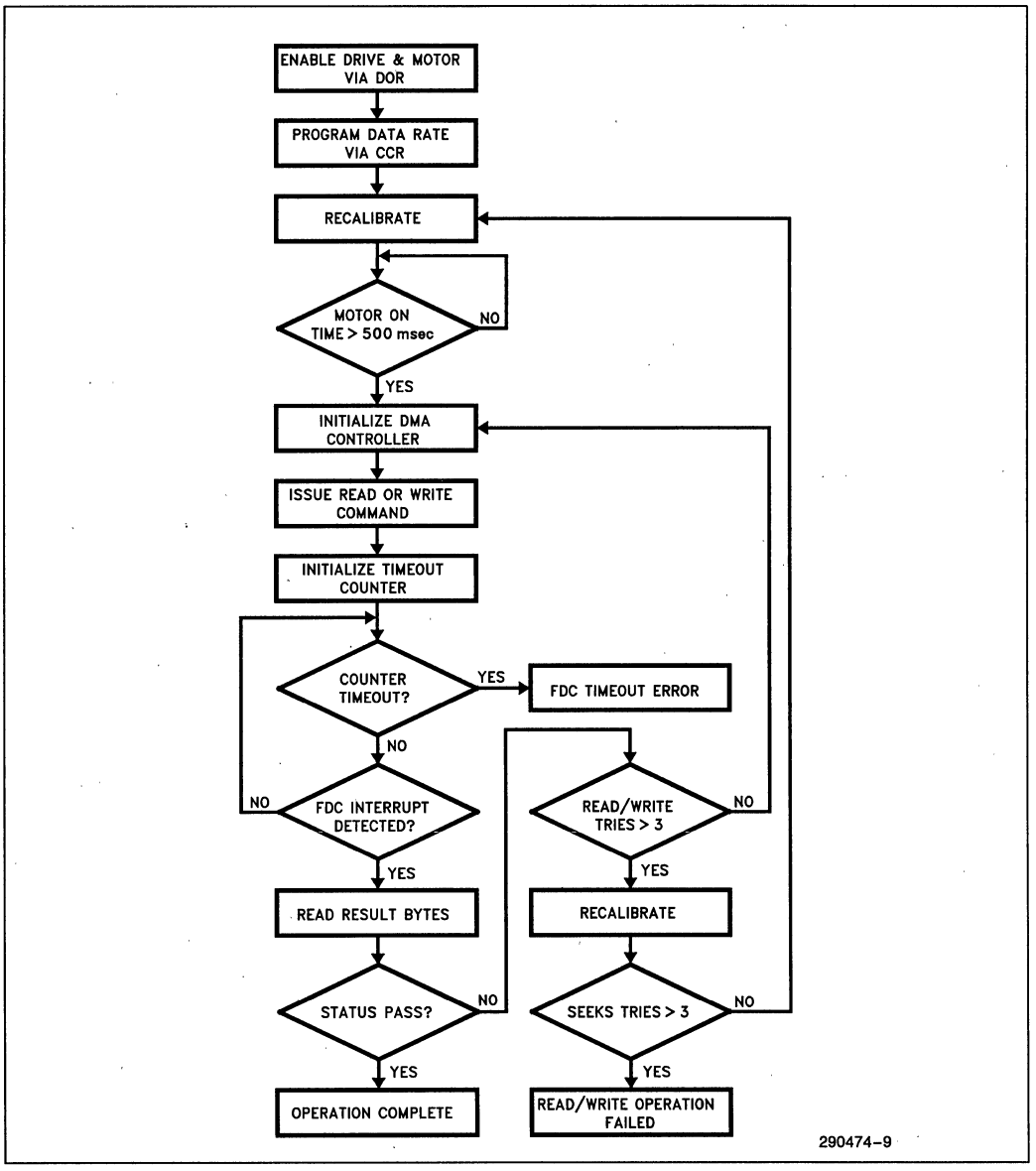


Figure 9-5. Read/Write Operation

If implied seeks are not enabled, the disk drive head must be positioned over the correct cylinder by executing a SEEK command. After the seek is complete, a head settling time needs to be asserted before the read or write operation begins. For most drives, this delay should be a minimum of 15 ms. When using implied seeks, the minimum head settling time can be enforced by the head load time (HLT) parameter designated in the SPECIFY command. For example, a HLT value of 8 will yield an effective head settling time of 16 ms for a programmed data rate of 500 Kbps. Of course if the head is already positioned over the correct cylinder, the head settling time does not need to be enforced.

The DMA controller is then initialized for the data transfer and the read or write command is executed. Typically the DMA controller will assert Terminal Count (TC) when the data transfer is complete. The 82078 will then complete the current data transfer and assert the INT signal signifying it has entered the result phase. The result phase can also be entered by the 82078 if an error is encountered or the last sector number equals the End of Track (EOT) parameter.

Based on the algorithm in Figure 9-5, if an error is encountered after reading the result bytes, two more retries are performed by reinitializing the DMA controller and re-issuing the read or write data command. A persisting failure could indicate the seek operation did not achieve proper alignment between the head and the track. The disk head should then be recalibrated and the seek repeated for a maximum of two more tries. Unsuccessful operation after this point should be reported as a disk failure to the operating system.

9.5 Formatting

The disk formatting procedure involves positioning the head on each track and creating a fixed format field used for organizing the data fields. The flowchart in Figure 9-6 highlights the typical format procedure.

After the motor has been turned on and the correct data rate programmed, the disk head is recalibrated to track 0. The disk is then allowed to come up to speed via a 500 ms delay. It is important the disk speed has stabilized before the actual formatting to avoid any data rate frequency variations. Since the format fields contain critical information used by the data separator of the disk controller for synchronization purposes, frequency stability of the data stream is imperative for media interchangeability among different systems.

The ID field data created on the disk during the format process is provided by the DMA controller during the execution phase. The DMA controller is initialized to send the C, H, R and N values for each sector ID field. For example, to format cylinder 7, on head 1, with 9 sectors, and a sector size of 2 (512 bytes), the DMA controller should be programmed to transfer 36 bytes (9 sectors × 4 bytes per sector) with the following data field: 7,1,1,2, 7,1,2,2, 7,1,3,2, ... 7,1,9,2. Since the values provided to the 82078 during the execution phase of the format command are directly recorded as the ID fields on the disk, the data contents can be arbitrary. Some forms of copy protection have been implemented by taking advantage of this capability.

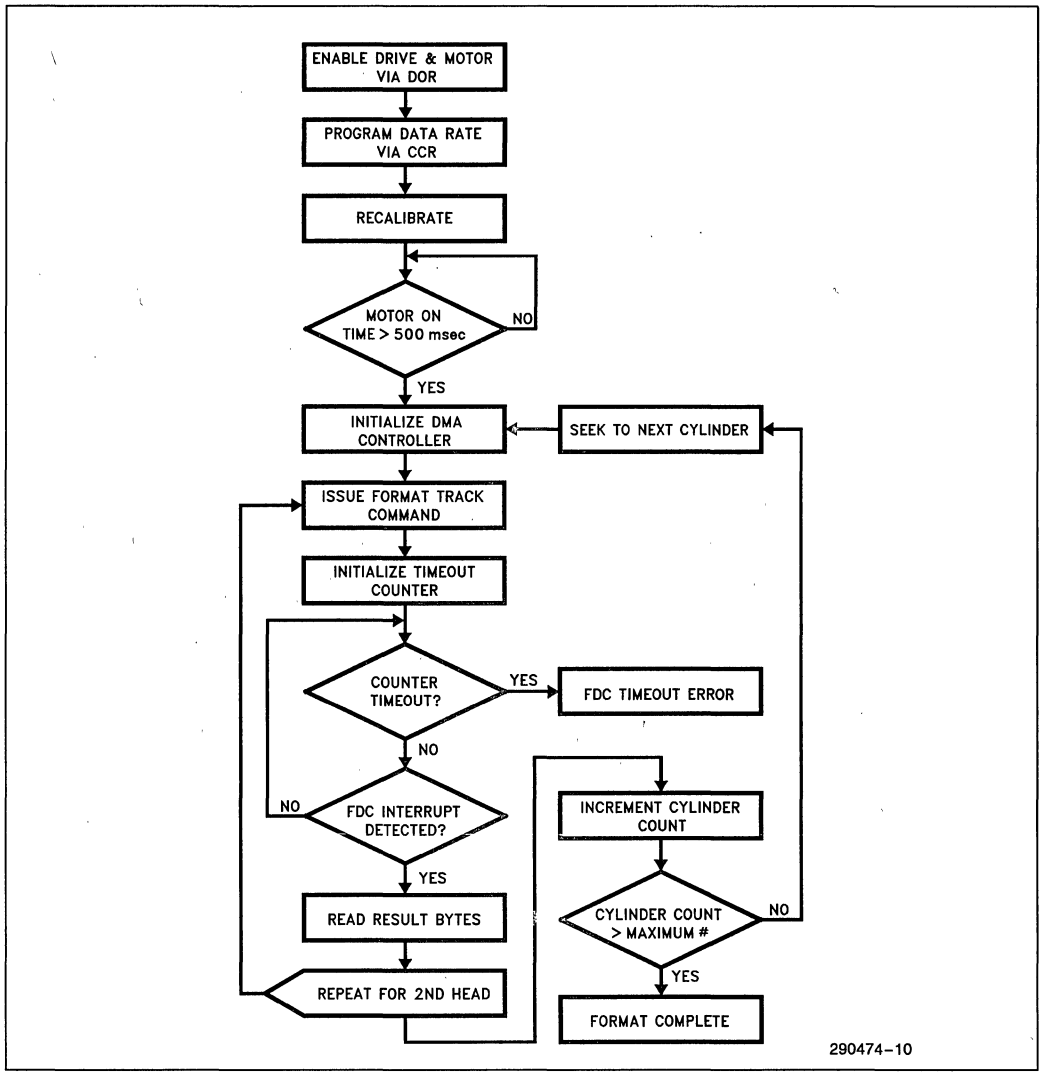


Figure 9-6. Formatting

After each head for a cylinder has been formatted, a seek operation to the next cylinder is performed and the format process is repeated. Since the FORMAT TRACK command does not have implied seek capability, the SEEK command must be used. Also, as discussed in Section 9-2, the head settling time needs to be adhered to after each seek operation.

9.6 Save and Restore

The Save and Restore commands were developed for portable systems that use zero-volt powerdown to conserve power. These systems turn off the V_{CC} to most of the system and retain the system status in a specific location. In older floppy controller designs, in order for system designers to retrieve the floppy controller status, a lot of separate commands and register reads were required. The Save command stores the key status information in a single command, the Restore command restores this information with a single command. These commands can be integrated into the SMM module that is responsible for zero-volt powerdown.

The sequence of initializing the 82078 after a reset occurred and assuming a Save command was issued follows:

- Issue the Drive Spec command (if the design utilizes this command)
- Issue the Restore command

The Restore command programs the data rate and precompensation value via the DSR. It then restores the values normally programmed through the Configure, Specify, and Perpendicular commands. It also enables the previously selected values for the AUTO Powerdown command. The command then restores the PCN values to its previous values. The user is responsible for issuing the seek and recalibrate commands to restore the head to the proper location. There are some drives that do not recalibrate in which case the Restore command will restore the previous state completely. The PDOSC bit is retrievable using the Save command, however it is up to the system designer to set it correctly. The software must allow at least 20μs to execute the Restore command. When using the BOOTSEL bit in the TDR, the user must restore or reinitialize this bit to it's proper value.

9.7 Verifies

In some applications, the sector data needs to be verified immediately after each write operation. One verify technique reinitializes the DMA controller to perform a read transfer or verify transfer (DACK# is asserted but not RD#) immediately after each write operation. Issue a read command to the disk controller and the resulting status indicates if the CRC validated the previously written data. This technique has the drawback of requiring additional software intervention by having to reprogram the DMA controller between each sector write operation. The 82078 supports this verify technique but also provides a VERIFY command that does not require the use of the DMA controller.

To verify a write data transfer or format track operation using the VERIFY command, the software simply issues the command with the same format as a READ DATA command but without the support of the DMA controller. The 82078 will then perform a disk read operation without a host data transfer. The CRC will be calculated for each sector read and compared against the value stored on the disk. When the VERIFY command is complete, the status register reports detected CRC errors.

9.8 Powerdown State and Recovery

The two power management modes coupled with the internal oscillator power management forms an important consideration for programming the 82078. The recovery of 82078 and the time it takes to achieve complete recovery depends on how 82078 is powered down and how it is awakened. The following sections describe all the programming concerns and subtleties involved in using power management features of the 82078. The 3.3V version of the 82078 has the same power saving features as the 5.0V versions.

9.8.1 OSCILLATOR POWER MANAGEMENT

Section 4.1 covers the power management scheme involved in powering down of both an internal and an external oscillator. Both types of oscillators face drop out effects and require recovery times on the order of tens of milliseconds (this may be objectionable to some application software). This means that if the oscillator is powered down then it is imperative for the software to assure enough time for the oscillator to recover to a stable state. Oscillator power management must be controlled by the system software especially to maintain software transparency. In cases where the system goes into a standby mode (by user request or system time-out), the power management software can turn off the oscillator to conserve power. This can also be controlled in hardware using the Powerdown (PD#) pin. Complete recovery from an oscillator powerdown state requires the software to turn on the oscillator sufficiently ahead of awakening the 82078.

9.8.2 PART POWER MANAGEMENT

The part powerdown and wake up modes are covered in Section 4.2 in detail. This section is meant to address the programming concerns for the part (excluding the oscillator) during these modes.

9.8.2.1 Powerdown Modes

For both types of powerdown modes—DSR powerdown and auto powerdown, if reset is used to exit the part from powerdown then the internal microcontroller will go through a standard sequence: register initialization followed after some delay by an interrupt.

Software transparency in auto powerdown mode is preserved by MSR retaining the value of 80H which indicates that the part is ready to receive a command. This feature allows the part to powerdown while maintaining its responsiveness to any application software.

The PD and IDLE status bits can be monitored via the Status Register B (SRB, EREG EN = 1). Since the IDLE# pin stays high when the 82078 is in idle state, the IDLEMSK bit can be used to set the pin low again (as part of a power management routine).

NOTE:
The IDLEMSK prevents the user from knowing if the part has entered auto powerdown or DSR powerdown.

9.8.2.2 Wake Up Modes

Wake up from DSR powerdown results in the part being internally reset and all present status being lost. During DSR powerdown the RQM bit in the MSR is set. A software or hardware reset will wake up the part.

The case for wake up from auto powerdown is different. The BIOS and application software are very sensitive to delays involved in writing the first command bytes to the 82078. Most programs have short error time-outs in these cases. Such programs would not tolerate any floppy disk controller that was unable to receive the first byte of a command at any time. The following describes how 82078 uniquely sustains its software transparency during wake up sequences.

Prior to writing a command to 82078, it is first necessary to read the MSR to ensure that the 82078 is ready (RQM bit must be set) to receive the command. When the part detects a MSR read, it assumes that another command will follow and begins the wake up process. While the part is waking up it does not change the state of the MSR (MSR = 80H) and is able to receive the command in the FIFO. At this point one of the two following scenarios can occur.

No other command is sent subsequent to the MSR read. The part wakes up and initializes the minimum power up timer. Upon the expiration of this timer the part is once again put in powerdown state.

Another command follows the MSR read. If the command is sent during the part's recovery from powerdown, the part remembers the command, clears the RQM bit (to prevent further bytes being written) and acts on the command once it is fully awake.

If the MSR was not checked prior to writing of a command, the part will proceed as stated above with the RQM bit cleared and the command byte held until the internal microcontroller is ready. Writing the motor enable bits in DOR active will initiate the wake up sequence with RQM set high, ready to receive any command.

As it is clear from the above discussion, the immediate access to the floppy disk controller for the first command byte is vital to software transparency. The recovery of the part from powerdown may involve a delay after the first command byte has been issued. However, all programs have tolerance for the delay after the first command byte is issued. In a powered up chip, it is possible for the microcontroller to be in its "polling loop". As a result, the tolerance for this delay provides an excellent window for recovery of the part.

10.0 DESIGN APPLICATIONS

10.1 Operating the 82078-3 in a 3.3V Design

The design for 3.3V is the same as it is for 5.0V, however the floppy drive interface signals can be at either 3.3V or 5.0V levels depending on the voltage on the V_{CCF} pin. The V_{CCF} pin allows the FDD interface to be operated in mixed (3.3V/5.0V) mode. For example, if the system operates at 3.3V and the floppy disk drive operates at 5.0V, the 82078 can be configured to operate at 3.3V with 5.0V available to the drive interface. See Figure 10-1 for a schematic.

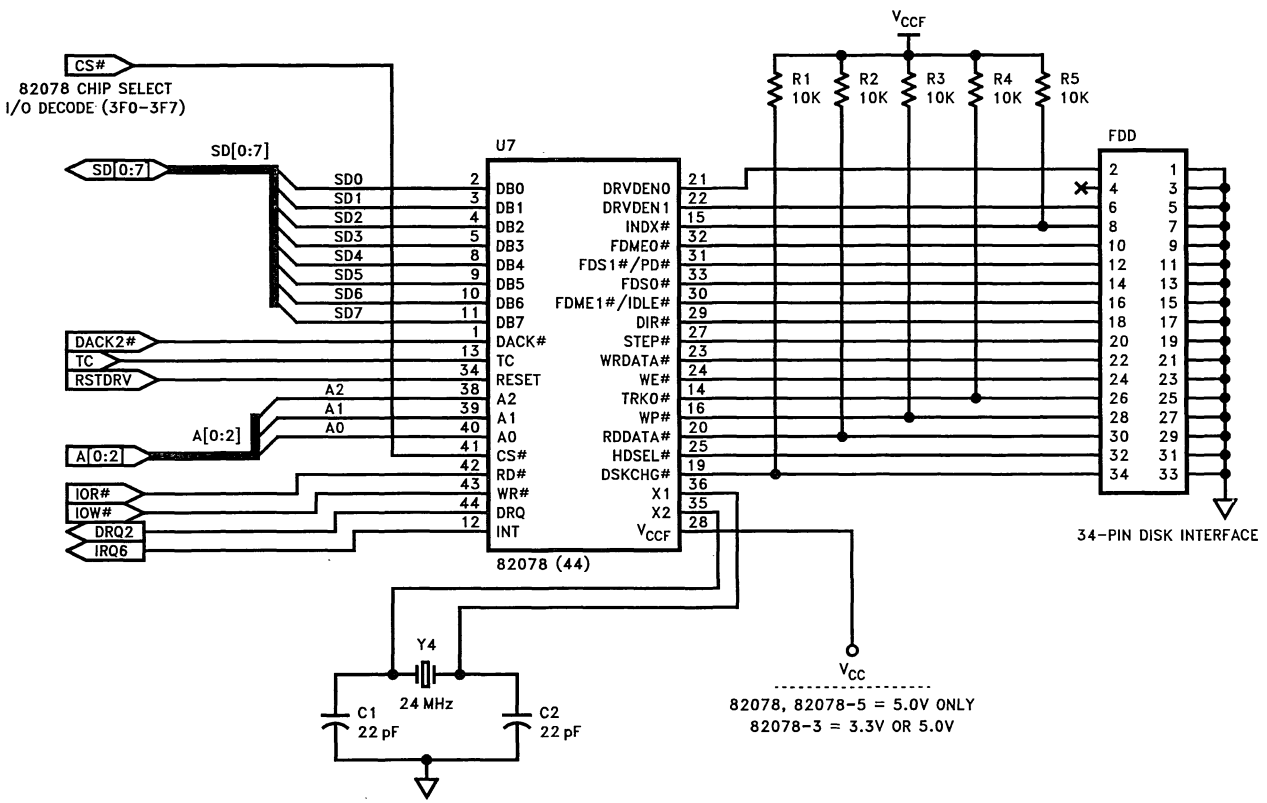


Figure 10-1. 82078 PC/AT Design

10.2 Selectable Boot Drive

Generally a standard personal computer is configured with a 1.2 MB 5.25″ disk drive and a 1.44 MB or 2.88 MB 3.5″ disk drive. Usually the drive that connects as "A:" is the boot drive. At times the user may want to configure "B:" as the boot drive. Currently some BIOS' use a special implementation in software to accomplish this. The 82078 now offers this capability more efficiently by configuring the boot drives.

The 82078 allows for virtual drive designations. This is a result of multiplexing the boot drive select and motor enable lines, as shown in Figure 10-2.

The DRIVE SEL1 and the DRIVE SEL2 bits in the DOR register decode internally to generate the signals DSn. The MEn signals generate directly from the DOR register. The DSn and MEn signals get mapped to actual FDSn and FDMEn pins based on the BOOTSELn bits (selected in the TDR register). The exact mapping of BOOTSEL vs the FDSn and FDMEn pins is shown in the following table.

44PD EN	BOOTSEL (TDR)	Mapping
0	0	Default → DS0 → FDS0, ME0 → FDME0 DS1 → FDS1, ME1 → FDME1
0	1	DS0 → FDS1, ME0 → FDME1 DS1 → FDS0, ME1 → FDME0
1	X	DS0 → FDS0, ME0 → FDME0 DS1 → PD, ME1 → IDLE

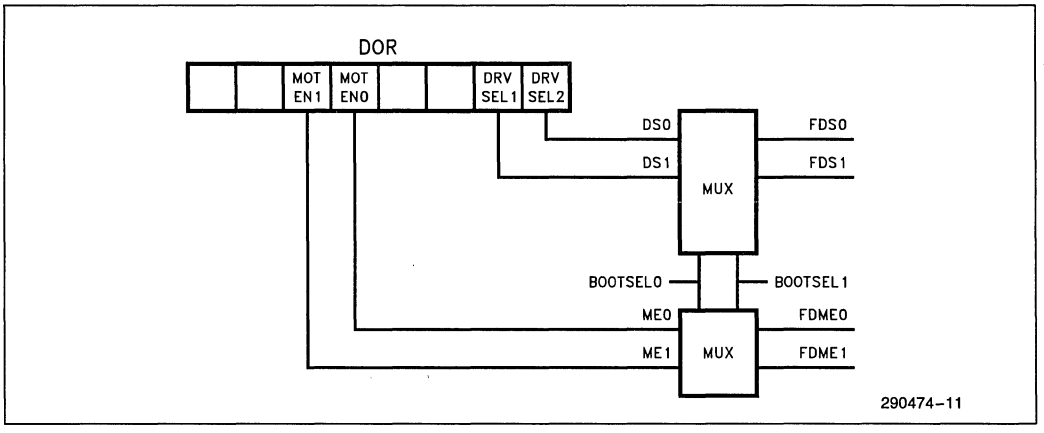


Figure 10-2. Virtual Drive Configuration

The BOOTSEL bit allows users to multiplex the output drive signals allowing different drives to be the boot drive. The DSn and MEn bits are considered virtual designations since the DSn and MEn signals get remapped to different corresponding physical FDSn and FDMEn pins. In other words, once the BOOTSEL bit is configured for a non-default selection, all future references made to the controller will be assumed as virtual designations. Note, due to the virtual designations TAPESEL[1:0] = 00 would never enable tape mode due to boot drive restrictions.

10.3 How to Disable the Native Floppy Contoller on the Motherboard

There are occasions when the floppy controller designed onto the motherboard of a system needs to be disabled in order to operate another floppy controller on the expansion bus. This can be done without changing the BIOS or remapping the address of the floppy controller (provided there is a jumper, or another way to disable the chip select on the native controller).

Upon reset, the DOR register in the 82078 is set to 00H. If the CS# is left enabled during the POST, the DOR is set to 0CH, this enables the DMA GATE# bit in the DOR. When this bit is set, the 82078 treats a DACK# and a RD# or WR# as an internal chip select (CS#). Bus contention will occur between the native controller and the auxiliary controller if the DMA GATE# bit becomes active, even if the CS# signal is not present.

The proper way to disable the native floppy controller is to disable the CS# before the system is turned on. This will prevent the native controller from getting initialized. Another option is to map the native controller to a secondary address space, then disable the DMA GATE# via the DOR disabling the DMA GATE#. This assumes that the native controller is switched to a secondary address space.

10.4 Replacing the 82077SL with a 82078 in a 5.0V Design

The 82078 easily replaces the 5.0V 82077SL with minimum design changes. With a few exceptions, most of the signals are named as they were in the 82077SL. Some pins were eliminated and others renamed to accommodate a reduced pin count and smaller package.

The connections to the AT bus are the same as the 82077SL with the following exceptions: MFM and IDENT have been removed. The PLL0 pin was removed. Tape drive mode on the 82078 must be configured via the Tape Drive Register (TDR).

The Drive Interface on the 82078 is also similar to the 82077SL except as noted: DRVDEN0 and DRVDEN1 on the 82078 take the place of DENSEL, DRATE0, and DRATE1 on the 82077SL. The Drive Specification Command configures the polarity of these pins, thus selecting the density type of the drive. The Motor Enable pins and the Drive Select pins are renamed FDME(0–1) and FDS(0–1) respectively on the 82078. 10K pull-up resistors can be used on the disk interface. See Figure 10-3 for a schematic of the connection.

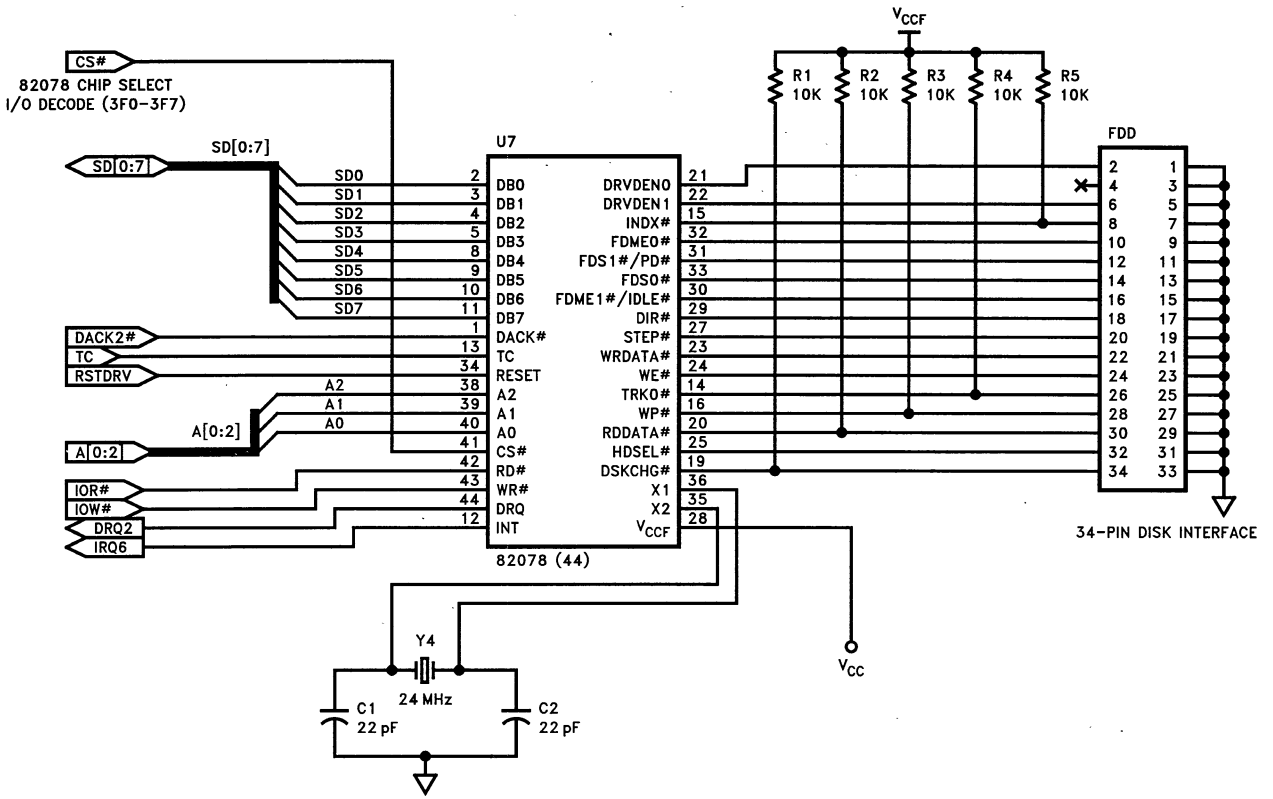


Figure 10-3. 82077SL Conversion to 82078

Pin Changes on the 44 Pin Part:

- If the 44PD EN bit in the powerdown command is set, then the FDS1# and FDME1# no longer function as drive select and motor enable. Instead these pins become functional as status outputs of PD and IDLE.
- INVERT# is removed.
- Four NCs (no connects) are removed.
- MFM, IDENT have been removed. The 44 pin 82078 only operates in AT/EISA mode.
- PLL0 is removed. Hardware configurability for tape drive mode is not supported. Configure tape mode via the TDR register.
- DENSEL, DRATE1, DRATE0 pins have been substituted by DRVDEN0, DRVDEN1. The new pins are configured for each drive via the Drive Specification command.
- DRV2 and RDGATE are not available.
- There are 3 V_{SS} pins, 2 V_{CC} pins, one AV_{SS} and one AV_{CC} pin.

11.0 D.C. SPECIFICATIONS

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

11.1 Absolute Maximum Ratings

**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.*

Storage Temperature ..... −65°C to +150°C
Supply Voltage ..... −0.5 to +8.0V
Voltage on Any Input ..... GND − 2V to 6.5V
Voltage on Any Output . GND − 0.5V to V_{CC} + 0.5V
Power Dissipation 1W

11.2 D.C. Characteristics T_A = 0°C to +70°C, V_{SS} = AV_{SS} = 0V

44 PIN D.C. CHARACTERISTICS

Symbol	Parameter	V_{CC} = +5V ±10%			V_{CC} = 3.3V ±0.3V		
		Min (V)	Max (V)	Test Conditions	Min (V)	Max (V)	Test Conditions
V_{ILC}	Input Low Voltage, X1	−0.5	0.8		−0.3	0.8	
V_{IHC}	Input High Voltage, X1	3.9	V_{CC} + 0.5		2.4	V_{CC} + 0.3	
V_{IL}	Input Low Voltage (All Pins except X1)	−0.5	0.8		−0.3	0.8	
V_{IH}	Input High Voltage (All Pins except X1)	2.0	V_{CC} + 0.5		2.0	V_{CC} + 0.3	
V_{OL}(5)	System Interface		0.45	I_{OL} = 12 mA		0.45	I_{OL} = 6 mA
	FDD Interface Outptus		0.45	I_{OL} = 12 mA		0.45	I_{OL} = 6 mA
V_{OH}	All Outputs	3.0		I_{OH} = −4.0 mA	2.4		I_{OH} = −2.0 mA
	All Outputs	V_{CC} − 0.4		I_{OH} = −100 µA	V_{CC} − 0.2		I_{OH} = −100 µA

44 PIN D.C. CHARACTERISTICS I_{CC}

Symbol	Parameter	V_{CC} = +5V ±10%			V_{CC} = +3.3V ±0.3V		
		Typical	Max	Test Condition	Typical	Max	Test Condition
I_{CC1}	1 Mbps Data Rate V_{IL} = V_{SS}, V_{IH} = V_{CC}	15.4 mA	25 mA	(Notes 1, 2)	8.4 mA	16 mA	(Notes 1, 2)
I_{CC2}	1 Mbps Data Rate V_{IL} = 0.45, V_{IH} = 2.4	20.8 mA	30 mA	(Notes 1, 2)	8.6 mA	16 mA	(Notes 1, 2)
I_{CC3}	500 Kbps Data Rate V_{IL} = V_{SS}, V_{IH} = V_{CC}	11.8 mA	20 mA	(Notes 1, 2)	6.2 mA	14 mA	(Notes 1, 2)
I_{CC4}	500 Kbps Data Rate V_{IL} = 0.45, V_{IH} = 2.4	17.6 mA	25 mA	(Notes 1, 2)	6.2 mA	14 mA	(Notes 1, 2)
I_{CCSB}	I_{CC} in Powerdown	0 µA	60 µA	(Notes 3, 4)	0 µA	60 µA	(Notes 3, 4)
I_{IL}	Input Load Current (All Input Pins)		10 µA −10 µA	V_{IN} = V_{CC} V_{IN} = 0V		10 µA −10 µA	V_{IN} = V_{CC} V_{IN} = 0V

44 PIN D.C. CHARACTERISTICS I_{CC} (Continued)

Symbol	Parameter	V_{CC} = +5V ±10%			V_{CC} = +3.3V ±0.3V		
		Typical	Max	Test Condition	Typical	Max	Test Condition
I_{OFL}	Data Bus Output Float Leakage		±10 μA	0.45 < V_{OUT} < V_{CC}		±10 μA	0.45 < V_{OUT} < V_{CC}

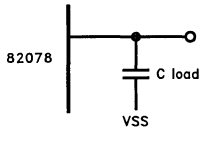
NOTES:
1. Only the data bus inputs may float.
2. Tested while reading a sync field of "00". Outputs not connected to D.C. loads.
3. V_{IL} = V_{SS}, V_{IH} = V_{CC}; Outputs not connected to D.C. loads.
4. Typical value with the oscillator off.
5. V_{OL} change effective for both 44-pin and 64-pin package offerings.

CAPACITANCE

C_{IN}	Input Capacitance	10	pF	f = 1 MHz, T_A = 25°C
C_{IN1}	Clock Input Capacitance	20	pF	Sampled, Not 100% Tested
$C_{I/O}$	Input/Output Capacitance	20	pF	

NOTE:
All pins except pins under test are tied to A.C. ground.

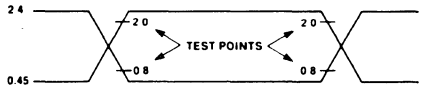
LOAD CIRCUIT



290474-13

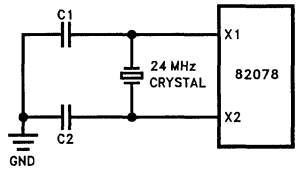
NOTE:
C_{load} = 50 pF for all logic outputs, 100 pF for the data bus.

A.C. TESTING INPUT, OUTPUT WAVEFORM

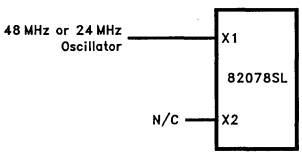


290474-14

11.3 Oscillator



290474-15



290474-28

The 24 MHz clock can be supplied either by a crystal or a MOS level square wave. All internal timings are referenced to this clock or a scaled count which is data rate dependent.

The crystal oscillator must be allowed to run for 10 ms after V_{CC} has reached 4.5V or exiting the POWERDOWN mode to guarantee that it is stable.

Frequency:	24 MHz ±0.1%
Mode:	Parallel Resonant Fundamental Mode
Series Resistance:	Less than 40Ω
Shunt Capacitance:	Less than 5 pF

12.0 A.C. SPECIFICATIONS

T_A = 0°C to +70°C, V_{CC} = +5V ±10%, +3.3V ±0.3V, V_{SS} = AV_{SS} = 0V

Symbol	Parameter	Min	Max	Unit
CLOCK TIMINGS				
t1	Clock Rise Time		10	ns
	Clock Fall Time		10	ns
t2	Clock High Time(7)	16	26	ns
t3	Clock Low Time(7)	16	26	ns
t4	Clock Period	41.66	41.66	ns
t5	Internal Clock Period (3)			
HOST READ CYCLES				
t7	Address Setup to RD#	5		ns
t8	RD# Pulse Width	90		ns
t9	Address Hold from RD#	0		ns
t10	Data Valid from RD#(12)		80	ns
t11	Command Inactive	60		ns
t12	Output Float Delay		35	ns
t13	INT Delay from RD#(16)		t5 + 125	ns
t14	Data Hold from RD#	5		ns
HOST WRITE CYCLES				
t15	Address Setup to WR#	5		ns
t16	WR# Pulse Width	90		ns
t17	Address Hold from WR#	0		ns
t18	Command Inactive	60		ns
t19	Data Setup to WR#	70		ns
t20	Data Hold from WR#	0		ns
t21	INT Delay from WR#(16)		t5 + 125	ns
DMA CYCLES				
t22	DRQ Cycle Period(1)	6.5		µs
t23	DACK# to DRQ Inactive		75	ns
t23a	DRQ to DACK# Inactive	(Note 15)		ns
t24	RD# to DRQ Inactive(4)		100	ns
t25	DACK# Setup to RD#, WR#	5		ns
t26	DACK# Hold from RD#, WR#	0		ns
t27	DRQ to RD#, WR# Active(1)	0	6	µs
t28	Terminal Count Width(10)	50		ns
t29	TC to DRQ Inactive		150	ns

12.0 A.C. SPECIFICATIONS

T_A = 0°C to +70°C, V_{CC} = +5V ±10%, +3.3V ±0.3V, V_{SS} = AV_{SS} = 0V (Continued)

Symbol	Parameter	Min	Max	Unit
RESET				
t30	"Hardware" Reset Width(5)	1.13		µs
t30a	"Software" Reset Width(5)	(Note 11)		ns
t31	Reset to Control Inactive		2	µs
WRITE DATA TIMING				
t32	Data Width(6)			ns
DRIVE CONTROL				
t35	DIR# Setup to STEP#(14)	1.0		µs
t36	DIR# Hold from STEP#	10		µs
t37	STEP# Active Time (High)	2.5		µs
t38	STEP# Cycle Time(2)			µs
t39	INDEX# Pulse Width	5		t5
t41	WE# to HDSEL# Change	(Note 13)		ms
READ DATA TIMING				
t40	Read Data Pulse Width	50		ns
t44	PLL Data Rate	90		ns
	82078		1M	bits/sec
t44	Data Rate Period = 1/f44			
tLOCK	Lockup Time		64	t44

NOTES:

1. This timing is for FIFO threshold = 1. When FIFO threshold is N bytes, the value should be multiplied by N and subtract 1.5 µs. The value shown is for 1 Mbps, scales linearly with data rate.
2. This value can range from 0.5 ms to 8.0 ms and is dependent upon data rate and the Specify command value.
3. Many timings are a function of the selected data rate. The nominal values for the internal clock period (t5) for the various data rates are:

1 Mbps	3 × oscillator period = 125 ns
500 Kbps	6 × oscillator period = 250 ns
300 Kbps	10 × oscillator period = 420 ns
250 Kbps	12 × oscillator period = 500 ns

4. If DACK# transitions before RD#, then this specification is ignored. If there is no transition on DACK#, then this becomes the DRQ inactive delay.

5. Reset requires a stable oscillator to meet the minimum active period.
6. Based on the internal clock period (t5). For various data rates, the Write Data Width minimum values are:

1 Mbps	5 × oscillator period −50 ns = 150 ns
500 Kbps	10 × oscillator period −50 ns = 360 ns
300 Kbps	16 × oscillator period −50 ns = 615 ns
250 Kbps	19 × oscillator period −50 ns = 740 ns

7. Test points for clock high time are 3.5V. Due to transitional times, clock high time max and clock low time max cannot be met simultaneously. Clock high time min and clock low time max can not be met simultaneously.
8. Based on internal clock period (t5).
9. Jitter tolerance is defined as:
   (Maximum bit shift from nominal position ÷ ¼ period of nominal data rate) × 100%
   is a measure of the allowable bit jitter that may be present and still be correctly detected. The data separator jitter tolerance is measured under dynamic conditions that jitters the bit stream according to a reverse precompensation algorithm.
10. TC width is defined as the time that both TC and DACK# are active. Note that TC and DACK# must overlap at least 50 ns.
11. The minimum reset active period for a software reset is dependent on the data rate, after the 82078 has been properly reset using the t30 spec. The minimum software reset period then becomes:

1 Mbps	3 × t4 = 125 ns
500 Kbps	6 × t4 = 250 ns
300 Kbps	10 × t4 = 420 ns
250 Kbps	12 × t4 = 500 ns

12. Status Register's status bits which are not latched may be updated during a Host read operation.
13. The minimum MFM values for WE to HDSEL change (t41) for the various data rates are:

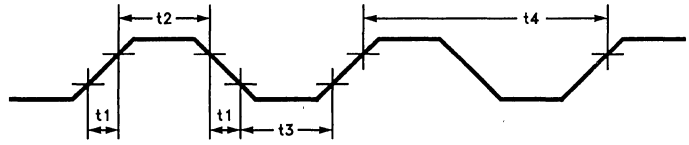
1 Mbps	0.5 ms + [8 × GPL]
500 Kbps	1.0 ms + [16 × GPL]
300 Kbps	1.6 ms + [26.66 × GPL]
250 Kbps	2.0 ms + [32 × GPL]

   GPL is the size of gap3 defined in the sixth byte of a Write Command.
14. This timing is a function of the selected data rate as follows:

1 Mbps	1.0 μs min
500 Kbps	2.0 μs min
300 Kbps	3.3 μs min
250 Kbps	4.0 μs min

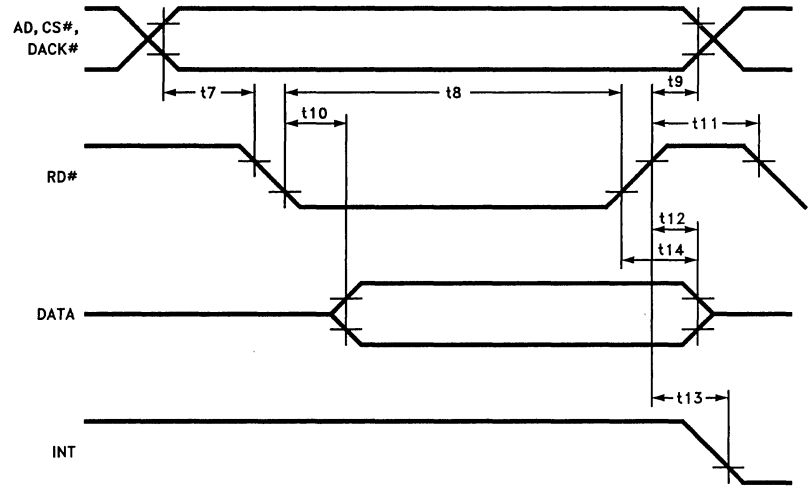
15. This timing is a function of the internal clock period (t5) and is given as (⅔) t5. The values of t5 are shown in Note 3.
16. The timings t13 and t21 are specified for INT signal in the polling mode only. These timings in case of the result phase of the read and write commands are microcode dependent.

Clock Timings



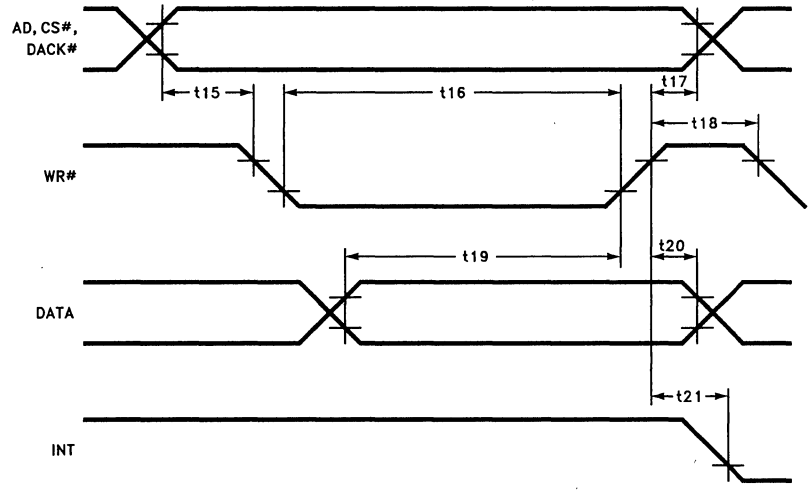
290474-16

Host Read Cycles

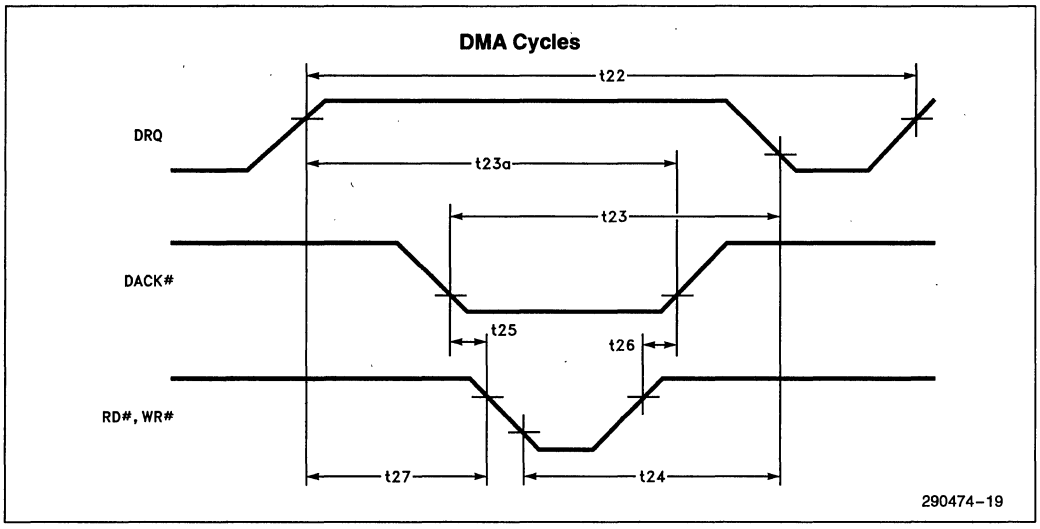


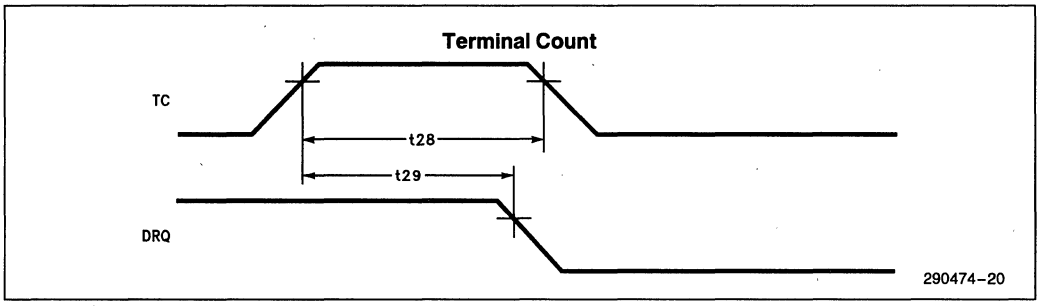
290474-17

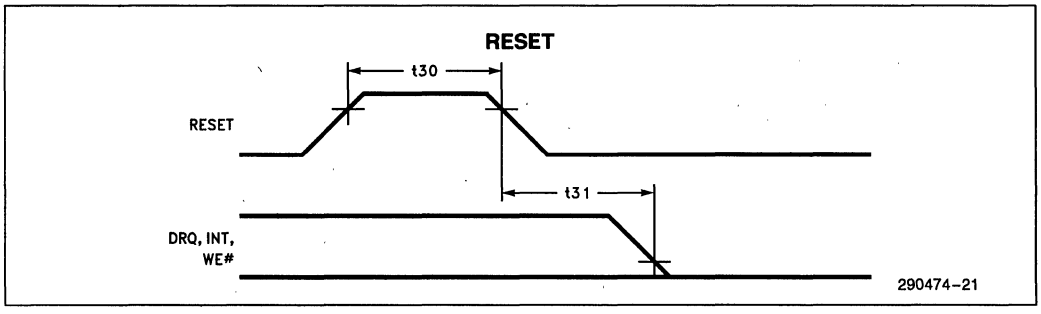
Host Write Cycles



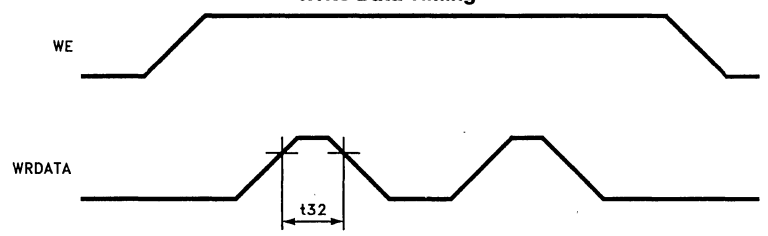
290474-18


DMA Cycles
t22
DRQ
t23a
t23
DACK#
t25
t26
RD#, WR#
t27
t24
290474-19


Terminal Count
TC
t28
t29
DRQ
290474-20

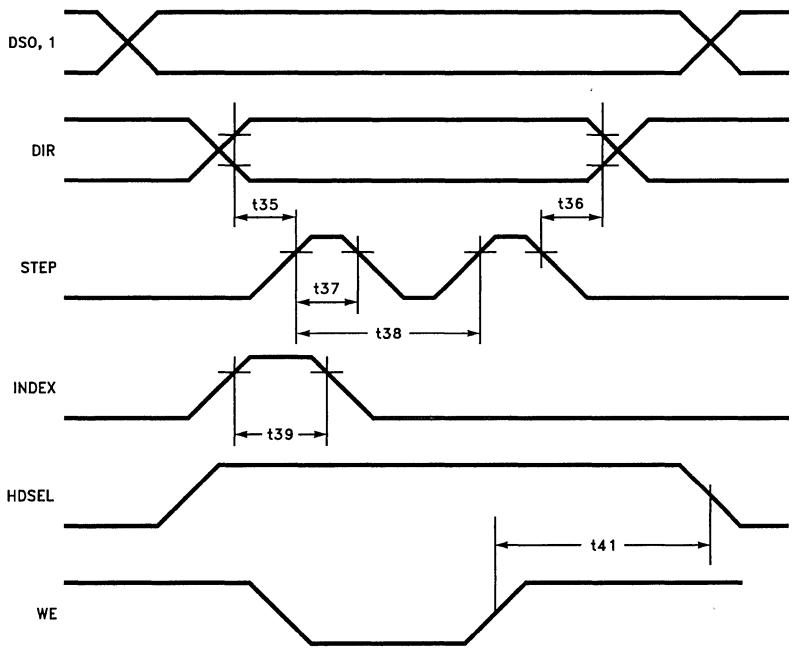

RESET
t30
RESET
t31
DRQ, INT,
WE#
290474-21

Write Data Timing



290474-22

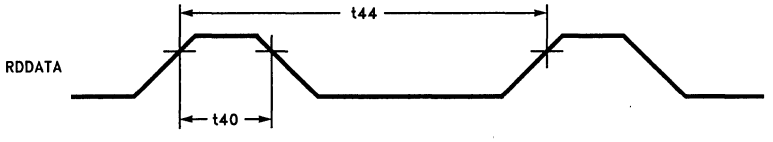
Drive Control



290474-23

NOTE:
For overlapped seeks, only one step pulse per drive section is issued. Non-overlapped seeks will issue all programmed step pulses.

Internal PLL

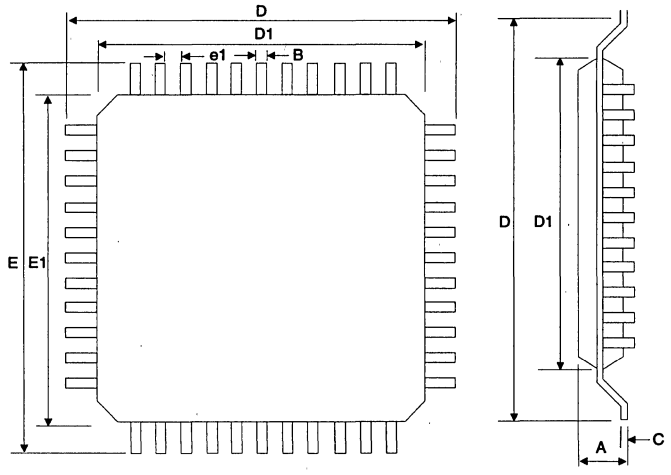


290474-24

12.1 Package Outline for the 44 Pin QFP Part

82078 addresses the current need of the smaller and thinner packages, for the current market. The size of the part is becoming increasingly important in the portable computer market. The QFP part considerably reduces the real estate consumed. The package outline with the appropriate dimensions are given below:

44-Lead Quad Flatpack Package



290474-25

Description	Symbol	44 Pin QFP Package	
		Nominal (mm)	Tolerance (mm)
Overall Height	A	2.10	±0.25
Stand Off	A1	0.35	±0.15
Lead Width	B	0.30	±0.10
Lead Thickness	C	0.15	±0.05
Terminal	D	12.4	±0.40
Long Side	D1	10.0	±0.10
Terminal	E	12.4	±0.40
Short Side	E1	10.0	±0.10
Lead Spacing	e1	0.80	±0.15
Lead Count	N	44	

13.0 REVISION HISTORY FOR THE 82078 44 PIN

The following list represents the key differences between version 002 and version 003 of the 82078 44 pin data sheet.

Section 2.1

Reference to register SRA removed. SRA is not available on the 44 pin 82078.

Section 2.1.2

DRIVE SEL 1 removed from DOR description. This bit is not available on the 44 pin version of the 82078.

Section 4.2

Clarification of PDOSC.

Section 4.4

Reference to register SRA removed. SRA is not available on the 44 pin 82078.

Section 5.2.3

Redundant information removed.

Section 5.2.4

Redundant information removed.

Section 6.3.2

Clarification of command.

Table 1.0

Reference to register SRA removed. SRA is not available on the 44 pin 82078.

Table 2-2 and Table 2-3

Table headings swapped to proper tables.

82078 64 PIN
CHMOS SINGLE-CHIP FLOPPY DISK CONTROLLER

- **Small Footprint and Low Height Packages**
- **Supports Standard 5.0V as well as Low Voltage 3.3V Platforms**
  - **— Selectable 3.3V and 5.0V Configuration**
  - **— 5.0V Tolerant Drive Interface**
- **Enhanced Power Management**
  - **— Application Software Transparency**
  - **— Programmable Powerdown Command**
  - **— Save and Restore Commands for Zero-Volt Powerdown**
  - **— Auto Powerdown and Wakeup Modes**
  - **— Two External Power Management Pins**
  - **— Consumes no Power when in Powerdown**
- **Integrated Analog Data Separator**
  - **— 250 Kbps**
  - **— 300 Kbps**
  - **— 500 Kbps**
  - **— 1 Mbps**
  - **— 2 Mbps**
- **Programmable Internal Oscillator**
- **Floppy Drive Support Features**
  - **— Drive Specification Command**
  - **— Media ID Capability Provides Media Recognition**
  - **— Drive ID Capability Allows the User to Recognize the Type of Drive**
  - **— Selectable Boot Drive**
  - **— Standard IBM and ISO Format Features**
  - **— Format with Write Command for High Performance in Mass Floppy Duplication**
- **Integrated Tape Drive Support**
  - **— Standard 1 Mbps/500 Kbps/ 250 Kbps Tape Drives**
  - **— New 2 Mbps Tape Drive Mode**
- **Perpendicular Recording Support for 4 MB Drives**
- **Integrated Host/Disk Interface Drivers**
- **Fully Decoded Drive Select and Motor Signals**
- **Programmable Write Precompensation Delays**
- **Addresses 256 Tracks Directly, Supports Unlimited Tracks**
- **16 Byte FIFO**
- **Single-Chip Floppy Disk Controller Solution for Portables and Desktops**
  - **— 100% PC AT* Compatible**
  - **— 100% PS/2* Compatible**
  - **— 100% PS/2 Model 30 Compatible**
  - **— Fully Compatible with Intel386™ SL Microprocessor SuperSet**
- **Integrated Drive and Data Bus Buffers**
- **Available in 64 Pin QFP Package**

The 82078, a 24 MHz crystal, a resistor package, and a device chip select implements a complete solution. All programmable options default to 82078 compatible values. The dual PLL data separator has better performance than most board level/discrete PLL implementations. The FIFO allows better system performance in multi-master (e.g., Microchannel, EISA).

The 82078 maintains complete software compatibility with the 82077SL/82077AA/8272A floppy disk controllers. It contains programmable power management features while integrating all of the logic required for floppy disk control. The power management features are transparent to any application software. There are two versions of 82078 floppy disk controllers, the 82078SL and 82078-1.

The 82078 is fabricated with Intel's advanced CHMOS III technology and is also available in a 44-lead QFP package.

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

October 1995
Order Number: 290475-004

82078 64 Pin CHMOS Single-Chip Floppy Disk Controller

CONTENTS

CONTENTS

CONTENTS

PAGE

CONTENTS

PAGE

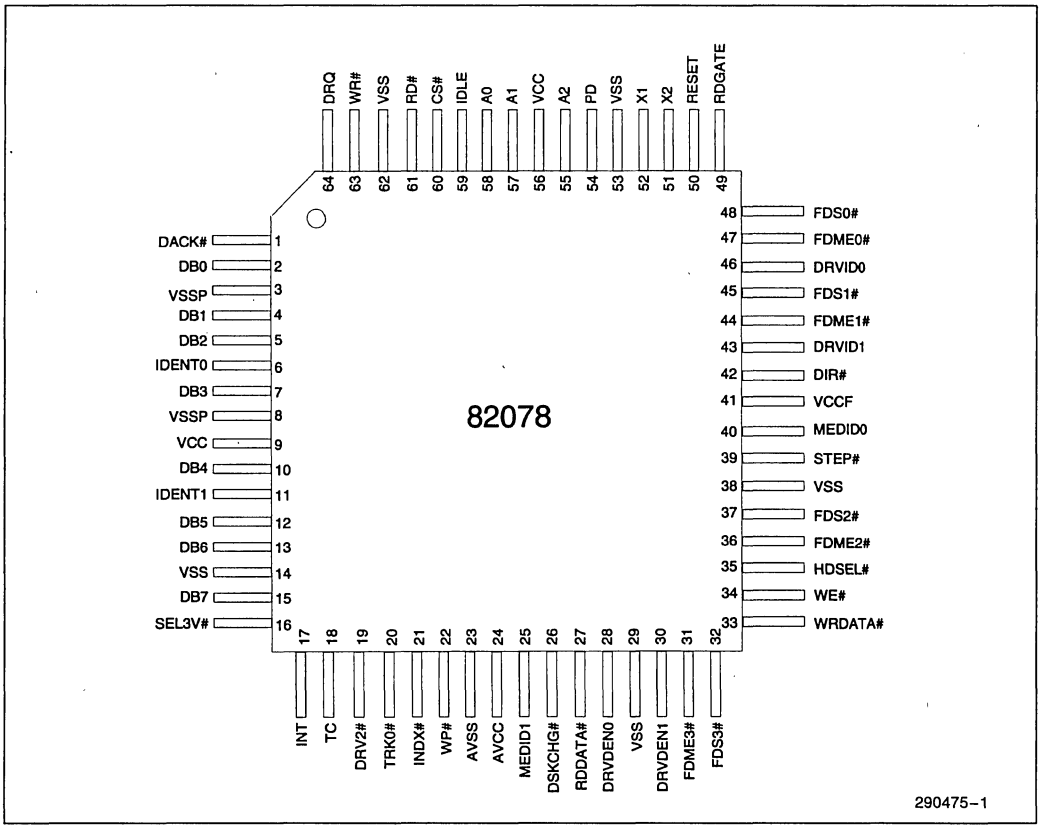


Figure 1-0. 82078 Pinout

Table 1-0. 82078 (64 Pin) Description

Symbol	Pin #	I/O	@ H/W Reset	Description
HOST INTERFACE				
RESET	50	I	N/A	**RESET:** A high level places the 82078 in a known idle state. All registers are cleared except those set by the Specify command.
A0 A1 A2	58 57 55	I	N/A	**ADDRESS:** Selects one of the host interface registers: **A2 A1 A0 Access Register** 0 0 0 R Status Register A SRA 0 0 1 R/W Status Register B SRB 0 1 0 R/W Digital Output Register DOR 0 1 1 R/W Tape Drive Register TDR 1 0 0 R Main Status Register MSR 1 0 0 W Data Rate Select Register DSR 1 0 1 R/W Data Register (FIFO) FIFO 1 1 0 Reserved 1 1 1 R Digital Input Register DIR 1 1 1 W Configuration Control Register CCR
CS#	60	I	N/A	**CHIP SELECT:** Decodes the base address range and qualifies RD# and WR#.

Table 1-0. 82078 (64 Pin) Description (Continued)

Symbol	Pin #	I/O	@ H/W Reset	Description
HOST INTERFACE (Continued)				
RD#	61	I	N/A	**READ:** Read control signal for data transfers from the floppy drive to the system.
WR#	63	I	N/A	**WRITE:** Write control signal for data transfers to the floppy drive from the system.
DRQ	64	O		**DMA REQUEST:** Requests service from a DMA controller. Normally active high, but will go to high impedance in AT and Model 30 modes when the appropriate bit is set in the DOR.
DACK#	1	I	N/A	**DMA ACKNOWLEDGE:** Control input that qualifies the RD#, WR# inputs in DMA cycles. Normally active low, but is disabled in AT and Model 30 modes when the appropriate bit is set in the DOR.
DB0 DB1 DB2 DB3 DB4 DB5 DB6 DB7	2 4 5 7 10 12 13 15	I/O		**DATA BUS:** 12 mA data bus.
IDENT0 IDENT1	6 11	I	N/A	**IDENTITY:** These inputs decode between the several operation modes available to the user. These pins have no effect on the DRVDEN pins. **IDENT0 IDENT1 INTERFACE** 1 1 AT mode 1 0 ILLEGAL 0 1 PS/2 mode 0 0 Model 30 **AT MODE:** Major options are: enables DMA gate logic, TC is active high, Status Register B is available based on a bit the powerdown command. **PS/2 MODE:** Major options are: no DMA gate logic, TC is active low, Status Registers A & B are available. **MODEL 30 MODE:** Major options are: enable DMA gate logic, TC is active high, Status Registers A & B are available.
INT	17	O		**INTERRUPT:** Signals a data transfer in non-DMA mode and when status is valid. Normally active high, but goes to high impedance when the appropriate bit is set in the DOR.
TC	18	I	N/A	**TERMINAL COUNT:** Control line from a DMA controller that terminates the current disk transfer. TC is effective only when qualified by DACK#. This input is active high in the AT, and Model 30 modes when the appropriate bit is set in the DOR.
X1 X2	52 51		N/A	**EXTERNAL CLOCK OR CRYSTAL:** Connection for a 24 MHz fundamental mode parallel resonant crystal. X1 can also be driven by an external clock (external oscillator) which can be either at 48 MHz or 24 MHz. If external oscillator is used then the PDOSC bit can be set to turn off the internal oscillator. Also, if a 48 MHz exernal oscillator is used then the CLK48 bit must be set in the enhanced CONFIGURE command.

Table 1-0. 82078 (64 Pin) Description (Continued)

Symbol	Pin #	I/O	@ H/W Reset	Description
POWER MANAGEMENT				
SEL3V#	16	I	N/A	**SELECT 3.3V:** This is a control pin that is used to select between 3.3V operation and 5.0V operation. This is an active low signal and selects 3.3V mode of operation when tied to ground.
PD	54	O		**POWERDOWN:** This pin is active high whenever the part is in powerdown state, either via DSR POWERDOWN bit or via the Auto Powerdown Mode. This pin can be used to disable an external oscillator's output.
IDLE	59	O		**IDLE:** This pin indicates that the part is in the IDLE state and can be powered down. IDLE state is defined as MSR = 80H, INT = 0, and the head being "unloaded" (as defined in Section 4.0, Power Management Features). Whenever the part is in this state, IDLE pin is active high. If the part is powered down by the Auto Powerdown Mode, IDLE pin is set high and if the part is powered down by setting the DSR POWERDOWN bit, IDLE pin is set low.
PLL SECTION				
RDDATA#	27	I	N/A	**READ DATA:** Serial data from the floppy disk.
RDGATE	49	O		**READ GATE:** This signal is basically used for diagnostic purposes.
DISK CONTROL				
DRV2#	19	I	N/A	**DRIVE2:** This is an active low signal that indicates whether a second drive is installed and is reflected in SRA.
TRK0#	20	I	N/A	**TRACK0:** This is an active low signal that indicates that the head is on track 0.
INDX#	21	I	N/A	**INDEX:** This is an active low signal that indicates the beginning of the track.
WP#	22	I	N/A	**WRITE PROTECT:** This is an active low signal that indicates whether the floppy disk in the drive is write protected.
MEDID1 MEDID0	25 40	I	N/A	**MEDIA ID:** These are active high signals that are output from the drive to indicate the density type of the media installed in the floppy drive. These should be tied low if not being used.
DSKCHG#	26	I	N/A	**DISK CHANGE:** This is an input from the floppy drive reflected in the DIR.
DRVDEN0 DRVDEN1	28 30	O		**DRIVE DENSITY:** These signals are used by the floppy drive to configure the drive for the appropriate media.
FDME3# FDME2# FDME1# FDME0#	31 36 44 47	O		**FLOPPY DRIVE MOTOR ENABLE:** Decoded motor enables for drives 0 to 3. The motor enable pins are directly controlled via the DOR and are a function of the mapping based on BOOTSEL bits in the TDR.
FDS3# FDS2# FDS1# FDS0#	32 37 45 48	O		**FLOPPY DRIVE SELECT:** Decoded floppy drive selects for drives 0 to 3. These outputs are decoded from the select bits in the DOR and are a function of the mapping based on BOOTSEL bits in the TDR.
WRDATA#	33	O		**WRITE DATA:** MFM serial data to the drive. Precompensation value is selectable through software.

Table 1-0. 82078 (64 Pin) Description (Continued)

Symbol	Pin #	I/O	@ H/W Reset	Description
DISK CONTROL (Continued)				
WE#	34	O		**WRITE ENABLE:** Floppy drive control signal that enables the head to write onto the floppy disk.
HDSEL#	35	O		**HEAD SELECT:** Selects which side of the floppy disk is to be used for the corresponding data transfer. It is active low and an active level selects head 1, otherwise it defaults to head 0.
STEP#	39	O		**STEP:** Supplies step pulses to the floppy drive to move the head between tracks.
DIR#	42	O		**DIRECTION:** It is an active low signal which controls the direction the head moves when a step signal is present. The head moves inwards towards the center if this signal is active.
DRVID0 DRVID1	46 43	I	N/A	**DRIVE ID:** These signals are input from the floppy drive and indicate the type of drive being used. These should be tied low if not being used.
POWER AND GROUND SIGNALS				
V_{CCF}	41		N/A	**VOLTAGE:** +5V for 5V floppy drive and 3.3V for 3.3V floppy drive.*
V_{CC}	9 56		N/A	**VOLTAGE:** +5V or 3.3V
V_{SSP}	3 8		N/A	**GROUND:** 0V
V_{SS}	14 29 38 53 62		N/A	**GROUND:** 0V
AV_{CC}	24		N/A	**ANALOG VOLTAGE**
AV_{SS}	23		N/A	**ANALOG GROUND**

***NOTE:**
The digital power supply V_{CC} and the analog power supply AV_{CC} should either be the same or regulated to be within 0.1V of either.

1.0 INTRODUCTION

The 82078, a 24 MHz (or 48 MHz) oscillator, a resistor package and a chip select implement a complete design. The power management features of the 82078 are transparent to application software, the 82078 seems awake to the software even in power-down mode. All drive control signals are fully decoded and have 24 mA (12 mA @ 3.3V) drive buffers. Signals returned from the drive are sent through on-chip input buffers with hysteresis for noise immunity. The integrated analog data separator needs no external compensation of components, yet allows for wide motor variation with exceptionally low soft error rates. The microprocessor interface has 12 mA drive buffers on the data bus plus 100% hardware register compatibility for PC-AT and Microchannel systems. The 16-byte FIFO with programmable thresholds is extremely useful in multi-master systems (Micro-Channel, EISA) or systems with large bus latency.

The 82078 features:

- 3.3V operation
- Small QFP package
- 2 Mbps data rate for tape drives
- Register enhancements from the 82077SL

Several pin changes accommodate the reduced pin count (from the 68 pin 82077SL) and the added features. Functional compatibility refers to software transparency between 82077SL/AA and the 82078. The 64 pin part will implement a superset of the features required to support all platforms, but is not pin to pin compatible to the 82077SL.

The 82078SL is capable of operating at both 3.3V and 5.0V. The 82078-1 only operates at 5.0V but has an available 2 Mbps tape drive data rate. All other features are available on both parts.

Part Specification	3.3V	5.0V	2 Mbps Data Rate
82078SL	X	X	
82078-1		X	X

Figure 1-1 is a block diagram of the 82078.

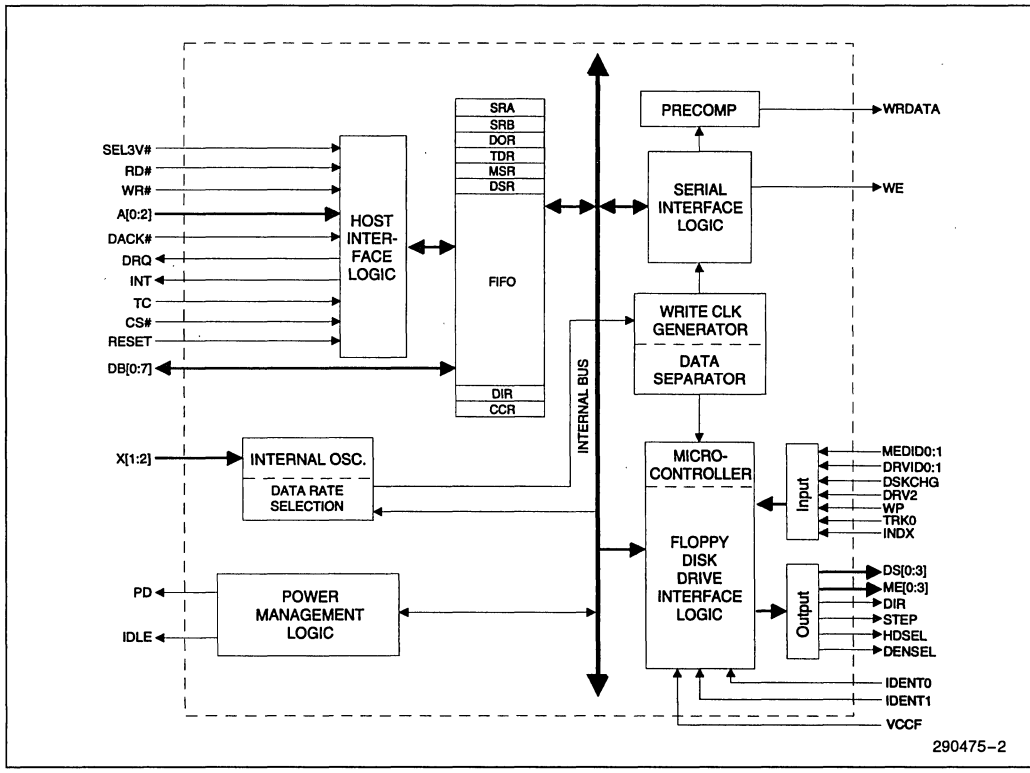


Figure 1-1. 82078 Block Diagram

2.0 MICROPROCESSOR INTERFACE

The interface consists of the standard asynchronous signals: RD#, WR#, CS#, A0–A2, INT, DMA control and a data bus. The address lines select between configuration registers, the FIFO and control/status registers. This interface can be switched between PC AT, Model 30, or PS/2 normal modes. The PS/2 register sets are a superset of the registers found in a PC-AT.

2.1 Status, Data and Control Registers

As shown below, the base address range is supplied via the CS# pin. For PC-AT or PS/2 designs, the primary and secondary address ranges are 3F0 Hex to 3F7 Hex and 370 Hex to 377 Hex respectively.

A2	A1	A0	Access Type	Register	
0	0	0	R	Status Register A	SRA
0	0	1	R/W	Status Register B	SRB
0	1	0	R/W	Digital Output Register	DOR
0	1	1	R/W	Tape Drive Register	TDR
1	0	0	R	Main Status Register	MSR
1	0	0	W	Data Rate Select Register	DSR
1	0	1	R/W	Data (First In First Out)	FIFO
1	1	0		Reserved	
1	1	1	R	Digital Input Register	DIR
1	1	1	W	Configuration Control Register	CCR

In the following sections, the various registers are shown in their powerdown state. The "UC" notation stands for a value that is returned without change from the active mode. The notation "*" means that the value is reflecting the required status (for powerdown). "n/a" means not applicable. "X" indicates that the value is undefined.

2.1.1 STATUS REGISTER A (SRA, PS/2 MODE)

This register is read-only and monitors the state of the interrupt pin and several disk interface pins. This register is part of the register set, and is not accessible in PC-AT mode.

This register can be accessed during powerdown state without waking up the 82078 from its powerdown state.

Bits	7	6	5	4	3	2	1	0
Function	INT PENDING	DRV2#	STEP	TRK0#	HDSEL	INDX#	WP#	DIR
H/W Reset State	0	DRV2#	0	TRK0#	0	INDX#	WP#	0
Auto PD State	0*	UC	0*	1	0*	1	1	0*

The INT PENDING bit is used by software to monitor the state of the 82078 INTERRUPT pin. By definition, the INT PENDING bit is low in powerdown state. The bits reflecting the floppy disk drive input pins (TRK0, INDEX, and WP) are forced inactive. Floppy disk drive outputs (HDSEL, STEP, and DIR) also go to their inactive, default state.

As a read-only register, there is no default value associated with a reset other than some drive bits will change with a reset. The INT PENDING, STEP, HDSEL, and DIR bits will be low after reset.

2.1.2 STATUS REGISTER A (SRA, MODEL 30 MODE)

Bits	7	6	5	4	3	2	1	0
Function	INT PENDING	DRQ	STEP F/F	TRK0	HDSEL#	INDX#	WP	DIR#
H/W Reset State	0	0	0	TRK0	1	INDX#	WP	1
Auto PD State	0*	0*	0	0	1*	0	0	1*

This register has the following changes in PS/2 Model 30 Mode. Disk interface pins (Bits 0, 1, 2, 3, and 4) are inverted from PS/2 Mode. The DRQ bit monitors the status of the DMA Request pin. The STEP bit is latched with the Step output going active and is cleared with a read to the DIR register, Hardware or Software RESET.

The DRQ bit is low by definition for 82078 to be in powerdown. The bits reflecting the floppy disk drive input pins (TRK0, INDEX, and WP) are forced to reflect an inactive state. The floppy disk drive outputs (HDSEL, STEP, and DIR) also go to their inactive, default state.

2

2.1.3 STATUS REGISTER B (SRB, ENHANCED AT/EISA)

In the AT/EISA mode the SRB is made available whenever the EREG EN bit in the auto powerdown command is set. The register functionality is defined as follows (bits 7 through 3 are reserved):

PD and IDLE reflect the values on the corresponding pins. The signal on the IDLE pin can be masked by setting IDLEMSK bit high in this register. The IDLE bit will remain unaffected. Since some systems will use the IDLE pin to provide interrupt to the SMM power management, its disabling allows less external interrupt logic and reduction in board space. Only hardware reset will clear the IDLEMSK bit to zero.

When the IDLEMSK bit is set, the user cannot distinguish between auto powerdown and DSR powerdown (i.e., by using the IDLE pin).

IDLEMSK	IDLE (pin)
0	unmasked
1	masked

SRB								
Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R	RSVD	RSVD	RSVD	RSVD	RSVD	IDLEMSK	PD	IDLE
H/W Reset	X	X	X	X	X	0	PD	IDLE
Auto PD	X	X	X	X	X	UC	UC	UC
W	0	0	0	0	0	IDLEMSK	RSVD	RSVD
H/W Reset	n/a	n/a	n/a	n/a	n/a	0	n/a	n/a
Auto PD	n/a	n/a	n/a	n/a	n/a	UC	n/a	n/a

2.1.4 STATUS REGISTER B (SRB, PS/2 MODE)

Bits	7	6	5	4	3	2	1	0
Function	1	1	DRIVE SEL 0	WRDATA TOGGLE	RDDATA TOGGLE	WE	MOT EN1	MOT EN2
H/W Reset State	1	1	0	0	0	0	0	0
Auto PD State	1	1	UC	0	0	0*	0	0

As the only drive input, RDATA TOGGLE's activity reflects the level as seen on the cable.

The two TOGGLE bits do not read back the state of their respective pins directly. Instead, the pins drive a Flip/Flop which produces a wider and more reliable read pulse. Bits 6 and 7 are undefined and always return to a 1.

After any reset, the activity on the TOGGLE pin is cleared. Drive select and Motor bits cleared by the RESET pin and not software resets.

2.1.5 STATUS REGISTER B (SRB, MODEL 30 MODE)

Bits	7	6	5	4	3	2	1	0
Function	DRV2#	DS1#	DS0#	WRDATA F/F	RDDATA F/F	WE F/F	DS3#	DS2#
H/W Reset State	DRV2#	1	1	0	0	0	1	1
Auto PD State	UC	UC	UC	0	0	0*	UC	UC

This register has the following changes in Model 30 Mode. Bits 0, 1, 5, and 6 return the decoded value of the Drive Select bits in the DOR register. Bits 2, 3, and 4 are set by their respective active going edges and are cleared by reading the DIR register. The WRDATA bit is triggered by raw WRDATA signals and is not gated by WE. Bits 2, 3, and 4 are cleared to low level by either Hardware or Software RESET.

2.1.6 DIGITAL OUTPUT REGISTER (DOR)

The Digital Output Register contains the drive select and motor enable bits, a reset bit and a DMA GATE# bit.

Bits	7	6	5	4	3	2	1	0
Function	MOT EN3	MOT EN2	MOT EN1	MOT EN0	DMA GATE#	RESET#	DRIVE SEL1	DRIVE SEL2
H/W Reset State	0	0	0	0	0	0	0	0
Auto PD State	0*	0*	0*	0*	UC	1*	UC	UC

The MOT ENx bits directly control their respective motor enable pins (FDME0–3). The DRIVE SELx bits are decoded to provide four drive select lines and only one may be active at a time. Standard programming practice is to set both MOT ENx and DRIVE SELx bits at the same time.

Table 2-1 lists a set of DOR values to activate the drive select and motor enable for each drive.

Table 2-1. Drive Activation Value

Drive	DOR Value
0	1CH
1	2DH
2	4EH
3	8FH

The DMAGATE# bit is enabled only in PC-AT and Model 30 Modes. If DMAGATE# is set low, the INT and DRQ outputs are tri-stated and the DACK# and TC inputs are disabled. DMAGATE# set high will enable INT, DRQ, TC, and DACK# to the system. In PS/2 Mode DMAGATE# has no effect upon INT, DRQ, TC, or DACK# pins, they are always active.

The DOR reset bit and the Motor Enable bits have to be inactive when the 82078 is in powerdown. The DMAGATE# and DRIVE SEL bits are unchanged. During powerdown, writing to the DOR does not awaken the 82078 with the exception of activating any of the motor enable bits. Setting the motor enable bits active (high) will wake up the part.

This RESET# bit clears the basic core of the 82078 and the FIFO circuits when the LOCK bit is set to "0" (see Section 5.3.2 for LOCK bit definitions). Once set, it remains set until the user clears this bit. This bit is set by a chip reset and the 82078 is held in a reset state until the user clears this bit. The RESET# bit has no effect upon the register.

2.1.7 TAPE DRIVE REGISTER (TDR AT/EISA, PS/2, MODEL 30)

Bits	7	6	5	4	3	2	1	0
Function	—	—	—	—	—	—	TAPE SEL1	TAPE SEL0
H/W Reset State	—	—	—	—	—	—	0	0
Auto PD State	—	—	—	—	—	—	UC	UC

(—) means these bits are not writable and remain tri-stated if read.

This register allows the user to assign tape support to a particular drive during initialization. Any future references to that drive number automatically invokes tape support. Hardware reset clears this register; software resets have no effect. By default, the tape select bits are hardware RESET to zeros, making Drive 0 not available for tape support.

2.1.8 ENHANCED TAPE DRIVE REGISTER (TDR, AT, PS/2, MODEL 30, EREG EN = 1)

In the PS/2 and Model 30 mode and AT/EISA mode the extended TDR is made available only when the EREG EN bit is set, otherwise the bits are tri-stated. The register functionality is defined as follows:

TDR								
Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R	MEDID1	MEDID0	DRVID1	DRVID0	BOOTSEL1	BOOTSEL0	TAPESEL1	TAPESEL0
H/W Reset	MEDID1	MEDID0	DRVID1	DRVID0	0	0	0	0
Auto PD	UC	UC	UC	UC	UC	UC	UC	UC
W	0	0	0	0	BOOTSEL1	BOOTSEL0	TAPESEL1	TAPESEL0
H/W Reset	n/a	n/a	n/a	n/a	0	0	0	0
Auto PD	n/a	n/a	n/a	n/a	BOOTSEL1	BOOTSEL0	TAPESEL1	TAPESEL0

MEDID1, MEDID0 reflect the values on the respective pins. Similarly, the DRVID0, DRVID1 reflect the values on the DRVID1 and DRVID0 pins.

The TAPESEL1, TAPESEL0 functionality is retained as defined in the non-enhanced TDR, except that the application of boot drive restriction (boot drive cannot be a tape drive) depends on what drive selected is by the BOOTSEL1, BOOTSEL0 bits.

The BOOTSEL1, BOOTSEL0 are not reset by software resets and are decoded as shown below. These bits allow for reconfiguring the boot up drive and only reset by hardware reset. A drive can be enabled by remapping the internal DS0 and ME0 to one of the other drive select and motor enable lines (Refer to "Selectable Boot Drives" in the Design applications chapter). Once a non-default value for BOOTSEL1 and BOOTSEL0 is selected, all programmable bits are virtual designations of drives, i.e., it is the user's responsibility to know the mapping scheme detailed in the following table.

BOOTSEL1	BOOTSEL0	Mapping:
0	0	DS0 → FDS0, ME0 → FDME0 DS1 → FDS1, ME1 → FDME1 DS2 → FDS2, ME2 → FDME2
0	1	DS0 → FDS1, ME0 → FDME1 DS1 → FDS0, ME1 → FDME0 DS2 → FDS2, ME2 → FDME2
1	0	DS0 → FDS2, ME0 → FDME2 DS1 → FDS1, ME1 → FDME1 DS2 → FDS0, ME2 → FDME0
1	1	Reserved

2.1.9 DATARATE SELECT REGISTER (DSR)

Bits	7	6	5	4	3	2	1	0
Function	S/W RESET	POWER DOWN	PDOSC	PRE COMP2	PRE COMP1	PRE COMP0	DRATE SEL1	DRATE SEL0
H/W Reset State	0	0	0	0	0	0	1	0
Auto PD State	S/W RESET	POWER DOWN	PDOSC	PRE COMP2	PRE COMP1	PRE COMP0	DRATE SEL1	DRATE SEL0

This register ensures backward compatibility with the 82072 floppy controller and is write-only. Changing the data rate changes the timings of the drive control signals. To ensure that drive timings are not violated when changing data rates, choose a drive timing such that the fastest data rate will not violate the timing.

The PDOSC bit is used to implement crystal oscillator power management. The internal oscillator in the 82078 can be programmed to be either powered on or off via PDOSC. This capability is independent of the chip's powerdown state. Auto powerdown mode and powerdown via POWERDOWN bit has no effect over the power state of the oscillator.

In the default state the PDOSC bit is low and the oscillator is powered up. When this bit is programmed to a one, the oscillator is shut off. Hardware reset clears this bit to a zero. Neither of the software resets (via DOR or DSR) have any effect on this bit. Note, PDOSC should only be set high when the part is in the powerdown state, otherwise the part will not function correctly and must be hardware reset once the oscillator has turned back on and stabilized. Setting the PDOSC bit has no effect on the clock input to the 82078 (the X1 pin). The clock input is separately disabled when the part is powered down. The SAVE command checks the status of PDOSC, however, the RESTORE command will not restore the bit high.

S/W RESET behaves the same as DOR RESET except that this reset is self cleaning.

POWERDOWN bit implements direct powerdown. Setting this bit high will put the 82078 into the powerdown state regardless of the state of the part. The part is internally reset and then put into powerdown. No status is saved and any operation in progress is aborted. A hardware or software reset will exit the 82078 from this powerdown state.

PRECOMP 0-2 adjusts the WRDATA output to the disk to compensate for magnetic media phenomena known as bit shifting. The data patterns that are susceptible to bit shifting are well understood and the 82078 compensates the data pattern as it is written to the disk. The amount of pre-compensation is dependent upon the drive and media, but in most cases the default value is acceptable.

Table 2-2. Precompensation Delays

PRECOMP	Precompensation Delays	
DSR[432]	x1 @ 24 MHz	x1 @ 48 MHz if CLK48 = 1, enabled only @ 2 Mbps if CLK48 = 0, enabled at all data rates
111	0.00 ns–disabled	
001	41.67	20.84
010	83.34	41.67
011	125.00	62.5
100	166.67	83.34
101	208.33	104.17
110	250.00	125
000	DEFAULT	

Table 2-3. Default Precompensation Delays

Data Rate	Precompensation Delays (ns)
2 Mbps	20.84
1 Mbps	41.67
0.5 Mbps	125
0.3 Mbps	125
0.25 Mbps	125

The 82078 starts pre-compensating the data pattern starting on Track 0. The CONFIGURE command can change the track that pre-compensating starts on. Table 2-2 lists the pre-compensation values that can be selected and Table 2-3 lists the default pre-compensation values. The default value is selected if the three bits are zeroes.

DRATE 0-1 select one of the four data rates as listed in Table 2-4. The default value is 250 Kbps after a "Hardware" reset. Other "Software" Resets do not affect the DRATE or PRECOMP bits.

Table 2-4. Data Rates

DRATESEL1	DRATESEL0	DATA RATE
1	1	1 Mbps
0	0	500 Kbps
0	1	300 Kbps
1	0	250 Kbps

2.1.10 MAIN STATUS REGISTER (MSR)

Bits	7	6	5	4	3*	2	1	0
Function	RQM	DIO	NON DMA	CMD BSY	DRV3 BUSY	DRV2 BUSY	DRV1 BUSY	DRV0 BUSY
H/W Reset State	0	X	X	X	X	X	X	X
Auto PD State	1	0	0	0	0	0	0	0

The Main Status Register is a read-only register and is used for controlling command input and result output for all commands.

RQM—Indicates that the host can transfer data if set to 1. No access is permitted if set to a 0.

DIO—Indicates the direction of a data transfer once RQM is set. A 1 indicates a read and a 0 indicates a write is required.

NON-DMA—This mode is selected in the SPECIFY command and will be set to a 1 during the execution phase of a command. This is for polled data transfers and helps differentiate between the data transfers and the reading of result bytes.

COMMAND BUSY—This bit is set to a one when a command is in progress. This bit goes active after the command byte has been accepted and goes inactive at the end of the results phase. If there is no result phase (SEEK, RECALIBRATE commands), this bit returns to a 0.

DRV x BUSY—These bits are set to ones when a drive is in the seek portion of a command, including seeks and recalibrates.

Some example values of the MSR are:

- MSR = 80H; The controller is ready to receive a command.
- MSR = 90H; executing a command or waiting for the host to read status bytes (assume DMA mode).
- MSR = D0H; waiting for the host to write status bytes.

2.1.11 FIFO (DATA)

All command parameter information and disk data transfers go through the FIFO. The FIFO is 16 bytes in size and has programmable threshold values. Data transfers are governed by the RQM and DIO bits in the Main Status Register.

The FIFO defaults to an 8272A compatible mode after a "Hardware" reset (Reset via pin 32). "Software" Resets (Reset via DOR or DSR register) can also place the 82078 into 8272A compatible mode if the LOCK bit is set to "0" (See the definition of the LOCK bit), maintaining PC-AT hardware compatibility. The default values can be changed through the CONFIGURE command (enable full FIFO operation with threshold control). The advantage of the FIFO is that it allows the system a larger DMA latency without causing a disk error. Table 2-5 gives several examples of the delays with a FIFO. The data is based upon the following formula:

$$\text{Threshold\#} \times 1/\text{DATA RATE} \times 8 - 1.5\ \mu s = \text{DELAY}$$

Table 2-5. FIFO Threshold Examples

FIFO Threshold Examples	Maximum Delay to Servicing at 1 Mbps Data Rate
1 byte	$1 \times 8\ \mu s - 1.5\ \mu s = 6.5\ \mu s$
2 bytes	$2 \times 8\ \mu s - 1.5\ \mu s = 14.5\ \mu s$
8 bytes	$8 \times 8\ \mu s - 1.5\ \mu s = 62.5\ \mu s$
15 bytes	$15 \times 8\ \mu s - 1.5\ \mu s = 118.5\ \mu s$

FIFO Threshold Examples	Maximum Delay to Servicing at 500 Kbps Data Rate
1 byte	$1 \times 16\ \mu s - 1.5\ \mu s = 14.5\ \mu s$
2 bytes	$2 \times 16\ \mu s - 1.5\ \mu s = 30.5\ \mu s$
8 bytes	$8 \times 16\ \mu s - 1.5\ \mu s = 126.5\ \mu s$
15 bytes	$15 \times 16\ \mu s - 1.5\ \mu s = 238.5\ \mu s$

At the start of a command, the FIFO action is always disabled and command parameters must be sent based upon the RQM and DIO bit settings. As the 82078 enters the command execution phase, it clears the FIFO of any data to ensure that invalid data is not transferred. An overrun or underrun will terminate the current command and the transfer of data. Disk writes will complete the current sector by generating a 00 pattern and valid CRC.

2.1.12 DIGITAL INPUT REGISTER (DIR, PC-AT MODE)

This register is read only in all modes. In PC-AT mode only bit 7 is driven, all other bits remain tri-stated.

Bits	7	6	5	4	3	2	1	0
Function	DSK CHG	—	—	—	—	—	—	—
H/2 Reset State	DSK CHG	—	—	—	—	—	—	—
Auto PD State	0	—	—	—	—	—	—	—

(—) means these bits are tri-stated when read.

DSKCHG monitors the pin of the same name and reflects the opposite value seen on the disk cable. The DSKCHG bit is forced inactive along with all the inputs from the floppy disk drive. All the other bits remain tri-stated.

2.1.13 DIGITAL INPUT REGISTER (DIR, PS/2 MODE)

DIR								
R/W	7	6*	5*	4*	3	2	1	0
R	DSK CHG	IDLE	PD	IDLEMSK	1	DRATE SEL1	DRATE SEL0	HIGH DENS#
H/W Reset	DSK CHG	1	1	1	1	1	0	1
Auto PD	0	1	1	UC	1	UC	UC	UC

(*) These bits are only available when PS2 STAT = 1: Bits 5 and 6 show the status of PD (powerdown) and IDLE respectively. Bit 4 shows the status of IDLEMSK, this bit disables the IDLE pin when active.

Bit 3 returns a value of "1", and the DRATE SEL1–0 return the value of the current data rate selected (see Table 2-4 for values).

HIGHDENS# is low whenever the 500 Kbps or 1 Mbps data rates are selected. It is high when either 250 Kbps, 300 Kbps, or 2 Mbps is selected.

The DSKCHG bit is forced inactive along with all the inputs from the floppy disk drive. All the other bits (as applicable) remain unchanged.

The Drive Specification Command modifies the DRATE SEL bits. Refer to Table 6-2 for a description.

2.1.14 DIGITAL INPUT REGISTER (DIR, MODEL 30 MODE)

Bits	7	6*	5*	4*	3	2	1	0
Function	DSK CHG#	IDLE	PD	IDLEMSK	DMA GATE#	NOPREC	DRATE SEL1	DRATE SEL0
H/W Reset State	N/A	0	0	0	0	0	1	0
Auto PD State	1	1	1	UC	UC	UC	UC	UC

(*) These bits are only available when PS2 STAT = 1: Bits 5 and 6 show the status of PD (powerdown) and IDLE respectively. Bit 4 shows the status of IDLEMSK, this bit disables the IDLE pin when active. Bit 7 (DSKCHG) is inverted in Model 30 Mode.

The DSKCHG# bit is forced inactive along with all the inputs from the floppy disk drive. All the other bits (as applicable) remain unchanged.

The Drive Specification Command modifies the DRATE SEL bits. Refer to Table 6-2 for information regarding the mapping of these bits.

Bit 3 reflects the value of DMAGATE# bit set in the DOR register.

Bit 2 reflects the value of NOPREC bit set in the CCR register.

2.1.15 CONFIGURATION CONTROL REGISTER (CCR, PC AT and PS/2 MODES)

This register sets the datarate and is write only.

Bits	7	6	5	4*	3	2	1	0
Function	—	—	—	IDLEMSK	—	—	DRATE SEL1	DRATE SEL0
H/W Restate State	—	—	—		—	—	1	0
Auto PD State	—	—	—	IDLEMSK	—	—	DRATE SEL1	DRATE SEL0

(*) This bit is enabled only when PS2 STAT = 1 (Powerdown Mode). Refer to the table in the Data Rate Select Register for values. Unused bits should be set to 0. IDLEMSK is not available in the CCR for PC AT mode. In PC AT, IDLEMSK is available in the SRB.

2.1.16 CONFIGURATION CONTROL REGISTER (CCR, MODEL 30 MODE)

Bits	7	6	5	4*	3	2	1	0
Function	—	—	—	IDLEMSK	—	NOPREC	DRATE SEL1	DRATE SEL0
H/W Reset State	—	—	—	0	—	0	1	0
Auto PD State	—	—	—	UC	—	UC	UC	UC

(*) This bit is enabled only when PS2 STAT = 1 (Powerdown Mode). NOPREC has no function, and is reset to "0" with a Hardware RESET only.

2.2 Reset

There are three sources of reset on the 82078; the RESET pin, a reset generated via a bit in the DOR and a reset generated via a bit in the DSR. All resets take the 82078 out of the powerdown state.

In entering the reset state, all operations are terminated and the 82078 enters an idle state. Activating reset while a disk write activity is in progress will corrupt the data and CRC.

On exiting the reset state, various internal registers are cleared, and the 82078 waits for a new command. Drive polling will start unless disabled by a new CONFIGURE command.

2.2.1 RESET PIN ("HARDWARE") RESET

The RESET pin is a global reset and clears all registers except those programmed by the SPECIFY command. The DOR Reset bit is enabled and must be cleared by the host to exit the reset state.

2.2.2 DOR RESET vs DSR RESET ("SOFTWARE" RESET)

These two resets are functionally the same. The DSR Reset is included to maintain 82072 compatibility. Both will reset the 82072 core which affects drive status information. The FIFO circuits will also be reset if the LOCK bit is a "0" (see definition of the LOCK bit). The DSR Reset clears itself automatically while the DOR Reset requires the host to manually clear it. DOR Reset has precedence over the DSR Reset. The DOR Reset is set automatically upon a pin RESET. The user must manually clear this reset bit in the DOR to exit the reset state.

The t30a specification in the A.C. Specifications gives the minimum amount of time that the DOR reset must be held active. This amount of time that the DOR reset must be held active is dependent upon the data rate. The 82078 requires that the DOR reset bit must be held active for at least 0.5 μs at 250 Kbps. This is less than a typical ISA I/O cycle time.

2.3 DMA Transfers

DMA transfers are enabled with the SPECIFY command and are initiated by the 82078 by activating the DRQ pin during a data transfer command. The FIFO is enabled directly by asserting DACK# and addresses need not be valid (CS# can be held inactive during DMA transfers).

3.0 DRIVE INTERFACE

The 82078 has integrated all of the logic needed to interface to a floppy disk or a tape drive which use floppy interface. All drive outputs have 24 mA drive capability and all inputs use a receive buffer with hysteresis. The internal analog data separator requires no external components, yet allows for an extremely wide capture range with high levels of read-data jitter, and ISV. The designer needs only to run the 82078 disk drive signals to the disk or tape drive connector.

3.1 Cable Interface

Generally, 5.25" drive uses open collector drivers and 3.5" drives (as used on PS/2) use totem-pole drivers. The output buffers on the 82078 do not change between open collector or totem-pole, they are always totem-pole.

DRVDEN0 and DRVDEN1 connect to pins 2 and 6 or 33 (on most disk drives) to select the data rate sent from the drive to the 82078. The polarity of DRVDEN0 and DRVDEN1 can be programmed through the Drive Specification command (see the command description for more information).

When the 82078SL is operating at 3.3V, the floppy drive interface can be configured to either 5.0V or 3.3V, via the V_{CCF} (pin 41). The drive interface follows the voltage level on V_{CCF}. A selectable drive interface allows the system designer the greatest flexibility when designing a low voltage system.

3.2 Data Separator

The function of the data separator is to lock onto the incoming serial read data. When lock is achieved, the serial front end logic of the chip is provided with a clock which is synchronized to the read data. The synchronized clock, called Data Window, is used to internally sample the serial data. One state of Data Window is used to sample the data portion of the bit cell, and the alternate state samples the clock portion. Serial to parallel conversion logic separates the read data into clock and data bytes.

To support reliable disk reads the data separator must track fluctuations in the read data frequency. Frequency errors primarily arise from two sources: motor rotation speed variation and instantaneous speed variation (ISV). A second condition, and one that opposes the ability to track frequency shifts is the response to bit jitter.

The internal data separator consists of two analog phase lock loops (PLLs) as shown in Figure 3-1. The two PLLs are referred to as the reference PLL and the data PLL. The reference PLL (the master PLL) is used to bias the data PLL (the slave PLL). The reference PLL adjusts the data PLL's operating point as a function of process, junction temperature and supply voltage. Using this architecture it was possible to eliminate the need for external trim components.

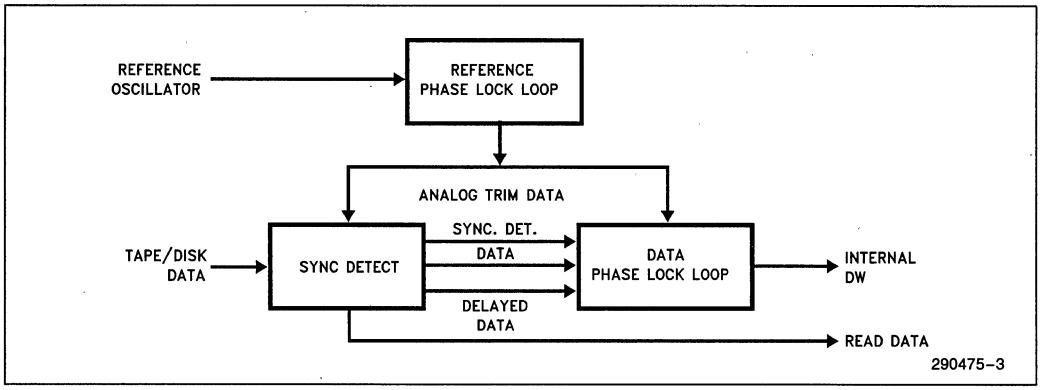


Figure 3-1. Data Separator Block Diagram

PHASE LOCK LOOP OVERVIEW

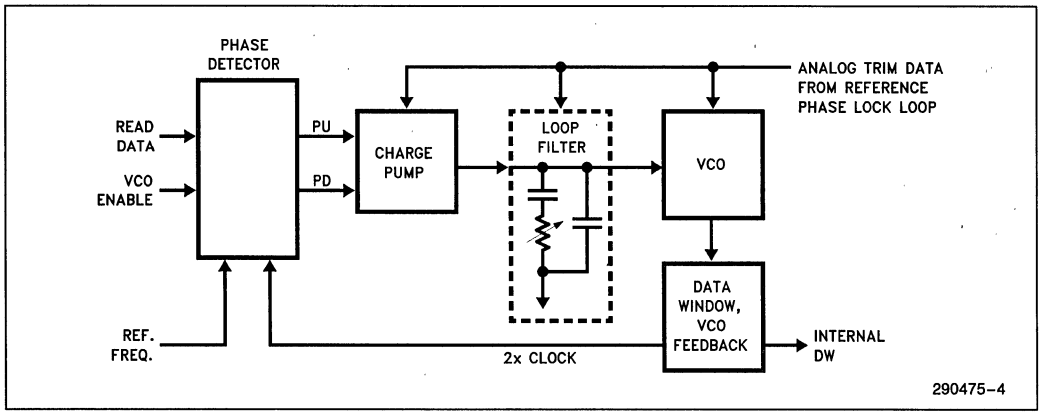


Figure 3-2. Data PLL

Figure 3-2 shows the data PLL. The reference PLL has control over the loop gain by its influence on the charge pump and the VCO. In addition the reference PLL controls the loop filter time constant. As a result the closed loop transfer function of the data PLL is controlled, and immune to the first order, to environmental factors and process variation.

Systems with analog PLLs are often very sensitive to noise. In the design of this data separator many steps were taken to avoid noise sensitivity problems. The analog section of the chip has a separate VSS pin (AVSS) which should be connected externally to a noise free ground. This provides a clean basis for VSS referenced signals. In addition many analog circuit features were employed to make the overall system as insensitive to noise as possible.

3.2.1 JITTER TOLERANCE

The jitter immunity of the system is dominated by the data PLL's response to phase impulses. This is measured as a percentage of the theoretical data window by dividing the maximum readable bit shift by a ¼ bit cell distance. For instance, if the maximum allowable bit shift is 300 ns for a 500 Kbps data stream, the jitter tolerance is 60%.

3.2.2 LOCKTIME (tLOCK)

The lock, or settling time of the data PLL is designed to be 64 bit times (8 sync bytes). The value assumes that the sync field jitter is 5% the bit cell or less. This level of jitter is realistic for a constant bit pattern. Intersymbol interference should be equal, thus nearly eliminating random bit shifting.

3.2.3 CAPTURE RANGE

Capture Range is the maximum frequency range over which the data separator will acquire phase lock with the incoming RDDATA signal. In a floppy disk environment, this frequency variation is composed of two components: drive motor speed error and ISV. Frequency is a factor which may determine the maximum level of the ISV (Instantaneous Speed Variation) component. In general, as frequency increases the allowed magnitude of the ISV component will decrease. When determining the capture range requirements, the designer should take the maximum amount of frequency error for the disk drive and double it to account for media switching between drives.

3.3 Write Precompensation

The write precompensation logic is used to minimize bit shifts in the RDDATA stream from the disk drive. The shifting of bits is a known phenomena of magnetic media and is dependent upon the disk media AND the floppy drive.

The 82078 monitors the bit stream that is being sent to the drive. The data patterns that require precompensation are well known. Depending upon the pattern, the bit is shifted either early or late (or not at all) relative to the surrounding bits. Figure 3-3 is a block diagram of the internal circuit.

The top block is a 13-bit shift register with the no delay tap being in the center. This allows 6 levels of early and late shifting with respect to nominal. The shift register is clocked at the main clock rate (24 MHz). The output is fed into 2 multiplexors one for early and one for late. A final stage of multiplexors combines the early, late and normal data stream back into one which is the WRDATA output.

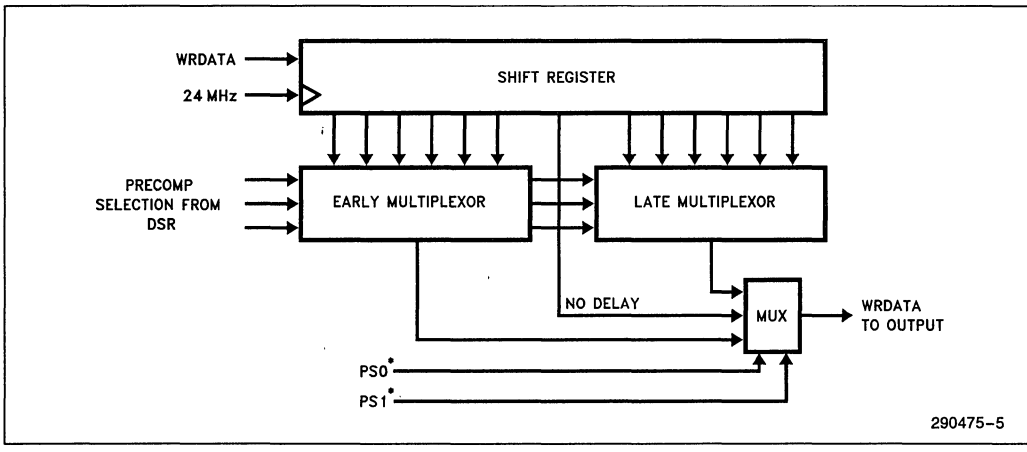


Figure 3-3. Precompensation Block Diagram

4.0 POWER MANAGEMENT FEATURES

The 82078 contains power management features that makes it ideal for design of portable personal computers. In addition to all of the power management features the 82078SL also operates at 3.3V. These features can be classified into power management of the part and that of the internal oscillator. The powerdown of the part is done independently of the internal oscillator in the 82078.

4.1 Power Management Scheme

The 82078 supports two powerdown modes, direct powerdown and automatic powerdown. Direct powerdown refers to direct action by the software to powerdown without dependence on external factors. Automatic powerdown results from 82078's monitoring of the current conditions according to a previously programmed mode. Any hardware reset disables the automatic POWERDOWN command, however, software resets have no effect on the command. The 82078 also supports powerdown of its internal crystal oscillator independent of the powerdown modes described above. By setting bit 5 (PDOSC) in the DSR register, the internal oscillator is turned off. This bit has sole control of the oscillator powerdown, allowing the internal clock to be turned off when an external oscillator is used.

4.2 3.3V Support for Portable Platforms

The portable market share of the personal computing market has increased significantly. To improve power conservation on portable platforms, designs are migrating from 5.0V to 3.3V. Intel's 82078SL allows designers to incorporate 3.3V floppy disk controller support in their systems. The 82078SL has a SEL3V# pin to allow selection of either 5.0V or 3.3V operation. In order to support the slower migration of floppy drives to 3.3V and allow system vendors to use standard 5.0V floppy drive inventory, the 82078SL accommodates a 5.0V tolerant floppy drive interface. This is achieved by changing the floppy drive's interface power supply, VCCF between 5.0V and 3.3V supplies. The 82078SL's 3.3V D.C. specification conforms to the JEDEC standard that describes the operating voltage levels for Integrated Circuits operating at 3.3V $\pm$0.3V. The 82077SL also maintains compatibility to 5.0V A.C. specifications.

4.3 Oscillator Power Management

The 82078 supports a built-in crystal oscillator that can be programmed to be either powered down or active, independent of the power state of the chip. This capability is implemented by the PDOSC bit in the DSR. When PDOSC is set low, the internal oscillator is on, it is off when the bit is high. Note, a DSR powerdown does not turn off the oscillator.

When the external oscillator is used, power can be saved by turning off the internal oscillator. If the internal oscillator is used, the oscillator may be powered up (even when the rest of the chip is powered off) allowing the chip to wake up quickly and in a stable state. It is recommended to keep the internal oscillator on even when in the powerdown state. The main reason for this is that the recovery time of the oscillator during wake up may take tens of milliseconds under the worst case, which may create problems with any sensitive application software. In a typical application the internal oscillator should be on unless the system goes into a power saving or standby mode (such a mode request would be made by a system time out or by a user). In this case, the system software would take over and must turn on the oscillator sufficiently ahead of awakening the part.

In the case of the external oscillators, the power up characteristics are similar. If the external source remains active during the time the 82078 is powered down, then the recovery time effect is minimized. The PD pin can be used to turn off the external source. While the PD pin is active, the 82078 does not require a clock source. However, when the PD pin is inactive, the clocking source, once it starts oscillating, must be completely stable to ensure that the 82078 operates properly.

4.4 Part Power Management

This section deals with the power management of the rest of the chip excluding the oscillator. This section explains powerdown modes and wake up modes.

4.4.1 DIRECT POWERDOWN

Direct powerdown is conducted via the POWERDOWN bit in the DSR register (bit 6). Programming this bit high will powerdown 82078. All status is lost if this type of powerdown mode is used. The part can exit powerdown from this mode via any hardware or software reset. This type of powerdown overrides the automatic powerdown. When the part is in automatic powerdown and the DSR powerdown is issued, the previous status of the part is lost and the 82078 resets to its default values.

4.4.2 AUTO POWERDOWN

Automatic powerdown is conducted via a "Powerdown Mode" command. There are four conditions required before the part will enter powerdown. All these conditions must be true for the part to initiate the powerdown sequence. These conditions follow:

1. The motor enable pins FDME[0:3] must be inactive.
2. The part must be idle; this is indicated by MSR = 80H and INT = 0 (INT may be high even if MSR = 80H due to polling interrupt).
3. The head unload timer (HUT, explained in the SPECIFY command) must have expired.
4. The auto powerdown timer must have timed out.

The command can be used to enable powerdown by setting the AUTO PD bit in the command to high. The command also provides a capability of programming a minimum power up time via the MIN DLY bit in the command. The minimum power up time refers to a minimum amount of time the part will remain powered up after being awakened or reset. An internal timer is initiated as soon as the auto powerdown command is enabled. The part is then powered down provided all the remaining conditions are met.

Any software reset will reinitialize the timer. Changing of data rate extends the auto powerdown timer by up to 10 ms, but only if the data rate is changed during the countdown.

Disabling the auto powerdown mode cancels the timers and holds the 82078 out of auto powerdown.

The IDLE pin can be masked via the IDLEMSK bit in Status Register B (Enhanced AT/EISA). When in PS/2 mode, the PS/2 STAT bit in the Powerdown Command can be set to enable PD and IDLE bits in the DIR register (bits 5 and 6) and IDLEMSK (bit 4) can be enabled.

4.4.3 WAKE UP MODES

This section describes the conditions for awakening the part from both direct and automatic powerdown. Power conservation or extension of battery life is the main reason power management is required. This means that the 82078 must be kept in powerdown state as long as possible and should be powered up as late as possible without compromising software transparency.

To keep the part in powerdown mode as late as possible implies that the part should wake up as fast as possible. However, some amount of time is required for the part to exit powerdown state and prepare the internal microcontroller to accept commands. Application software is very sensitive to such a delay and in order to maintain software transparency, the recovery time of the wake up process must be carefully controlled by the system software.

4.4.3.1 Wake Up from DSR Powerdown

If the 82078 enters the powerdown through the DSR powerdown bit, it must be reset to exit. Any form of software or hardware reset will serve, although DSR is recommended. No other register access will awaken the part, including writing to the DOR's motor enable (FDME[0:3]) bits.

If DSR powerdown is used when the part is in auto powerdown, the DSR powerdown will override the auto powerdown. However, when the part is awakened by a software reset, the auto powerdown command (including the minimum delay timer) will once again become effective as previously programmed. If the part is awakened via a hardware reset, the auto powerdown is disabled.

After reset, the part will go through a normal sequence. The drive status will be initialized. The FIFO mode will be set to default mode on a hardware reset or on a software reset if the LOCK command has not blocked it. Finally, after a delay, the polling interrupt will be issued.

4.4.3.2 Wake Up from Auto Powerdown

If the part enters the powerdown state through the auto powerdown mode, then the part can be awakened by reset or by appropriate access to certain registers.

If a hardware or software reset is used then the part will go through the normal reset sequence. If the access is through the selected registers, then the 82078 resumes operation as though it was never in powerdown. Besides activating the RESET pin or one of the software reset bits in the DOR or DSR, the following register accesses will wake up the part:

1. Enabling any one of the motor enable bits in the DOR register (reading the DOR does not awaken the part).
2. A read from the MSR register.
3. A read or write to the FIFO register.

Any of these actions will wake up the part. Once awake, 82078 will reinitiate the auto powerdown timer for 10 ms or 0.5 sec. (depending on the MIN DLY bit the auto powerdown command). When operating at 2 Mbps, the time is halved to 5 ms or 0.25 sec. depending on the MIN DLY bit. The part will powerdown again when all the auto powerdown conditions are satisfied.

4.5 Register Behavior

The register descriptions and their values in the powerdown state are listed in the Microprocessor Interface section. Table 4-1 reiterates the AT and PS/2 (including model 30) configuration registers available. It also shows the type of access permitted. In order to maintain software transparency, access to all the registers must be maintained. As Table 4-1 shows, two sets of registers are distinguished based on whether their access results in the part remaining in powerdown state or exiting it.

Table 4-1. 82078 Register Behavior

Address	Available Registers	Access	
	PC-AT	PS/2 (Model 30)	Permitted
Access to these registers DOES NOT wake up the part			
000	—	SRA	R
001	SRB (EREG EN = 1)	SRB	R/W
010	DOR*	DOR*	R/W
011	TDR	TDR	R/W
100	DSR*	DSR*	W
110	—	—	—
111	DIR	DIR	R
111	CCR	CCR	W
Access to these registers wake up the part			
100	MSR	MSR	R
101	FIFO	FIFO	R/W

*Writing to the DOR or DSR does not wake up the part, however, writing any of the motor enable bits or doing a software reset (either via DOR or DSR reset bits) will wake up the part.

Access to all other registers is possible without awakening the part. These registers can be accessed during powerdown without changing the status of the part. A read from these registers will reflect the true status as shown in the register description in Section 2.1. A write to the part will result in the part retaining the data and subsequently reflecting it when the part awakens. Accessing the part during powerdown may cause an increase in the power consumption by the part. The part will revert back to its low power mode when the access has been completed. None of the extended registers effect the behavior of the powerdown mode.

4.6 Pin Behavior

The 82078 is specifically designed for the portable PC systems in which the power conservation is a primary concern. This makes the behavior of the pins during powerdown very important.

The pins of 82078 can be divided into two major categories; system interface and floppy disk drive interface. The floppy disk drive pins are disabled such that no power will be drawn through the 82078 as a result of any voltage applied to the pin within the 82078's power supply range. The floppy disk drive interface pins are configurable by the FDI TRI bit in the auto powerdown command. When the bit is set the output pins of the floppy disk drive retain their original state. All other pins are either disabled or unchanged as depicted in Table 4-4. Most of the system interface pins are left active to monitor system accesses that may wake up the part.

4.6.1 SYSTEM INTERFACE PINS

Table 4-2 gives the state of the system interface pins in the powerdown state. Pins unaffected by powerdown are labeled "UC". Input pins are "DISABLED" to prevent them from causing currents internal to the 82078 when they have indeterminate input values.

Table 4-2. System Interface Pins

System Pins	State In Powerdown	System Pins	State In Powerdown
Input Pins		**Output Pins**	
CS#	UC	DRQ	UC (Low)
RD#	UC	INT	UC (Low)
WR#	UC	PD	HIGH
A[0:2]	UC	IDLE	High (Auto PD) Low (DSR PD)
DB[0:7]	UC	DB[0:7]	UC
RESET	UC		
IDENTn	UC		
DACK#	Disabled		
TC	Disabled		
X[1:2]	Programmable		

Two pins which can be used to indicate the status of the part are IDLE and PD. Table 4-3 shows how these pins reflect the 82078 status.

Table 4-3. 82078 Status Pins

PD	IDLE	MSR	Part Status
1	1	80H	Auto Powerdown
1	0	RQM = 1; MSR[6:0] = X	DSR Powerdown
0	1	80H	Idle
0	0	—	Busy

The IDLE pin indicates when the part is idle state and can be powered down. It is a combination of MSR equalling 80H, the head being unloaded and the INT pin being low. As shown in the table the IDLE pin will be low when the part is in DSR powerdown state. The PD pin is active whenever the part is in the powerdown state. It is active for either mode of powerdown. The PD pin can be used to turn off an external oscillator of other floppy disk drive interface hardware.

4.6.2 FDD INTERFACE PINS

The FDD interface "input" pins during powerdown are disabled or unchanged as shown in Table 4-4. The floppy disk drive "output" pins are programmable by the FDI TRI bit in the auto powerdown command. Setting of the FDI TRI bit in the auto powerdown command results in the interface retaining its normal state. When this bit is low (default state) all output pins in the FDD interface to the floppy disk drive itself are tri-stated. Pins used for local logic control or part programming are unaffected. Table 4-4 depicts the state of the floppy disk interface pins in the powerdown state (FDI TRI is low).

Table 4-4. 82078 FDD Interface Pins

FDD Pins	State In Powerdown	FDD Pins	State in Powerdown
Input Pins		Output Pins (FDI TRI = 0)	
RDDATA	Disabled	FDME[0:3]#	Tristated
WP	Disabled	FDS[0:3]#	Tristated
TRK0	Disabled	DIR#	Tristated
INDX#	Disabled	STEP#	Tristated
DRV2#	Disabled	WRDATA#	Tristated
DSKCHG#	Disabled	WE#	Tristated
		HDSEL#	Tristated
		DRVDEN[0:1]#	Tristated

5.0 CONTROLLER PHASES

For simplicity, command handling in the 82078 can be divided into three phases: Command, Execution and Result. Each phase is described in the following sections.

When there is no command in progress, the 82078 can be in idle, drive polling or powerdown state.

5.1 Command Phase

After a reset, the 82078 enters the command phase and is ready to accept a command from the host. For each of the commands, a defined set of command code bytes and parameter bytes has to be written to the 82078 before the command phase is complete (Please refer to Section 6.0 for the command descriptions). These bytes of data must be transferred in the order prescribed.

Before writing to the 82078, the host must examine the RQM and DIO bits of the Main Status Register. RQM, DIO must be equal to "1" and "0" respectively before command bytes may be written. RQM is set false by the 82078 after each write cycle until the received byte is processed. The 82078 asserts RQM again to request each parameter byte of the command, unless an illegal command condition is detected. After the last parameter byte is received, RQM remains "0", and the 82078 automatically enters the next phase as defined by the command definition.

The FIFO is disabled during the command phase to retain compatibility with the 8272A, and to provide for the proper handling of the "Invalid Command" condition.

5.2 Execution Phase

All data transfers to or from the 82078 occur during the execution phase, which can proceed in DMA or non-DMA mode as indicated in the SPECIFY command.

Each data byte is transferred by an INT or DRQ depending on the DMA mode. The CONFIGURE command can enable the FIFO and set the FIFO threshold value.

The following paragraphs detail the operation of the FIFO flow control. In these descriptions, (threshold) is defined as the number of bytes available to the 82078 when service is requested from the host, and ranges from 1 to 16. The parameter FIFOTHR which the user programs is one less, and ranges from 0 to 15.

A low threshold value (i.e., 2) results in longer periods of time between service requests, but requires faster servicing of the request, for both read and write cases. The host reads (writes) from (to) the FIFO until empty (full), then the transfer request goes inactive. The host must be very responsive to the service request. This is the desired case for use with a "fast" system.

A high value of threshold (i.e., 12) is used with a "sluggish" system by affording a long latency period after a service request, but results in more frequent service requests.

5.2.1 NON-DMA MODE, TRANSFERS FROM THE FIFO TO THE HOST

The INT pin and RQM bits in the Main Status Register are activated when the FIFO contains 16 (or set threshold) bytes, or the last bytes of a full sector transfer have been placed in the FIFO. The INT pin can be used for interrupt driven systems and RQM can be used for polled sytems. The host must respond to the request by reading data from the FIFO. This process is repeated until the last byte is transferred out of the FIFO, then 82078 deactivates the INT pin and RQM bit.

5.2.2 NON-DMA MODE, TRANSFERS FROM THE HOST TO THE FIFO

The INT pin and RQM bit in the Main Status Register are activated upon entering the execution phase of data transfer commands. The host must respond to the request by writing data into the FIFO. The INT pin and RQM bit remain true until the FIFO becomes full. They are set true again when the FIFO has (threshold) bytes remaining in the FIFO. The INT pin will also be deactivated if TC and DACK# both go inactive. The 82078 enters the result phase after the last byte is taken by the 82078 from the FIFO (i.e., FIFO empty condition).

5.2.3 DMA MODE, TRANSFERS FROM THE FIFO TO THE HOST

The 82078 activates the DRQ pin when the FIFO contains 16 (or set threshold) bytes, or the last byte of a full sector transfer has been placed in the FIFO. The DMA controller must respond to the request by reading data from the FIFO. The 82078 will deactivate the DRQ pin when the FIFO becomes empty. DRQ goes inactive after DACK# goes active for the last byte of a data transfer (or on the active edge of RD#, on the last byte, if no edge is present on DACK#). Note that DACK# and TC must overlap for at least 50 ns for proper functionality.

5.2.4 DMA MODE, TRANSFERS FROM THE HOST TO THE FIFO

The 82078 activates the DRQ pin when entering the execution phase of the data transfer commands. The DMA controller must respond by activating the DACK# and WR# pins and placing data in the FIFO. DRQ remains active until the FIFO becomes full. DRQ is again set true when the FIFO has (threshold) bytes remaining in the FIFO. The 82078 will also deactivate the DRQ pin when TC becomes true (qualified by DACK# by overlapping by 50 ns), indicating that no more data is required. DRQ goes inactive after DACK# goes active for the last byte of a data transfer (or on the active edge of WR# of the last byte, if no edge is present on DACK#).

5.2.5 DATA TRANSFER TERMINATION

The 82078 supports terminal count explicitly through the TC pin and implicitly through the underrun/overrun and end-of-track (EOT) functions. For full sector transfers, the EOT parameter can define the last sector to be transferred in a single or multisector transfer. If the last sector to be transferred is a partial sector, the host can stop transferring the data in mid-sector, and the 82078 will continue to complete the sector as if a hardware TC was received. The only difference between these implicit functions and TC is that they return "abnormal termination" result status. Such status indications can be ignored if they were expected.

Note that when the host is sending data to the FIFO of the 82078, the internal sector count will be complete when 82078 reads the last byte from its side of the FIFO. There may be a delay in the removal of the transfer request signal of up to the time taken for the 82078 to read the last 16 bytes from the FIFO. The host must tolerate this delay.

5.3 Result Phase

The generation of INT determines the beginning of the result phase. For each of the commands, a defined set of result bytes has to be read from the 82078 before the result phase is complete. (Refer to Section 6.0 on command descriptions.) These bytes of data must be read out for another command to start.

RQM and DIO must both equal "1" before the result bytes may be read from the FIFO. After all the result bytes have been read, the RQM and DIO bits switch to "1" and "0" respectively, and the CB bit is cleared. This indicates that the 82078 is ready to accept the next command.

6.0 COMMAND SET/DESCRIPTIONS

Commands can be written whenever the 82078 is in the command phase. Each command has a unique set of needed parameters and status results. The 82078 checks to see that the first byte is a valid command and, if valid, proceeds with the command. If it was invalid, the next time the RQM bit in the MSR register is a "1" the DIO and CB bits will also be "1", indicating the FIFO must be read. A result byte of 80H will be read out of the FIFO, indicating an invalid command was issued. After reading the result byte from the FIFO the 82078 will return to the command phase. Table 6-1 is a summary of the Command set.

Table 6-1. 82078 Command Set

Phase	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
READ DATA										
Command	W	MT	MFM	SK	0	0	1	1	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W					C				Sector ID Information prior to Command Execution
	W					H				
	W					R				
	W					N				
	W					EOT				
	W					GPL				
	W					DTL				
Execution										Data Transfer between the FDD and System
Result	R					ST 0				Status Information after Command Execution
	R					ST 1				
	R					ST 2				
	R					C				
	R					H				Sector ID Information after Command Execution
	R					R				
	R					N				
READ DELETED DATA										
Command	W	MT	MFM	SK	0	1	1	0	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W					C				Sector ID Information prior to Command Execution
	W					H				
	W					R				
	W					N				
	W					EOT				
	W					GPL				
	W					DTL				
Execution										Data Transfer between the FDD and System
Result	R					ST 0				Status Information after Command Execution
	R					ST 1				
	R					ST 2				
	R					C				
	R					H				Sector ID Information after Command Execution
	R					R				
	R					N				
WRITE DATA										
Command	W	MT	MFM	0	0	0	1	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W					C				Sector ID Information prior to Command Execution
	W					H				
	W					R				
	W					N				
	W					EOT				
	W					GPL				
	W					DTL				
Execution										Data Transfer between the FDD and System
Result	R					ST 0				Status Information after Command Execution
	R					ST 1				
	R					ST 2				
	R					C				
	R					H				Sector ID Information after Command Execution
	R					R				
	R					N				

Table 6-1. 82078 Command Set (Continued)

Phase	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
WRITE DELETED DATA										
Command	W	MT	MFM	0	0	1	0	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W					C				Sector ID Information prior to Command Execution
	W					H				
	W					R				
	W					N				
	W					EOT				
	W					GPL				
	W					DTL				
Execution										Data Transfer between the FDD and System
Result	R					ST 0				Status Information after Command Execution
	R					ST 1				
	R					ST 2				
	R					C				Sector ID Information after Command Execution
	R					H				
	R					R				
	R					N				
READ TRACK										
Command	W	0	MFM	0	0	0	0	1	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W					C				Sector ID Information prior to Command Execution
	W					H				
	W					R				
	W					N				
	W					EOT				
	W					GPL				
	W					DTL				
Execution										Data Transfer between the FDD and System. FDC Reads All Sectors from Index Hole to EOT
Result	R					ST 0				Status Information after Command Execution
	R					ST 1				
	R					ST 2				
	R					C				Sector ID Information after Command Execution
	R					H				
	R					R				
	R					N				
VERIFY										
Command	W	MT	MFM	SK	1	0	1	1	0	Command Codes
	W	EC	0	0	0	0	HDS	DS1	DS0	
	W					C				Sector ID Information prior to Command Execution
	W					H				
	W					R				
	W					N				
	W					EOT				
	W					GPL				
	W					DTL/SC				
Execution										Data Transfer between the FDD and System
Result	R					ST 0				Status Information after Command Execution
	R					ST 1				
	R					ST 2				
	R					C				Sector ID Information after Command Execution
	R					H				
	R					R				
	R					N				
VERSION										
Command	W	0	0	0	1	0	0	0	0	Command Code
Result	R	1	0	0	1	0	0	0	0	Enhanced Controller

Table 6-1. 82078 Command Set (Continued)

Phase	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
FORMAT TRACK										
Command	W	0	MFM	0	0	1	1	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				N					Bytes/Sector
	W				SC					Sectors/Cylinder
	W				GPL					Gap3
	W				D					Filler Byte
Execution For Each Sector Repeat:	W				C					Input Sector Parameters
	W				H					
	W				R					
	W				N					
										82078 Formats an Entire Cylinder
Result	R				ST 0					Status Information after Command Execution
	R				ST 1					
	R				ST 2					
	R				Undefined					
	R				Undefined					
	R				Undefined					
	R				Undefined					
SCAN EQUAL										
Command	W	MT	MFM	SK	1	0	0	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				C					Sector ID Information prior to Command Execution
	W				H					
	W				R					
	W				N					
	W				EOT					
	W				GPL					
	W				STP					
Execution										Data Compared between the FDD and Main-System
Result	R				ST 0					Status Information after Command Execution
	R				ST 1					
	R				ST 2					
	R				C					Sector ID Information after Command Execution
	R				H					
	R				R					
	R				N					

2

Table 6-1. 82078 Command Set (Continued)

Phase	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
SCAN LOW OR EQUAL										
Command	W	MT	MFM	SK	1	1	0	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				C					Sector ID Information
	W				H					Prior to Command
	W				R					Execution
	W				N					
	W				EOT					
	W				GPL					
	W				STP					
Execution										Data Compared Between the FDD and Main-System
Result	R				ST 0					Status Information
	R				ST 1					After Command
	R				ST 2					Execution
	R				C					
	R				H					Sector ID Information
	R				R					After Command
	R				N					Execution
SCAN HIGH OR EQUAL										
Command	W	MT	MFM	SK	1	1	1	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				C					Sector ID Information
	W				H					Prior to Command
	W				R					Execution
	W				N					
	W				EOT					
	W				GPL					
	W				STP					
Execution										Data Compared Between the FDD and Main-System
Result	R				ST 0					Status Information
	R				ST 1					After Command
	R				ST 2					Execution
	R				C					
	R				H					Sector ID Information
	R				R					After Command
	R				N					Execution

Table 6-1. 82078 Command Set (Continued)

Phase	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
RECALIBRATE										
Command	W	0	0	0	0	0	1	1	1	Command Codes
	W	0	0	0	0	0	0	DS1	DS0	Enhanced Controller
Execution										Head Retracted to Track 0 Interrupt
SENSE INTERRUPT STATUS										
Command	W	0	0	0	1	0	0	0	0	Command Codes
Result	R				ST 0					Status Information at the End of each Seek Operation
	R				PVN					
SPECIFY										
Command	W	0	0	0	0	0	0	1	1	Command Codes
	W	SRT						HUT		
	W	HLT							ND	
SENSE DRIVE STATUS										
Command	W	0	0	0	0	0	1	0	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
Result	R				ST 3					Status Information about FDD
DRIVE SPECIFICATION COMMAND										
Command	W	1	0	0	0	1	1	1	0	Command Codes
Phase	W	0	FD1	FD0	PTS	DRT1	DRT0	DT1	DT0	
	:	:	:	:	:	:	:	:	:	0–4 bytes issued
	W	DN	NRP	0	0	0	0	0	0	
Result	R	0	0	0	PTS	DRT1	DRT0	DT1	DT0	Drive 0
Phase	R	0	0	0	PTS	DRT1	DRT0	DT1	DT0	Drive 1
	R	0	0	0	PTS	DRT1	DRT0	DT1	DT0	Drive 2
	R	0	0	0	PTS	DRT1	DRT0	DT1	DT0	Drive 3
SEEK										
Command	W	0	0	0	0	1	1	1	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				NCN					
Execution										Head is Positioned over Proper Cylinder on Diskette
CONFIGURE										
Command	W	CLK48	0	0	1	0	0	1	1	Command Code
	W	0	0	0	0	0	0	0	0	
	W	0	EIS	EFIFO	POLL			FIFOTHR		
	W				PRETRK					
RELATIVE SEEK										
Command	W	1	DIR	0	0	1	1	1	1	
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				RCN					

Table 6-1. 82078 Command Set (Continued)

Phase	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
DUMPREG										
Command Execution	W	0	0	0	0	1	1	1	0	*Note Registers Placed in FIFO
Result	R				PCN-Drive 0					
	R				PCN-Drive 1					
	R				PCN-Drive 2					
	R				PCN-Drive 3					
	R		SRT				HUT			
	R				HLT				ND	
	R				SC/EOT					
	R	LOCK	0	D_3	D_2	D_1	D_0	GAP	WGATE	
	R	0	EIS	EFIFO	POLL		FIFOTHR			
	R				PRETRK					
READ ID										
Command	W	0	MFM	0	0	1	0	1	0	Commands
	W	0	0	0	0	0	HDS	DS1	DS0	
Execution										The First Correct ID Information on the Cylinder is Stored in Data Register
Result	R				ST 0					Status Information after Command Execution
	R				ST 1					
	R				ST 2					
	R				C					
	R				H					Disk Status after the Command has Completed
	R				R					
	R				N					
PERPENDICULAR MODE										
Command	W	0	0	0	1	0	0	1	0	Command Codes
	W	OW	0	D_3	D_2	D_1	D0	GAP	WGATE	
LOCK										
Command	W	LOCK	0	0	1	0	1	0	0	Command Codes
Result	R	0	0	0	LOCK	0	0	0	0	
PART ID										
Command	W	0	0	0	1	1	0	0	0	Command Code
Result	R	0	0	0		—STEPPING—			1	Part ID Number
POWERDOWN MODE										
Command	W	0	0	0	1	0	1	1	1	Command Code
	W	0	0	EREG EN	X	PS2 STAT	FDI TRI	MIN DLY	AUTO PD	
Result	R	0	0	EREG EN	X	PS2 STAT	FDI TRI	MIN DLY	AUTO PD	
OPTION										
Command	W	0	0	1	1	0	0	1	1	Command Code
	W				—RSVD—				ISO	

Table 6-1. 82078 Command Set (Continued)

Phase	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
					SAVE					
Command Phase	W	0	0	1	0	1	1	1	0	Command Code
Result Phase	R	CLK 48	SEL 3V#	PD OSC	PC2	PC1	PC0	DRATE1	DRATE0	Save Info to Reprogram the FDC
	R	0	0	0	0	0	0	0	ISO	
	R				PCN-Drive 0					
	R				PCN-Drive 1					
	R				PCN-Drive 2					
	R				PCN-Drive 3					
	R		SRT					HUT		
	R				HLT				ND	
	R				SC/EOT					
	R	LOCK	0	D_3	D_2	D_1	D_0	GAP	WGATE	
	R	0	EIS	EFIFO	POLL			FIFOTHR		
	R				PRETRK					
	R	0	0	EREG EN	RSVD	PS2 STAT	FDI TRI	MIN DLY	AUTO PD	
	R				DISK/STATUS					
	R				RSVD					
	R				RSVD					
					RESTORE					
Command Phase	W	0	1	0	0	1	1	1	0	Command Code
Result	W	CLK48	SEL 3V#	0	PC2	PC1	PC0	DRATE1	DRATE0	Restore Original Register Status
	W	0	0	0	0	0	0	0	ISO	
	W				PCN-Drive 0					
	W				PCN-Drive 1					
	W				PCN-Drive 2					
	W				PCN-Drive 3					
	W		SRT					HUT		
	W				HLT				ND	
	W				SC/EOT					
	W	LOCK	0	D_3	D_2	D_1	D_0	GAP	WGATE	
	W	0	EIS	EFIFO	POLL			FIFOTHR		
	W				PRETRK					
	W	0	0	EREG EN	RSVD	PS2 STAT	FDI TRI	MIN DLY	AUTO PD	
	W				DISK/STATUS					
	W				RSVD					
	W				RSVD					

2

Table 6-1. 82078 Command Set (Continued)

Phase	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
FORMAT AND WRITE										
Command	W	1	MFM	1	0	1	1	0	1	Command Code
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				N					
	W				SC					
	W				GPL					
	W				D					
Execution	W				C					Input
Repeated	W				H					Sector
for each	W				R					Parameters
Sector	W				N					
	W	Data Transfer of N Bytes								82078 Formats and Writes Entire Track
Result	R				ST 0					
Phase	R				ST 1					
	R				ST 2					
	R				Undefined					
	R				Undefined					
	R				Undefined					
	R				Undefined					
INVALID										
Command	W				Invalid Codes					Invalid Command Codes (NoOp — 82078 goes into Standby State)
Result	R				ST 0					ST 0 = 80H

PARAMETER ABBREVIATIONS

Symbol	Description
AUTO PD	Auto powerdown control. If this bit is 0, then the automatic powerdown is disabled. If it is set to 1, then the automatic powerdown is enabled.
C	Cylinder address. The currently selected cylinder address, 0 to 255.
CLK48	CLK48 = 1 indicates an external 48 MHz oscillator is being used. CLK48 = 0 indicates a 24 MHz clock.
D0, D1, D2, D3	Drive Select 0-3. Designates which drives are Perpendicular drives, a "1" indicating Perpendicular drive.
D	Data pattern. The pattern to be written in each sector data field during formatting.
DN	Done. This bit indicates that this is the last byte of the drive specification command. The 82078 checks to see if this bit is high or low. If it is low, it expects more bytes. DN = 0 82078 expects more subsequent bytes. DN = 1 Terminates the command phase and jumps to the results phase. An additional benefit is that by setting this bit high, a direct check of the current drive specifications can be done.
DIR	Direction control. If this bit is 0, then the head will step out from the spindle during a relative seek. If set to a 1, the head will step in toward the spindle.
DS0, DS1	Disk Drive Select.

DS1	DS0	
0	0	Drive 0
0	1	Drive 1
1	0	Drive 2
1	1	Drive 3

DTL	Special sector size. By setting N to zero (00), DTL may be used to control the number of bytes transferred in disk read/write commands. The sector size (N = 0) is set to 128. If the actual sector (on the diskette) is larger than DTL, the remainder of the actual sector is read but is not passed to the host during read commands; during write commands, the remainder of the actual sector is written with all zero bytes. The CRC check code is calculated with the actual sector. When N is not zero, DTL has no meaning and should be set to FF HEX.
DRATE1, DRATE0	Data rate values from the DSR register.
DRT0, DRT1	Data rate table select. These two bits select between the different data rate tables. The default is the conventional table. These also provide mapping of the data rates selected in the DSR and CCR. The mapped values are provided for read back by the system software are as shown in the DIR (in PS/2 Mode only). Table 6-2 shows this.
DT0, DT1	Drive density select type. These bits select the outputs on DRVDEN0 and DRVDEN1 based on mode of operation that was selected via the IDENT1 and IDENT0 pins. More information is available in the Design Applications section.

Table 6-2. Data Rate Select Table

		Bits in DSR/CCR			Bits returned via DIR (Only available in PS/2)		
DRT0	**DRT1**	**DRATE0**	**DRATE1**	**Data Rate**	**DRATE0**	**DRATE1**	**Operation**
0	0	1	1	1 Mbps	1	1	Default
		0	0	500 Kbps	0	0	
		1	0	300 Kbps	1	0	
		0	1	250 Kbps	0	1	
0	1	1	1	1 Mbps	1	1	2 Mbps Tape Drive
		0	0	500 Kbps	0	0	
		1	0	2 Mbps	1	1	
		0	1	250 Kbps	0	1	
1	0			——RSVD——			RSVD
1	1	1	1	1 Mbps	1	1	Perpendicular mode FDDs
		0	0	500 Kbps	0	0	
		1	0	RSVD			
		0	1	250 Kbps	0	1	

EC — Enable Count. When this bit is "1" the "DTL" parameter of the Verify Command becomes SC (Number of sectors per track).

EFIFO — Enable FIFO. When this bit is 0, the FIFO is enabled. A "1" puts the 82078 in the 8272A compatible mode where the FIFO is disabled.

EIS — Enable implied seek. When set, a seek operation will be performed before executing any read or write command that requires the C parameter in the command phase. A "0" disables the implied seek.

EOT — End of track. The final sector number of the current track.

EREG EN — Enhanced Register Enable.

EREG EN = 1 In PS/2 mode the TDR register is extended. In AT/EISA mode, the TDR register is extended and SRB is made visible to the user.

EREG EN = 0 Standard AT/EISA and PS/2 registers are used.

FDI TRI — Floppy Drive Interface Tri-state: If this bit is 0, then the output pins of the floppy disk drive interface are tri-stated. This is also the default state. If it is set to 1, then the floppy disk drive interface remains unchanged.

FD0, FD1 — Floppy drive select. These two bits select which physical drive is being specified. The FDn corresponds to FDSn and FDMEn on the floppy drive interface. The drive is selected independent of the BOOTSELn bits. Please refer to Section 2.1.1 which explains the distinction between physical drives and their virtual mapping as defined by the BOOTSEL1 and BOOTSEL0 bits.

FD0	FD1	Drive Slot
0	0	Drive 0
0	1	Drive 1
1	0	Drive 2
1	1	Drive 3

GAP — Alters Gap 2 length when using Perpendicular Mode.

GPL — Gap length. The gap 3 size. (Gap 3 is the space between sectors excluding the VCO synchronization field).

HDS — Head address. Selected head: 0 or 1 (disk side 0 or 1) as encoded in the sector ID field.

HLT — Head load time. The time interval that 82078 waits after loading the head and before initiating a read or write operation. Refer to the SPECIFY command for actual delays.

HUT — Head unload time. The time interval from the end of the execution phase (of a read or write command) until the head is unloaded. Refer to the SPECIFY command for actual delays.

ISO — ISO Format: If this bit is set high the ISO format is used for all data transfer commands. When this bit is set low the normal IBM system 34 and perpendicular is used. The default is ISO = 0.

Lock — Lock defines whether EFIFO, FIFOTHR, and PRETRK parameters of the CONFIGURE command can be reset to their default values by a "Software Reset" (Reset made by setting the proper bit in the DSR or DOR registers).

MFM — MFM mode. A one selects the double density (MFM) mode. A zero is reserved.

MIN DLY — Minimum power up time control. This bit is active only if AUTO PD bit is enabled. Setting this bit to a 0, assigns a 10 ms minimum power up time and setting this bit to a 1, assigns a 0.5 sec. minimum power up time (unless 2 Mbps, then 5 ms to 0.25 sec.).

MT — Multi-track selector. When set, this flag selects the multi-track operating mode. In this mode, the 82078 treats a complete cylinder, under head 0 and 1, as a single track. The 82078 operates as if this expanded track started at the first sector under head 0 and ended at the last sector under head 1. With this flag set, a multitrack read or write operation will automatically continue to the first sector under head 1 when the 82078 finishes operating on the last sector under head 0.

N — Sector size code. This specifies the number of bytes in a sector. If this parameter is "00", then the sector size is 128 bytes. The number of bytes transferred is determined by the DTL parameter. Otherwise the sector size is (2 raised to the "N'th" power) times 128. All values up to "07" hex are allowable. "07"h would equal a sector size of 16k. It is the users responsibility to not select combinations that are not possible with the drive.

N	Sector Size
00	128 bytes
01	256 bytes
02	512 bytes
03	1024 bytes
—	—
07	16 Kbytes

NCN — New cylinder number. The desired cylinder number.

ND — Non-DMA mode flag. When set to 1, indicates that the 82078 is to operate in the non-DMA mode. In this mode, the host is interrupted for each data transfer. When set to 0, the 82078 operates in DMA mode, interfacing to a DMA controller by means of the DRQ and DACK# signals.

NRP — No Results phase. When this bit is set high the result phase is skipped. When this bit is low the result phase will be generated.

OW — The bits denoted D0, D1, D2, and D3 of the PERPENDICULAR MODE command can only be overwritten when the OW bit is set to "1".

PCN — Present cylinder number. The current position of the head at the completion of SENSE INTERRUPT STATUS command.

PC2, PC1, PC0 — Precompensation values from the DSR register.

PDOSC — When this bit is set, the internal oscillator is turned off.

This may be done if using the external 48 MHz oscillator.

PS/2 STAT — PS/2 status. This bit is functional only in the PS/2 mode. In all other modes this bit will not have any effect. When set high this bit enables two bits (bits 5 and 6) in the DIR register to reflect the values of PD and IDLE respectively except when IDLEMSK (bit 4) is set. Default value is 0.

PTS — Precompensation table select. This bit selects whether to enable the precompensation value programmed in the DSR or not. In the default state, the value programmed in DSR will be used.

PTS = 0 DSR programmed precompensation delays.

PTS = 1 No precompensation delay is selected for the corresponding drive.

POLL — Polling disable. When set, the internal polling routine is disabled. When clear, polling is enabled.

PRETRK — Precompensation start track number. Programmable from track 00 to FFH.

R — Sector address. The sector number to be read or written. In multi-sector transfers, this parameter specifies the sector number of the first sector to be read or written.

RCN — Relative cylinder number. Relative cylinder offset from present cylinder as used by the RELATIVE SEEK command.

SC — Number of sectors. The number of sectors to be initialized by the FORMAT command. The number of sectors to be verified during a Verify Command, when EC is set.

SEL3V# — SEL3V# = 1 indicates that the part is operating at 5.0V.

SEL3V# = 0 indicates that the part is operating at 3.3V.

SK — Skip flag. When set to 1, sectors containing a deleted data address mark will automatically be skipped during the execution of READ DATA. If READ DELETED is executed, only sectors with a deleted address mark will be accessed. When set to "0", the sector is read or written the same as the read and write commands.

SRT — Step rate interval. The time interval between step pulses issued by the 82078. Programmable from 0.5 ms to 8 ms, in increments of 0.5 ms at the 1 Mbit data rate. Refer to the SPECIFY command for actual delays.

ST0–3 — Status registers 0–3. Registers within the 82078 that store status information after a command has been executed. This status information is available to the host during the result phase after command execution.

STEPPING — These bits identify the stepping of the 82078.

WGATE — Write gate alters timing of WE, to allow for pre-erase loads in perpendicular drives.

6.1 Data Transfer Commands

All of the READ DATA, WRITE DATA and VERIFY type commands use the same parameter bytes and return the same results information. The only difference being the coding of bits 0–4 in the first byte.

An implied seek will be executed if the feature was enabled by the CONFIGURE command. This seek is completely transparent to the user. The Drive Busy bit for the drive will go active in the Main Status Register during the seek portion of the command. If the seek portion fails, it will be reflected in the results status normally returned for a READ/WRITE DATA command. Status Register 0 (ST0) would contain the error code and C would contain the cylinder on which the seek failed.

6.1.1 READ DATA

A set of nine (9) bytes is required to place the 82078 into the Read Data Mode. After the READ DATA command has been issued, the 82078 loads the head (if it is in the unloaded state), waits the specified head settling time (defined in the SPECIFY command), and begins reading ID Address Marks and ID fields. When the sector address read off the diskette matches with the sector address specified in the command, the 82078 reads the sector's data field and transfers the data to the FIFO.

After completion of the read operation from the current sector, the sector address is incremented by one, and the data from the next logical sector is read and output via the FIFO. This continuous read function is called "Multi-Sector Read Operation". Upon receipt of TC, or an implied TC (FIFO overrun/underrun), the 82078 stops sending data, but will continue to read data from the current sector, check the CRC bytes, and at the end of the sector terminate the READ DATA Command.

N determines the number of bytes per sector (see Table 6-3). If N is set to zero, the sector size is set to 128. The DTL value determines the number of bytes to be transferred. If DTL is less than 128, the 82078 transfers the specified number of bytes to the host. For reads, it continues to read the entire 128 byte sector and checks for CRC errors. For writes it completes the 128 byte sector by filling in zeroes. If N is not set to 00 Hex, DTL should be set to FF Hex, and has no impact on the number of bytes transferred.

Table 6-3. Sector Sizes

N	Sector Size
00	128 bytes
01	256 bytes
02	512 bytes
03	1024 bytes
—	—
07	16 Kbytes

The amount of data which can be handled with a single command to the 82078 depends upon MT (multi-track) and N (Number of bytes/sector).

Table 6-4. Effects of MT and N Bits

MT	N	Max. Transfer Capacity	Final Sector Read from Disk
0	1	256 × 26 = 656	26 at side 0 or 1
1	1	256 × 52 = 13312	26 at side 1
0	2	512 × 15 = 7680	15 at side 0 or 1
1	2	512 × 30 = 15360	15 at side 1
0	3	1024 × 8 = 8192	8 at side 0 or 1
1	3	1024 × 16 = 16384	16 at side 1

The Multi-Track function (MT) allows the 82078 to read data from both sides of the diskette. For a particular cylinder, data will be transferred starting at Sector 1, Side 0 and completing at the last sector of the same track at Side 1.

If the host terminates a read or write operation in the 82078, then the ID information in the result phase is dependent upon the state of the MT bit and EOT byte. Refer to Table 6-7. The termination must be normal.

At the completion of the READ DATA Command, the head is not unloaded until after the Head Unload Time Interval (specified in the SPECIFY command) has elapsed. If the host issues another command before the head unloads then the head settling time may be saved between subsequent reads.

If the 82078 detects a pulse on the INDX# pin twice without finding the specified sector (meaning that the diskette's index hole passes through index detect logic in the drive twice), the 82078 sets the IC code in Status Register 0 to "01" (Abnormal termination), and sets the ND bit in Status Register 1 to "1" indicating a sector not found, and terminates the READ DATA Command.

After reading the ID and Data Fields in each sector, the 82078 checks the CRC bytes. If a CRC error occurs in the ID or data field, the 82078 sets the IC code in Status Register 0 to "01" (Abnormal termination), sets the DE bit flag in Status Register 1 to "1", sets the DD bit in Status Register 2 to "1" if CRC is incorrect in the ID field, and terminates the READ DATA Command.

Table 6-5 describes the affect of the SK bit on the READ DATA command execution and results.

Table 6-5. Skip Bit vs READ DATA Command

SK Bit Value	Data Address Mark Type Encountered	Results		
		Sector Read?	CM Bit of ST2 Set?	Description of Results
0	Normal Data	Yes	No	Normal Termination.
0	Deleted Data	Yes	Yes	Address Not Incremented. Next Sector Not Searched For.
1	Normal Data	Yes	No	Normal Termination.
1	Deleted Data	No	Yes	Normal Termination Sector Not Read ("Skipped").

Except where noted in Table 6-5, the C or R value of the sector address is automatically incremented (see Table 6-7).

6.1.2 READ DELETED DATA

This command is the same as the READ DATA command, only it operates on sectors that contain a Deleted Data Address Mark at the beginning of a Data Field.

Table 6-6 describes the affect of the SK bit on the READ DELETED DATA command execution and results.

Table 6-6. Skip Bit vs READ DELETED DATA Command

SK Bit Value	Data Address Mark Type Encountered	Results		
		Sector Read?	CM Bit of ST2 Set?	Description of Results
0	Normal Data	Yes	Yes	Normal Termination.
0	Deleted Data	Yes	No	Address Not Incremented. Next Sector Not Searched For.
1	Normal Data	No	Yes	Normal Termination Sector Not Read ("Skipped").
1	Deleted Data	Yes	No	Normal Termination.

Except where noted in Table 6-6 above, the C or R value of the sector address is automatically incremented (see Table 6-7).

6.1.3 READ TRACK

This command is similar to the READ DATA command except that the entire data field is read continuously from each of the sectors of a track. Immediately after encountering a pulse on the INDX# pin, the 82078 starts to read all data fields on the track as continuous blocks of data without regard to logical sector numbers. If the 82078 finds an error in the ID or DATA CRC check bytes, it continues to read data from the track and sets the appropriate error bits at the end of the command. The 82078 compares the ID information read from each sector with the specified value in the command, and sets the ND flag of Status Register 1 to a "1" if there is no comparison.

Multi-track or skip operations are not allowed with this command. The MT and SK bits (Bits D7 and D5 of the first command byte respectively) should always be set to "0".

Table 6-7. Result Phase Table

MT	Head	Final Sector Transferred to Host	ID Information at Result Phase			
			C	H	R	N
0	0	Less than EOT	NC	NC	R+1	NC
		Equal to EOT	C+1	NC	01	NC
	1	Less than EOT	NC	NC	R+1	NC
		Equal to EOT	C+1	NC	01	NC
1	0	Less than EOT	NC	NC	R+1	NC
		Equal to EOT	NC	LSB	01	NC
	1	Less than EOT	NC	NC	R+1	NC
		Equal to EOT	C+1	LSB	01	NC

NC: No Change, the same value as the one at the beginning of command execution.
LSB: Least Significant Bit, the LSB of H is complemented.

This command terminates when the EOT specified number of sectors have been read. If the 82078 does not find an ID Address Mark on the diskette after the second occurrence of a pulse on the INDX# pin, then it sets the IC code in Status Register 0 to "01" (Abnormal termination), sets the MA bit in Status Register 1 to "1", and terminates the command.

6.1.4 WRITE DATA

After the WRITE DATA command has been issued, the 82078 loads the head (if it is in the unloaded state), waits the specified head load time if unloaded (defined in the SPECIFY command), and begins reading ID Fields. When the sector address read from the diskette matches the sector address specified in the command, the 82078 reads the data from the host via the FIFO, and writes it to the sector's data field.

After writing data into the current sector, the 82078 computes the CRC value and writes it into the CRC field at the end of the sector transfer. The Sector Number stored in "R" is incremented by one, and the 82078 continues writing to the next data field. The 82078 continues this "Multi-Sector Write Operation". Upon receipt of a terminal count signal or if a FIFO over/under run occurs while a data field is being written, then the remainder of the data field is filled with zeroes.

The 82078 reads the ID field of each sector and checks the CRC bytes. If it detects a CRC error in one of the ID Fields, it sets the IC code in Status Register 0 to "01" (Abnormal termination), sets the DE bit of Status Register 1 to "1", and terminates the WRITE DATA command.

The WRITE DATA command operates in much the same manner as the READ DATA command. The following items are the same. Please refer to the READ DATA Command for details:

- Transfer Capacity
- (End of Cylinder) bit
- ND (No Data) bit
- Head Load, Unload Time Interval
- ID information when the host terminates the command.
- Definition of DTL when N = 0 and when N does not = 0.

6.1.5 WRITE DELETED DATA

This command is almost the same as the WRITE DATA command except that a Deleted Data Address Mark is written at the beginning of the Data Field instead of the normal Data Address Mark. This command is typically used to mark a bad sector containing an error on the floppy disk.

6.1.6 VERIFY

The VERIFY command is used to verify the data stored on a disk. This command acts exactly like a READ DATA command except that no data is transferred to the host. Data is read from the disk, CRC computed and checked against the previously stored value.

Because no data is transferred to the host, TC (pin 25) cannot be used to terminate this command. By setting the EC bit to "1" an implicit TC will be issued to the 82078. This implicit TC will occur when the SC value has decrement to 0 (an SC value of 0 will verify 256 sectors). This command can also be terminated by setting the EC bit to "0" and the EOT value equal to the final sector to be checked. If EC is set to "0" DTL/SC should be programmed to 0FFH. Refer to Table 6-6 and Table 6-7 for information concerning the values of MT and EC versus SC and EOT value.

Definitions:

Sectors Per Side = Number of formatted sectors per each side of the disk.

Sectors Remaining = Number of formatted sectors left which can be read, including side 1 of the disk if MT is set to "1".

Table 6-8. Verify Command Result Phase Table

MT	EC	SC/EOT Value	Termination Result
0	0	SC = DTL EOT ≤ # Sectors Per Side	Successful Termination Result Phase Valid
0	0	SC = DTL EOT > # Sectors Per Side	Unsuccessful Termination Result Phase Invalid
0	1	SC ≤ # Sectors Remaining AND EOT ≤ # Sectors Per Side	Successful Termination Result Phase Valid
0	1	SC > # Sectors Remaining OR EOT > # Sectors Per Side	Unsuccessful Termination Result Phase Invalid
1	0	SC = DTL EOT ≤ # Sectors Per Side	Successful Termination Result Phase Valid
1	0	SC = DTL EOT > # Sectors Per Side	Unsuccessful Termination Result Phase Invalid
1	1	SC ≤ # Sectors Remaining AND EOT ≤ # Sectors Per Side	Successful Termination Result Phase Valid
1	1	SC > # Sectors Remaining OR EOT > # Sectors Per Side	Unsuccessful Termination Result Phase Invalid

NOTE:
If MT is set to "1" and the SC value is greater than the number of remaining formatted sectors on Side 0, verifying will continue on Side 1 of the disk.

6.1.7 FORMAT TRACK

The FORMAT command allows an entire track to be formatted. After a pulse from the INDX# pin is detected, the 82078 starts writing data on the disk including Gaps, Address Marks, ID Fields and Data Fields, per the IBM System 34 (MFM). The particular values that will be written to the gap and data field are controlled by the values programmed into N, SC, GPL, and D which are specified by the host during the command phase. The data field of the sector is filled with the data byte specified by D. The ID Field for each sector is supplied by the host; that is, four data bytes per sector are needed by the 82078 for C, H, R, and N (cylinder, head, sector number and sector size respectively).

After formatting each sector, the host must send new values for C, H, R and N to the 82078 for the next sector on the track. The R value (sector number) is the only value that must be changed by the host after each sector is formatted. This allows the disk to be formatted with nonsequential sector addresses (interleaving). This incrementing and formatting continues for the whole track until the 82078 encounters a pulse on the INDX# pin again and it terminates the command.

Table 6-9 contains typical values for gap fields which are dependent upon the size of the sector and the number of sectors on each track. Actual values can vary due to drive electronics.

Table 6-9. Typical PC-AT Values for Formatting

Drive Form	MEDIA	Sector Size	N	SC	GPL1	GPL2
5.25″	1.2M	512	02	0F	2A	50
	360K	512	02	09	2A	50
3.5″	2.88M	512	02	24	38	53
	1.44M	512	02	18	1B	54
	720K	512	02	09	1B	54

NOTE:
All values except Sector Size are in Hex.

Gap3 is programmable during reads, writes, and formats.

GPL1 = suggested Gap3 values in read and write commands to avoid splice point between data field and ID field of contiguous sections.

GPL2 = suggested Gap3 value in FORMAT TRACK command.

6.1.7.1 Format Fields

Table 6-10. System 34 Format Double Density

GAP 4a	SYNC	IAM		GAP 1	SYNC	IDAM		CYL	HD	SEC	NO	CRC	GAP 2	SYNC	DATA AM		DATA	CRC	GAP 3	GAP 4b
80x 4E	12x 00	3x C2	FC	50x 4E	12x 00	3x A1	FE						22x 4E	12x 00	3x A1	FB F8				

Table 6-11. ISO Format

GAP 1	SYNC	IDAM		CYL	HD	SEC	NO	CRC	GAP 2	SYNC	DATA AM		DATA	CRC	GAP 3	GAP 4b
32x 4E	12x 00	3x A1	FE						22x 4E	12x 00	3x A1	FB F8				

Table 6-12. Perpendicular Format

GAP 4a	SYNC	IAM		GAP 1	SYNC	IDAM		CYL	HD	SEC	NO	CRC	GAP 2	SYNC	DATA AM		DATA	CRC	GAP 3	GAP 4b
80x 4E	12x 00	3x C2	FC	50x 4E	12x 00	3x A1	FE						41x 4E	12x 00	3x A1	FB F8				

6.2 Control Commands

Control commands differ from the other commands in that no data transfer takes place. Three commands generate an interrupt when complete; READ ID, RECALIBRATE and SEEK. The other control commands do not generate an interrupt.

6.2.1 READ ID

The READ ID command is used to find the present position of the recording heads. The 82078 stores the values from the first ID Field it is able to read into its registers. If the 82078 does not find an ID Address Mark on the diskette after the second occurrence of a pulse on the INDX# pin, it then sets the IC code in Status Register 0 to "01" (Abnormal termination), sets the MA bit in Status Register 1 to "1", and terminates the command.

The following commands will generate an interrupt upon completion. They do not return any result bytes. It is highly recommended that control commands be followed by the SENSE INTERRUPT STATUS command. Otherwise, valuable interrupt status information will be lost.

6.2.2 RECALIBRATE

This command causes the read/write head within the 82078 to retract to the track 0 position. The 82078 clears the contents of the PCN counter, and checks the status of the TRK0 pin from the FDD. As long as the TRK0 pin is low, the DIR pin remains 0 and step pulses are issued. When the TRK0 pin goes high, the SE bit in Status Register 0 is set to "1", and the command is terminated. If the TRK0 pin is still low after 79 step pulses have been issued, the 82078 sets the SE and the EC bits of Status Register 0 to "1", and terminates the command. Disks capable of handling more than 80 tracks per side may require more than one RECALIBRATE command to return the head back to physical Track 0.

The RECALIBRATE command does not have a result phase. SENSE INTERRUPT STATUS command must be issued after the RECALIBRATE command to effectively terminate it and to provide verification of the head position (PCN). During the command phase of the recalibrate operation, the 82078 is in the BUSY state, but during the execution phase it is in a NON BUSY state. At this time another RECALIBRATE command may be issued, and in this manner, parallel RECALIBRATE operations may be done on up to 4 drives at once.

Upon power up, the software must issue a RECALIBRATE command to properly initialize all drives and the controller.

6.2.3 DRIVE SPECIFICATION COMMAND

The 82078 uses two pins, DRVDEN0 and DRVDEN1 to select the density for modern drives. These signals inform the drive of the type of diskette in the drive. The Drive Specification command specifies the polarity of the DRVDEN0 and DRVDEN1 pins. It also enables or disables DSR programmed precompensation.

This command removes the need for a hardware workaround to accommodate differing specifications among drives. By programming this command during BIOS's POST routine, the floppy disk controller will internally configure the correct values for DRVDEN0 and DRVDEN1 with corresponding precompensation value and data rate table enabled for the particular type of drive.

This command is protected from software resets. After executing the DRIVE SPEC command, subsequent software resets will not clear the programmed parameters. Only another DRIVE SPEC command or H/W reset can reset it to default values. The 6 LSBs of the last byte of this command are reserved for future use.

The DRATE0 and DRATE1 are values as programmed in the DSR register. The DENSEL is high for high data rates (1 Mbps and 500 Kbps) and low for low data rates (300 Kbps and 250 Kbps).

Table 6-13 describes the drives that are supported with the DT0, DT1 bits of the Drive Specification command:

Table 6-13. Drive Support via the Drive Specification Command

DRVDENn Polarities for AT/EISA Mode (IDENT0, IDENT1 = 11)

DT0	DT1	Data Rate	DRVDEN0	DRVDEN1
0*	0*	1 Mbps	1	1
		500 Kbps	1	0
		300 Kbps	0	1
		250 Kbps	0	0
0	1	1 Mbps	1	1
		500 Kbps	0	0
		300 Kbps	0	1
		250 Kbps	1	0
1	0	1 Mbps	0	1
		500 Kbps	0	0
		300 Kbps	1	1
		250 Kbps	1	0
1	1	1 Mbps	1	1
		500 Kbps	0	0
		300 Kbps	1	0
		250 Kbps	0	1

(*) Denotes the default setting

DRVDEN Polarities for PS/2, Model 30 Mode (IDENT0, IDENT1 = 0X)

DT0	DT1	Data Rate	DRVDEN0	DRVDEN1
0*	0*	1 Mbps	1	1
		500 Kbps	0	0
		300 Kbps	1	0
		250 Kbps	0	1
0	1	1 Mbps	1	1
		500 Kbps	1	0
		300 Kbps	0	1
		250 Kbps	0	0
1	0	1 Mbps	0	1
		500 Kbps	0	0
		300 Kbps	1	1
		250 Kbps	1	0
1	1	1 Mbps	1	1
		500 Kbps	0	0
		300 Kbps	0	1
		250 Kbps	1	0

(*) Denotes the default setting

6.2.4 SEEK

The read/write head within the drive is moved from track to track under the control of the SEEK Command. The 82078 compares the PCN which is the current head position with the NCN and performs the following operation if there is a difference:

PCN < NCN: Direction signal to drive set to "1" (step in), and issues step pulses.

PCN > NCN: Direction signal to drive set to "0" (step out), and issues step pulses.

The rate at which step pulses are issued is controlled by SRT (Stepping Rate Time) in the SPECIFY command. After each step pulse is issued, NCN is compared against PCN, and when NCN = PCN, then the SE bit in Status Register 0 is set to "1", and the command is terminated.

During the command phase of the seek or recalibrate operation, the 82078 is in the BUSY state, but during the execution phase it is in the NON BUSY state.

Note that if implied seek is not enabled, the read and write commands should be preceded by:

1. SEEK command; Step to the proper track
2. SENSE INTERRUPT STATUS command; Terminate the Seek command
3. READ ID. Verify head is on proper track
4. Issue READ/WRITE command.

The SEEK command does not have a result phase. Therefore, it is highly recommended that the SENSE INTERRUPT STATUS Command be issued after the SEEK command to terminate it and to provide verification of the head position (PCN). The H bit (Head Address) in ST0 will always return a "0". When exiting DSR POWERDOWN mode, the 82078 clears the PCN value and the status information to zero. Prior to issuing the DSR POWERDOWN command, it is highly recommended that the user service all pending interrupts through the SENSE INTERRUPT STATUS command.

6.2.5 SCAN COMMANDS

The SCAN Commands allow data which is being read from the diskette to be compared against data which is being supplied from the main system (Processor in NON-DMA mode, and DMA Controller in DMA mode). The FDC compares the data on a byte-by-byte basis, and looks for a sector of data which meets the conditions of $D_{FDD} = D_{Processor}$, $D_{FDD} \leq D_{Processor}$, or $D_{FDD} \geq D_{Processor}$. Ones complement arithmetic is used for comparison (FF = largest number, 00 = smallest number). After a whole sector of data is compared, if the conditions are not met, the sector number is incremented (R + STP → R), and the scan operation is continued. The scan operation continues until one of the following conditions occur; the conditions for scan are met (equal, low, or high), the last sector on the track is reached (EOT), or the terminal count signal is received.

If the conditions for scan are met then the FDC sets the SH (Scan Hit) flag of Status Register 2 to a 1 (high), and terminates the Scan Command. If the conditions for scan are not met between the starting sector (as specified by R) and the last sector on the cylinder (EOT), then the FDC sets the SN (Scan Not Satisfied) flag of Status Register 2 to a 1 (high), and terminates the Scan Command. The receipt of a TERMINAL COUNT signal from the Processor or DMA Controller during the scan operation will cause the FDC to complete the comparison of the particular byte which is in process, and then to terminate the command. Table 6-9 shows the status of bits SH and SN under various conditions of SCAN.

If the FDC encounters a Deleted Data Address Mark on one of the sectors (and SK = 0), then it regards the sector as the last sector on the cylinder, sets CM (Control Mark) flag of Status Register 2 to a 1 (high) and terminates the command. If SK = 1, the FDC skips the sector with the Deleted Address Mark, and reads the next sector. In the second case (SK = 1), the FDC sets the CM (Control Mark) flag of Status Register 2 to a 1 (high) in order to show that a Deleted Sector has been encountered.

When either the STP (contiguous sectors STP = 01, or alternate sectors STP = 02 sectors are read) or the MT (Multi-Track) are programmed, it is necessary to remember that the last sector on the track must be read. For example, if STP = 02, MT = 0, the sectors are numbered sequentially 1 through 26, and we start the Scan Command at sector 21; the following will happen. Sectors 21, 23, and 25 will be read, then the next sector (26) will be skipped and the Index Hole will be encountered before the EOT value of 26 can be read. This will result in an abnormal termination of the command. If the EOT had been set at 25 or the scanning started at sector 20, then the Scan Command would be completed in a normal manner.

During the Scan Command data is supplied by either the processor or DMA Controller for comparison against the data read from the diskette. In order to avoid having the OR (Over Run) flag set in Status Register 1, it is necessary to have the data available in less than 13 μs. If an Overrun occurs the FDC terminates the command.

Table 6-13. Scan Status Codes

Command	Status Register 2		Comments
	Bit 2 = SN	Bit 3 = SH	
Scan Equal	0	1	$D_{FDD} = D_{Processor}$
	1	0	$D_{FDD} \neq D_{Processor}$
Scan Low or Equal	0	1	$D_{FDD} = D_{Processor}$
	0	0	$D_{FDD} < D_{Processor}$
	1	0	$D_{FDD} \not> D_{Processor}$
Scan High or Equal	0	1	$D_{FDD} = D_{Processor}$
	0	0	$D_{FDD} > D_{Processor}$
	1	0	$D_{FDD} \not< D_{Processor}$

6.2.6 SENSE INTERRUPT STATUS

An interrupt signal on INT pin is generated by the 82078 for one of the following reasons:

1. Upon entering the Result Phase of:
   a. READ DATA Command
   b. READ TRACK Command
   c. READ ID Command
   d. READ DELETED DATA Command
   e. WRITE DATA Command
   f. FORMAT TRACK Command
   g. WRITE DELETED DATA Command
   h. VERIFY Command
2. End of SEEK, RELATIVE SEEK or RECALIBRATE Command
3. 82078 requires a data transfer during the execution phase in the non-DMA Mode

The SENSE INTERRUPT STATUS command resets the interrupt signal and via the IC code and SE bit of Status Register 0, identifies the cause of the interrupt. If a SENSE INTERRUPT STATUS command is issued when no active interrupt condition is present, the status register ST0 will return a value of 80H (invalid command).

Table 6-14. Interrupt Identification

SE	IC	Interrupt Due To
0	11	Polling
1	00	Normal Termination of SEEK or RECALIBRATE command
1	01	Abnormal Termination of SEEK or RECALIBRATE command

The SEEK, RELATIVE SEEK and the RECALIBRATE commands have no result phase. SENSE INTERRUPT STATUS command must be issued immediately after these commands to terminate them and to provide verification of the head position (PCN). The H (Head Address) bit in ST0 will always return a "0". If a SENSE INTERRUPT STATUS is not issued, the drive will continue to be BUSY and may effect the operation of the next command.

6.2.7 SENSE DRIVE STATUS

SENSE DRIVE STATUS obtains drive status information. It has no execution phase and goes directly to the result phase from the command phase. STATUS REGISTER 3 contains the drive status information.

6.2.8 SPECIFY

The SPECIFY command sets the initial values for each of the three internal timers. The HUT (Head Unload Time) defines the time from the end of the execution phase of one of the read/write commands to the head unload state. The SRT (Step Rate Time) defines the time interval between adjacent step pulses. Note that the spacing between the first and second step pulses may be shorter than the remaining step pulses. The HLT (Head Load Time) defines the time between the command phase to the execution phase of a read/write data command. The Head Unload Time (HUT) timer starts at the end of the execution phase to the beginning of the result phase of a read/write command. The values change with the data rate speed selection and are documented in Table 6-15.

Table 6-15. Drive Control Delays (ms)

	HUT				SRT			
	1M	500K	300K	250K	1M	500K	300K	250K
0	128	256	426	512	8.0	16	26.7	32
1	8	16	26.7	32	7.5	15	25	30
—	—	—	—	—	—	—	—	—
A	80	160	267	320	3.0	6.0	10.2	12
B	88	176	294	352	2.5	5.0	8.35	10
C	96	192	320	384	2.0	4.0	6.68	8
D	104	208	346	416	1.5	3.0	5.01	6
E	112	224	373	448	1.0	2.0	3.33	4
F	120	240	400	480	0.5	1.0	1.67	2

	HLT			
	1M	500K	300K	250K
00	128	256	426	512
01	1	2	3.3	4
02	2	4	6.7	8
—	—	—	—	—
7E	126	252	420	504
7F	127	254	423	508

The choice of DMA or NON-DMA operations is made by the ND bit. When this bit is "1", the NON-DMA mode is selected, and when ND is "0", the DMA mode is selected. In DMA mode, data transfers are signalled by the DRQ pin. Non-DMA mode uses the RQM bit and the INT pin to signal data transfers.

6.2.9 CONFIGURE

Issue the configure command to enable features like the programmable FIFO and set the beginning track for pre-compensation. A CONFIGURE command need not be issued if the default values of the 82078 meet the system requirements. The CLK48 bit allows the 82078 to connect to a 48 MHz oscillator, this can reduce board space if there is a 48 MHz signal already available on the system.

CONFIGURE DEFAULT VALUES:

EIS — No Implied Seeks
EFIFO — FIFO Disabled
POLL — Polling Enabled
FIFOTHR — FIFO Threshold Set to 1 Byte
PRETRK — Pre-Compensation Set to Track 0

EIS—Enable implied seek. When set to "1", the 82078 will perform a SEEK operation before executing a read or write command. Defaults to no implied seek.

EFIFO—A "1" puts the FIFO into the 8272A compatible mode where the FIFO is disabled. This means data transfers are asked for on a byte by byte basis. Defaults to "1", FIFO disabled. The threshold defaults to one.

POLL—Disable polling of the drives. Defaults to "0", polling enabled. When enabled, a single interrupt is generated after a RESET. No polling is performed while the drive head is loaded and the head unload delay has not expired.

FIFOTHR—The FIFO threshold in the execution phase of read or write commands. This is programmable from 1 byte to 16 bytes. Defaults to one byte. A "00" selects one byte, "0F" selects 16 bytes.

PRETRK—Pre-compensation start track number. Programmable from track 0 to 255. Defaults to track 0. A "00" selects track 0, "FF" selects 255.

CLK48—Default is "0", external clock is assumed to be 24 MHz. If a 48 MHz external oscillator is used the bit must be set high. Note that the 82078 does not support a 48 MHz crystal, only an external oscillator. Note, this must be enabled first during the initialization routine of the POST if a 48 MHz oscillator is used.

6.2.10 VERSION

The VERSION command checks to see if the controller is an enhanced type (82077, 82077AA, 82077SL) or the older type (8272A/765A). A value of 90H is returned as the result byte, defining an enhanced FDD controller is in use. No interrupts are generated.

6.2.11 RELATIVE SEEK

The command is coded the same as for SEEK, except for the MSB of the first byte and the DIR bit.

DIR Head Step Direction Control.

DIR	Action
0	Step Head Out
1	Step Head In

RCN Relative Cylinder Number that determines how many tracks to step the head in or out from the current track number.

The RELATIVE SEEK command differs from the SEEK command in that it steps the head the absolute number of tracks specified in the command instead of making a comparison against an internal register. The SEEK command is good for drives that support a maximum of 256 tracks. RELATIVE SEEKs cannot be overlapped with other RELATIVE SEEKs. Only one RELATIVE SEEK can be active at a time. Bit 4 of Status Register 0 (EC) will be set if RELATIVE SEEK attempts to step outward beyond Track 0.

As an example, assume that a floppy drive has 300 useable tracks and that the host needs to read track 300 and the head is on any track (0–255). If a SEEK

command was issued, the head would stop at track 255. If a RELATIVE SEEK command was issued, the 82078 would move the head the specified number of tracks, regardless of the internal cylinder position register (but would increment the register). If the head had been on track 40 (D), the maximum track that the 82078 could position the head on using RELATIVE SEEK, would be 296 (D), the initial track, +256 (D). The maximum count that the head can be moved with a single RELATIVE SEEK command is 256 (D).

The internal register, PCN, would overflow as the cylinder number crossed track 255 and would contain 40 (D). The resulting PCN value is thus (NCN + PCN) mod 256. Functionally, the 82078 starts counting from 0 again as the track number goes above 255(D). It is the users responsibility to compensate 82078 functions (precompensation track number) when accessing tracks greater than 255. The 82078 does not keep track that it is working in an "extended track area" (greater than 255). Any command issued would use the current PCN value except for the RECALIBRATE command which only looks for the TRACK0 signal. RECALIBRATE would return an error if the head was farther than 79 due to its limitation of issuing a maximum 80 step pulses. The user simply needs to issue a second RECALIBRATE command. The SEEK command and implied seeks will function correctly within the 44 (D) track (299–255) area of the "extended track area". It is the users responsibility not to issue a new track position that would exceed the maximum track that is present in the extended area.

To return to the standard floppy range (0–255) of tracks, a RELATIVE SEEK would be issued to cross the track 255 boundary.

A RELATIVE SEEK can be used instead of the normal SEEK but the host is required to calculate the difference between the current head location and the new (target) head location. This may require the host to issue a READ ID command to ensure that the head is physically on the track that software assumes it to be. Different 82078 commands will return different cylinder results which may be difficult to keep track of with software without the READ ID command.

6.2.12 DUMPREG

The DUMPREG command is designed to support system run-time diagnostics and application software development and debug. The command returns pertinent information regarding the internal status of the 82078. This can be used to verify the values initialized in the 82078.

2

6.2.13 PERPENDICULAR MODE COMMAND

6.2.13.1 About Perpendicular Recording Mode

An added capability of the 82078 is the ability to interface directly to perpendicular recording floppy drives. Perpendicular recording differs from the traditional longitudinal method by orienting the magnetic bits vertically. This scheme packs in more data bits for the same area.

6.2.13.2 The Perpendicular Mode Command

The PERPENDICULAR MODE Command allows the system designers to designate specific drives as

Table 6-16. Effects of WGATE and GAP Bits

GAP	WGATE	MODE	VCO Low Time after Index Pulse	Length of Gap2 Format Field	Portion of Gap2 Written by Write Data Operation	Gap2 VCO Low Time for Read Operations
0	0	Conventional Mode	33 Bytes	22 Bytes	0 Bytes	24 Bytes
0	1	Perpendicular Mode (500 Kbps Data Rate)	33 Bytes	22 Bytes	19 Bytes	24 Bytes
1	0	Reserved (Conventional)	33 Bytes	22 Bytes	0 Bytes	24 Bytes
1	1	Perpendicular Mode (1 Mbps Data Rate)	18 Bytes	41 Bytes	38 Bytes	43 Bytes

NOTE:
When either GAP or WGATE bit is set, the current value of precompensation in the DSR is used.

Perpendicular recording drives. Data transfers between Conventional and Perpendicular drives are allowed without having to issue PERPENDICULAR MODE commands between the accesses of the two different drives, nor having to change write pre-compensation values.

With this command, the length of the Gap2 field and VCO enable timing can be altered to accommodate the unique requirements of these drives. Table 6-16 describes the effects of the WGATE and GAP bits for the PERPENDICULAR MODE command.

When both GAP and WGATE equal "0" the PERPENDICULAR MODE command will have the following effect on the 82078: 1) If any of the new bits D0, D1, D2, and D3 are programmed to "1" the corresponding drive will automatically be programmed for Perpendicular mode (i.e.: GAP2 being written during a write operation, the programmed Data Rate will determine the length of GAP2), and data will be written with 0 ns write pre-compensation. 2) any of the new bits (D0–D3) that are programmed for "0" the designated drive will be programmed for Conventional Mode and data will be written with the currently programmed write pre-compensation value. 3) Bits D0, D1, D2, and D3 can only be over written when the OW bit is written as a "1". The status of these bits can be determined by interpreting the eighth result byte of the DUMPREG Command. (Note: if either the GAP or WGATE bit is a "1", then bits D0–D3 are ignored.)

"Software" and "Hardware" RESET will have the following effects on the enhanced PERPENDICULAR MODE command:

1. "Software" RESETs (Reset via DOR or DSR registers) will only clear GAP and WGATE bits to "0", D3, D2, D1, and D0 will retain their previously programmed values.
2. "Hardware" RESETs (Reset via pin 32) will clear all bits (GAP, WGATE, D0, D1, D2, and D3) to "0" (All Drives Conventional Mode).

6.2.14 POWERDOWN MODE COMMAND

The POWERDOWN MODE command allows the automatic power management and enables the enhanced registers (EREG EN) of the 82078. The use of the command can extend the battery life in portable PC applications. To enable auto powerdown the command may be issued during the BIOS power on self test (POST).

This command includes the ability to configure the 82078 into the enhanced AT/EISA and PS/2 mode. In the enhanced PS/2 and Model 30 modes, this makes the PD and IDLE pin status visible in the DIR register. In the enhanced AT/EISA modes, this command extends the SRB and TDR register. These extended registers accommodate bits that give more information about floppy drive interface, allow for boot drive selection, and identify the values of the PD and IDLE status.

As soon as the command is enabled, a 10 ms or a 0.5 sec. (5 ms or 0.25 with 2Mbps tape mode) minimum powerup timer is initiated depending on whether the MIN DLY bit is set to 0 or 1. This timer is one of the required conditions that has to be satisfied before the part will enter auto powerdown. Any software reset will reinitialize the timer. The timer countdown is also extended by up to 10 ms if the data rate is changed during the timer's countdown. Without this timer 82078 would have been put to sleep immediately after 82078 is idle. The minimum delay gives software a chance to interact with 82078 without incurring an additional overhead due to recovery time.

The command also allows the output pins of floppy disk drive interface to be tri-stated or left unaltered during auto powerdown. This is done by the FDI TRI bit. In the default condition (FDI TRI = 0) the output pins of the floppy disk drive are tri-stated. Setting this bit leaves the interface unchanged from the normal state.

The results phase returns the values programmed for MIN DLY, FDI TRI and AUTO PD. The auto powerdown mode is disabled by a hardware reset. Software results have no effect on the POWERDOWN MODE command parameters.

6.2.15 PART ID COMMAND

This command can be used to identify the floppy disk controller as an enhanced controller. The first stepping of both versions of the 64 pin 82078 will yield 0x01 in the result phase of this command. Any future enhancements on these parts will be denoted by the 5 LSBs (0x01 to 0x1F).

6.2.16 OPTION COMMAND

The standard IBM format includes an index address field consisting of 80 bytes of GAP 4a, 12 bytes of the sync field, four bytes identifying the IAM and 50 bytes of GAP 1. Under the ISO format most of this preamble is not used. The ISO format allows only 32 bytes of GAP 1 after the index mark. The ISO bit in this command allows the 82078 to configure the data transfer commands to recognize this format. The MSBs in this command are reserved for any other enhancements made available to the user in the future.

6.2.17 SAVE COMMAND

The first byte corresponds to the values programmed in the DSR with the exception of CLK48. The DRATE1, DRATE0 used here are unmapped. The second byte is used for configuring the bits from the OPTION command. All future enhancements to the OPTION command will be reflected in this byte as well. The next nine result bytes are explained in the Parameter Abbreviations section after the command summary. The 13th byte is the value associated with the auto powerdown command. The disk status is used internally by 82078. There are two reserved bytes at the end of this command for future use.

This command is similar to the DUMPREG command but it additionally allows the user to read back the precompensation values as well as the programmed data rate. It also allows the user to read the values programmed in the auto powerdown command. The precompensation values will be returned as programmed in the DSR register. This command is used in conjunction with the Restore command should prove very useful for SMM power management. This command reserves the last two bytes for future enhancements.

6.2.18 RESTORE COMMAND

Using Restore with the Save command, allows the SMM power management to restore the 82078 to its original state after a system powerdown. It also serves as a succinct way to provide most of the initialization requirements normally handled by the system. The sequence of initializing the 82078 after a reset occurred and assuming a Save command was issued follows:

- Issue the Drive Spec command (if the design utilizes this command)
- Issue the Restore command (pass the 16 bytes retrieved previously during SAVE)

The Restore command will program the data rate and precompensation value via the DSR. It then restores the values normally programmed through the Configure, Specify, and Perpendicular commands. It also enables the previously selected values for the AUTO Powerdown command. The PCN values are set restored to their previous values and the user is responsible for issuing the seek and recalibrate commands to restore the head to the proper location. There are some drives that do not recalibrate in which case the Restore command will restore the previous state completely. The PDOSC bit is retrievable using the Save command, however, the system designer must set it correctly. The software must allow at least 20 μs to execute the Restore command. When using the BOOTSEL bits in the TDR, the user must restore or reinitialize these bits to their proper values.

6.2.19 FORMAT AND WRITE COMMAND

The format and write command is capable of simultaneously formatting and writing data to the diskette. It is essentially the same as the normal format command. With the exception that included in the execution for each sector is not only the C, H, R, and N but also the data transfer of N bytes. The D value is ignored. This command formats the entire track. High speed floppy diskette duplication can be done fast and efficiently with this command. The user can format the diskette and put data on it in a single pass. This is very useful for software duplication applications by reducing the time required to format and copy diskettes.

6.2.20 LOCK

The LOCK command is included to protect a system with long DMA latencies against older application software packages that can disable the 82078's FIFO.

NOTE:
This command should only be used by the system's FDC routines, and ISVs (Independent Software Vendors) should refrain from using it. If an ISV's application calls for having the 82078 FIFO disabled a CONFIGURE Command should be used to toggle the EFIFO (Enable FIFO) bit. ISV can determine the value of the LOCK bit by interpreting the eighth result byte of an DUMPREG Command.

The LOCK command defines whether EFIFO, FIFOTHR, and PRETRK parameters of the CONFIGURE command can be RESET by the DOR and DSR registers. When the LOCK bit is set to a "1" all subsequent "software" RESETs by the DOR and DSR registers will not change the previously set parameter values in the CONFIGURE command. When the LOCK bit is set to a "0" 'software" RESETs by the DOR or DSR registers will return these parameters to their default values. All "hardware" Resets will set the LOCK bit to a "0" value, and will return EFIFO, FIFOTHR, and PRETRK to their default values. A Status byte is returned immediately after issuing the command byte. This Status byte reflects the value of the Lock bit set by the command byte.

NOTE:
No interrupts are generated at the end of this command.

7.0 STATUS REGISTER ENCODING

The contents of these registers are available only through a command sequence.

7.1 Status Register 0

Bit No.	Symbol	Name	Description
7, 6	IC	Interrupt Code	00—Normal termination of command. The specified command was properly executed and completed without error. 01—Abnormal termination of command. Command execution was started, but was not successfully completed. 10—Invalid command. The requested command could not be executed. 11—Abnormal termination caused by Polling.
5	SE	Seek End	The 82078 completed a SEEK or RECALIBRATE command, or a READ or WRITE with implied seek command.
4	EC	Equipment Check	The TRK0 pin failed to become a "1" after: 1. 80 step pulses in the RECALIBRATE command. 2. The RELATIVE SEEK command causes the 82078 to step outward beyond Track 0.
3	—	—	Unused. This bit is always "0".
2	H	Head Address	The current head address.
1, 0	DS1, 0	Drive Select	The current selected drive.

7.2 Status Register 1

Bit No.	Symbol	Name	Description
7	EN	End of Cylinder	The 82078 tried to access a sector beyond the final sector of the track (255D). Will be set if TC is not issued after Read or Write Data Command.
6	—	—	Unused. This bit is always "0".
5	DE	Data Error	The 82078 detected a CRC error in either the ID field or the data field of a sector.
4	OR	Overrun/ Underrun	Becomes set if the 82078 does not receive CPU or DMA service within the required time interval, resulting in data overrun or underrun.
3	—	—	Unused. This bit is always "0".
2	ND	No Data	Any one of the following: 1. READ DATA, READ DELETED DATA command, the 82078 did not find the specified sector. 2. READ ID command, the 82078 cannot read the ID field without an error. 3. READ TRACK command, the 82078 cannot find the proper sector sequence.
1	NW	Not Writable	WP pin became a "1" while the 82078 is executing a WRITE DATA, WRITE DELETED DATA, or FORMAT TRACK command.
0	MA	Missing Address Mark	Any one of the following: 1. The 82078 did not detect an ID address mark at the specified track after encountering the index pulse from the INDX# pin twice. 2. The 82078 cannot detect a data address mark or a deleted data address mark on the specified track.

7.3 Status Register 2

Bit No.	Symbol	Name	Description
7	—	—	Unused. This bit is always "0".
6	CM	Control Mark	Any one of the following: 1. READ DATA command, the 82078 encounters a deleted data address mark. 2. READ DELETED DATA command, the 82078 encountered a data address mark.
5	DD	Data Error in Data Field	The 82078 detected a CRC error in the data field.
4	WC	Wrong Cylinder	The track address from the sector ID field is different from the track address maintained inside the 82078.
3	—	—	Unused. This bit is always "0".
2	—	—	Unused. This bit is always "0".
1	BC	Bad Cylinder	The track address from the sector ID field is different from the track address maintained inside the 82078 and is equal to FF hex which indicates a bad track with a hard error according to the IBM soft-sectored format.
0	MD	Missing Data Address Mark	The 82078 cannot detect a data address mark or a deleted data address mark.

7.4 Status Register 3

Bit No.	Symbol	Name	Description
7	—	—	Unused. This bit is always "0".
6	WP	Write Protected	Indicates the status of the WP pin.
5	—	—	Unused. This bit is always "1".
4	T0	TRACK 0	Indicates the status of the TRK0 pin.
3	—	—	Unused. This bit is always "1".
2	HD	Head Address	Indicates the status of the HDSEL pin.
1, 0	DS1, 0	Drive Select	Indicates the status of the DS1, DS0 pins.

8.0 COMPATIBILITY

The 82078 was designed with software compatibility in mind. It is a fully backwards compatible solution with the older generation 8272A and NEC765A/B disk controllers. The 82078 also implements on-board registers for compatibility with the Personal System/2s as well as PC/AT and PC/XT floppy disk controller subsystems. The 82078 is fully compatible with Intel's 386/486SL Microprocessor Superset. Upon reset, the 82078 samples IDENT0 and IDENT1 to determine PS/2, PC/AT or PS/2 Model 30 mode.

8.1 PS/2 vs AT vs Model 30 Mode

The 82078 operates in three different modes: PS/2, PC/AT, and Model 30. The 82078 is placed into the proper mode of operations upon Hardware RESET with the appropriate settings of the IDENT0 and IDENT1 pins.

8.2 Compatibility with the FIFO

The FIFO of the 82078 is designed to be transparent to non-FIFO disk controller software developed on the older generation 8272A standard. Operation of the 82078 FIFO can be broken down into two tiers of compatibility. For first tier compatibility, the FIFO is left in the default disabled condition upon a "Hardware" reset. In this mode the FIFO operates in a byte mode and provides complete compability with non-FIFO based software. For second tier compatibility, the FIFO is enabled via the CONFIGURE command. When the FIFO is enabled, it will temporarily enter a byte mode during the command and result phase of disk controller operation. This allows for compatible operation when interrogating the Main Status Register (MSR) for the purpose of transferring a byte at a time to or from the disk controller. For normal disk controller applications, the system designer can still take advantage of the FIFO for time critical data transfers during the execution phase and not create any conflicts with non-FIFO software during the command or result phase.

In some instances, use of the FIFO in any form has conflicted with certain specialized software. An example of a compatibility conflict using the FIFO is with software that monitors the progress of a data transfer during the execution phase. If the software assumed the disk controller was operating in a single byte mode and counted the number of bytes transferred to or from the disk controller to trigger some time dependent event on the disk media (i.e., head position over a specific data field), the same software will not have an identical time relationship if the FIFO is enabled. This is because the FIFO allows data to be queued up, and then burst transferred across the host bus. To accommodate software of this type, it is recommended that the FIFO be disabled.

8.3 Drive Polling

The 82078 supports the polling mode of the older generation 8272A. This mode is enabled upon a reset and can be disabled via the CONFIGURE command. This mode is supported for the sole purpose of providing backward compatibility with software that expects its presence.

The intended purpose of drive polling dates back to 8" drives as a means to monitor any change in status for each disk drive present in the system. Each of the drives is selected for a period of time and its READY signal sampled. After a delay, the next drive is selected. Since the 82078 does not support READY in this capacity (internally tied true), the polling sequence is only simulated and does not affect the drive select lines (DS0–DS3) when it is active. If enabled, it occurs whenever the 82078 is waiting for a command or during SEEKs and RECALIBRATEs (but not IMPLIED SEEKs). Each drive is assumed to be not ready after a reset and a "ready" value for each drive is saved in an internal register as the simulated drive is polled. An interrupt will be generated on the first polling loop because of the initial "not ready" status. This interrupt must be followed with a SENSE INTERRUPT STATUS command from the host to clear the interrupt condition for each of the four logical drives.

9.0 Programming Guidelines

Programming the 82078 is identical to any other 8272A compatible disk controller with the exception of some additional commands. For the new designer, it is useful to provide some guidelines on how to program the 82078. A typical disk operation involves more than issuing a command and waiting for the results. The control of the floppy disk drive is a low level operation that requires software intervention at different stages. New commands and features have been added to the 82078 to reduce the complexity of this software interface.

9.1 Command and Result Phase Handshaking

Before a command or parameter byte can be issued to the 82078, the Main Status Register (MSR) must be interrogated for a ready status and proper FIFO direction. A typical floppy controller device driver should contain a subroutine for sending command or

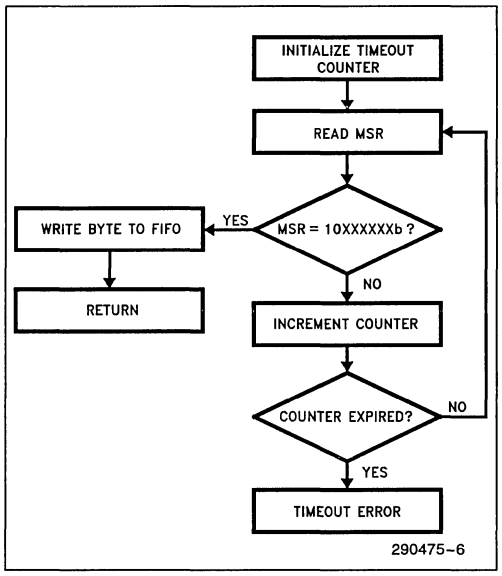


Figure 9-1. Send_Byte Routine

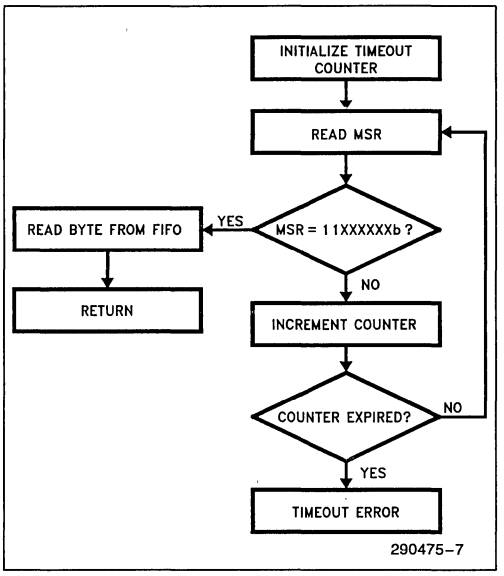


Figure 9-2. Get_Byte Routine

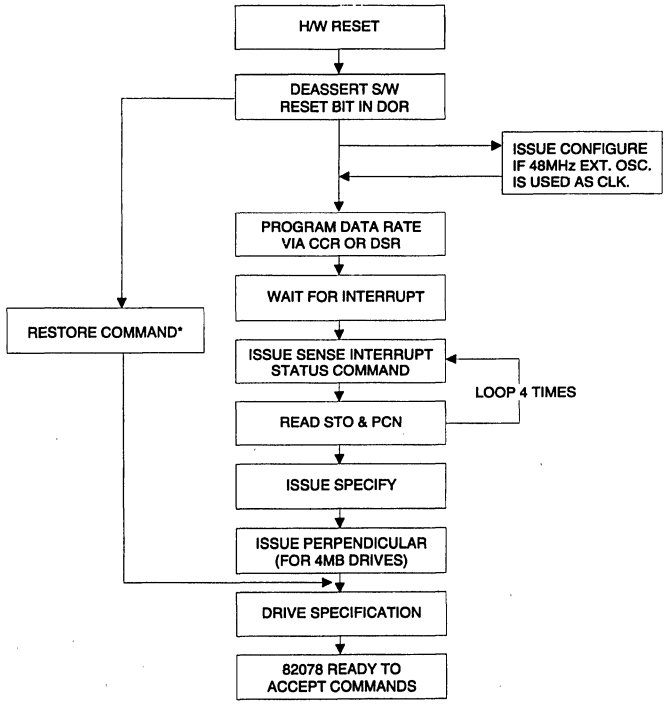
parameter bytes. For this discussion, the routine will be called "Send_byte" with the flowchart shown in Figure 9-1.

The routine loops until RQM is 1 and DIO is 0 indicating a ready status and FIFO direction is inward. If this condition is true, the 82078 is ready to accept a command or parameter byte. A timeout counter is used to insure software response within a reasonable amount of time in case of no response by the 82078. As a note, the programmer must be careful how the maximum delay is chosen to avoid unnecessary timeouts. For example, if a new command is issued when the 82078 is in the middle of a polling routine, the MSR will not indicate a ready status for the next parameter byte until the polling sequence completes the loop. This could cause a delay between the first and second bytes of up to 250 μs (@ 250 Kbps). If polling is disabled, this maximum delay is 175 μs. There should also be enough timeout margin to accommodate a shift of the software to a higher speed system. A timeout value that results in satisfactory operation on a 16 MHz CPU might fail when the software is moved to a system with a 25 MHz CPU. A recommended solution is to derive the timeout counter from a system hardware counter that is fixed in frequency from CPU clock to CPU clock.

For reading result bytes from the 82078, a similar routine is used. Figure 9-2 illustrates the flowchart for the routine "Get_byte". The MSR is polled until RQM is 1 and DIO is 1, which indicates a ready status and outward FIFO direction. At this point, the host can read a byte from the FIFO. As in the Send_byte routine, a timeout counter should be incorporated in case of a disk controller lock-up condition. For example, if a disk was not inserted into the disk drive at the time of a read operation, the controller would fail to receive the index pulse and lock-up since the index pulses are required for termination of the execution phase.

9.2 Initialization

Initializing the 82078 involves setting up the appropriate configuration after a reset. Parameters set by the SPECIFY command are undefined after a system reset and will need to be reinitialized. CONFIGURE command parameters default to a known state after a system reset but will need to be reinitialized if the system requirements are different from the default settings. This can be accomplished in two ways, either issue the individual commands, or issue the Restore command (assuming the Save command was issued). The Restore command is a succinct way to initialize the 82078, this is the preferable method if the system power management powers the 82078 on and off frequently. The flowchart for the recommended initialization sequence of the 82078 is shown in Figure 9-3.



*Sense Interrupt Status may be required if Restore CMD is not issued within 250 μs of trailing edge of H/W reset (@1 Mbps).

Figure 9-3. Initialization Flowchart

Following a reset of the 82078, the Configuration Control Register (CCR) should be reinitialized for the appropriate data rate. An external reset via the RESET pin will cause the data rate and write precompensation values to default to 250 Kbps (10b) and 125 ns (000b) respectively. Since the 125 ns write precompensation value is optimal for the 5 ¼" and 3 ½" disk drive environment, most applications will not require the value to be changed in the initialization sequence. As a note, a software reset issued via the DOR or DSR will not affect the data rate or write precompensation values. But it is recommended as a safe programming practice to always program the data rate after a reset, regardless of the type.

Since polling is enabled after a reset of the 82078, four SENSE INTERRUPT STATUS commands need to be issued afterwards to clear the status flags for each drive. The flowchart in Figure 9-3 illustrates how the software clears each of the four interrupt status flags internally queued by the 82078. It should be noted that although four SENSE INTERRUPT STATUS commands are issued, the INT pin is only active until the first SENSE INTERRUPT STATUS command is executed.

As a note, if the CONFIGURE command is issued within 250 μs of the trailing edge of reset (@1 Mbps), the polling mode of the 82078 can be disabled before the polling initiated interrupt occurs. Since polling stops when the 82078 enters the command phase, it is only time critical up to the first byte of the CONFIGURE command. If disabled in time, the system software no longer needs to issue the four SENSE INTERRUPT STATUS commands to clear the internal interrupt flags normally caused by polling.

The CONFIGURE command should also be issued if the system requirements are different from the default settings. For example, the CONFIGURE command can be used to enable the FIFO, set the threshold, and enable Implied Seeks.

The non-DMA mode flag, step rate (SRT), head load (HLT), and head unload times (HUT) programmed by the SPECIFY command do not default to a known state after a reset. This behavior is consistent with the 8272A and has been preserved here for compatibility. Thus, it is necessary to always issue a SPECIFY command in the initialization routine.

9.3 Recalibrates and Seeks

Commands that position the disk head are different from the typical READ/WRITE/FORMAT command in the sense that there is no result phase. Once a RECALIBRATE, SEEK, or RELATIVE SEEK command has been issued, the 82078 will return a ready status in the Main Status Register (MSR) and perform the head positioning operation as a background task. When the seek is complete, the 82078 will assert the INT signal to request service. A SENSE INTERRUPT STATUS command should then be asserted to clear the interrupt and read the status of the operation. Since the drive and motor enable signals are directly controlled through the Digital Output Register (DOR) on the 82078, a write to the DOR will need to precede the RECALIBRATE or SEEK command if the drive and motor is not already enabled. Figure 9-4 shows the flow chart for this operation.

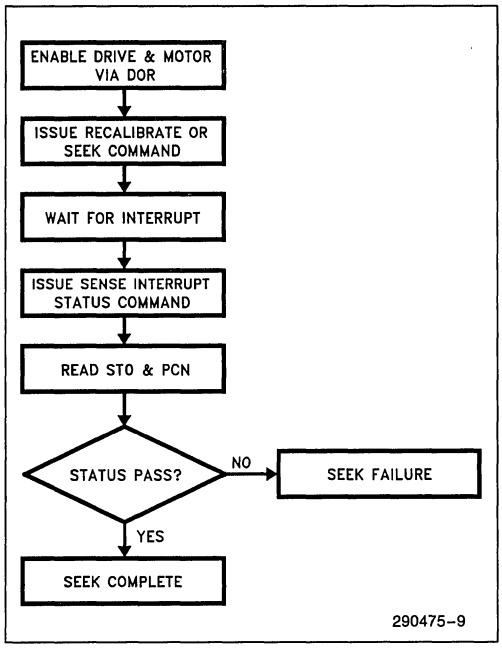


Figure 9-4. Recalibrate and Seek Operations

9.4 Read/Write Data Operations

A read or write data operation requires several steps to complete successfully. The motor needs to be turned on, the head positioned to the correct cylinder, the DMA controller initialized, the read or write command initiated, and an error recovery scheme implemented. The flowchart in Figure 9-5 highlights a recommended algorithm for performing a read or write data operation.

Before data can be transferred to or from the diskette, the disk drive motor must be brought up to speed. For most 3½" disk drives, the spin-up time is 300 ms, while the 5¼" drive usually requires about 500 ms due to the increased moment of inertia associated with the larger diameter diskette.

One technique for minimizing the motor spin-up delay in the read data case is to begin the read operation immediately after the motor is turned on. When the motor is not initially up to speed, the internal data separator will fail to lock onto the incoming data stream and report a failure in the status registers. The read operation is then repeated until successful status is obtained. There is no risk of a data integrity problem since the data field is CRC validated. But, it is not recommended to use this technique for the write data operation even though it requires successful reading of the ID field before the write takes place. The data separator performance of the 82078 is such that locking to the data stream could take place while the motor speed variation is still significant. This could result in errors when an attempt is made to read the disk media by other disk controllers that have a narrower incoming data stream frequency bandwidth.

After the motor has been turned on, the matching data rate for the media inserted into the disk drive should then be programmed to the 82078 via the Configuration Control Register (CCR). The 82078 is designed to allow a different data rate to be programmed arbitrarily without disrupting the integrity of the device. In some applications, it is required to automatically determine the recorded data rate of the inserted media. One technique for doing this is to perform a READ ID operation at each available data rate until a successful status is returned in the result phase.

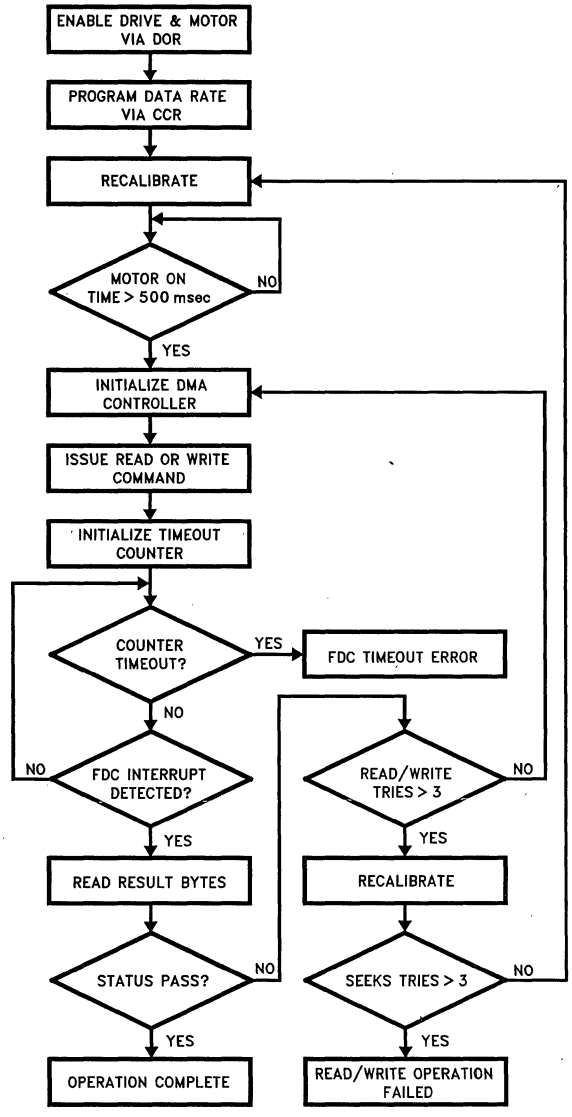


Figure 9-5. Read/Write Operation

If implied seeks are not enabled, the disk drive head must be positioned over the correct cylinder by executing a SEEK command. After the seek is complete, a head settling time needs to be asserted before the read or write operation begins. For most drives, this delay should be a minimum of 15 ms. When using implied seeks, the minimum head settling time can be enforced by the head load time (HLT) parameter designated in the SPECIFY command. For example, a HLT value of 8 will yield an effective head settling time of 16 ms for a programmed data rate of 500 Kbps. Of course if the head is already positioned over the correct cylinder, the head settling time does not need to be enforced.

The DMA controller is then initialized for the data transfer and the read or write command is executed. Typically the DMA controller will assert Terminal Count (TC) when the data transfer is complete. The 82078 will then complete the current data transfer and assert the INT signal signifying it has entered the result phase. The result phase can also be entered by the 82078 if an error is encountered or the last sector number equals the End of Track (EOT) parameter.

Based on the algorithm in Figure 9-5, if an error is encountered after reading the result bytes, two more retries are performed by reinitializing the DMA controller and re-issuing the read or write data command. A persisting failure could indicate the seek operation did not achieve proper alignment between the head and the track. The disk head should then be recalibrated and the seek repeated for a maximum of two more tries. Unsuccessful operation after this point should be reported as a disk failure to the operating system.

9.5 Formatting

The disk formatting procedure involves positioning the head on each track and creating a fixed format field used for organizing the data fields. The flowchart in Figure 9-6 highlights the typical format procedure.

After the motor has been turned on and the correct data rate programmed, the disk head is recalibrated to track 0. The disk is then allowed to come up to speed via a 500 ms delay. It is important the disk speed has stabilized before the actual formatting to avoid any data rate frequency variations. Since the format fields contain critical information used by the data separator of the disk controller for synchronization purposes, frequency stability of the data stream is imperative for media interchangeability among different systems.

The ID field data created on the disk during the format process is provided by the DMA controller during the execution phase. The DMA controller is initialized to send the C, H, R and N values for each sector ID field. For example, to format cylinder 7, on head 1, with 9 sectors, and a sector size of 2 (512 bytes), the DMA controller should be programmed to transfer 36 bytes (9 sectors × 4 bytes per sector) with the following data field: 7,1,1,2, 7,1,2,2, 7,1,3,2, ... 7,1,9,2. Since the values provided to the 82078 during the execution phase of the format command are directly recorded as the ID fields on the disk, the data contents can be arbitrary. Some forms of copy protection have been implemented by taking advantage of this capability.

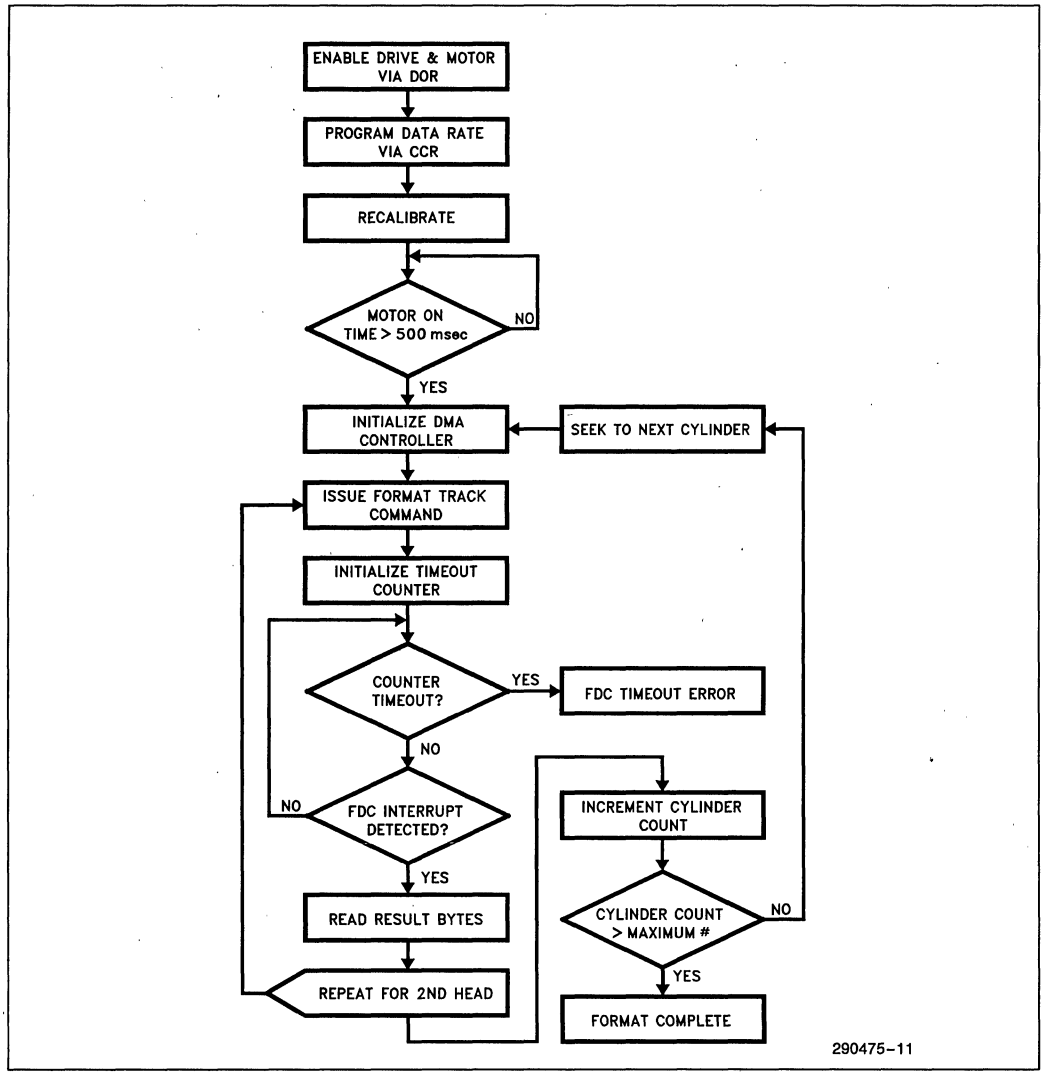


Figure 9-6. Formatting

After each head for a cylinder has been formatted, a seek operation to the next cylinder is performed and the format process is repeated. Since the FORMAT TRACK command does not have implied seek capability, the SEEK command must be used. Also, as discussed in Section 9.2, the head settling time needs to be adhered to after each seek operation.

9.6 Save and Restore

The Save and Restore commands were developed for portable systems that use zero-volt powerdown to conserve power. These systems turn off the V_{CC} to most of the system and retain the system status in a specific location. In older floppy controller designs, in order for system designers to retrieve the floppy controller status, a lot of separate commands and register reads were required. The Save command stores the key status information in a single command, the Restore command restores this information with a single command. These commands can be integrated into the SMM module that is responsible for zero-volt powerdown.

The sequence of initializing the 82078 after a reset occurred and assuming a Save command was issued follows:

- Issue the Drive Spec command (if the design utilizes this command)
- Issue the Restore command

The Restore command programs the data rate and precompensation value via the DSR. It then restores the values normally programmed through the Configure, Specify, and Perpendicular commands. It also enables the previously selected values for the AUTO Powerdown command. The command then restores the PCN values to its previous values. The user is responsible for issuing the seek and recalibrate commands to restore the head to the proper location. There are some drives that do not recalibrate in which case the Restore command will restore the previous state completely. The PDOSC bit is retrievable using the Save command, however it is up to the system designer to set it correctly. The software must allow at least 20 μs to execute the Restore command. When using the BOOTSEL bits in the TDR, the user must restore or reinitialize these bits to their proper values.

9.7 Verifies

In some applications, the sector data needs to be verified immediately after each write operation. One verify technique reinitializes the DMA controller to perform a read transfer or verify transfer (DACK# is asserted but not RD#) immediately after each write operation. Issue a read command to the disk controller and the resulting status indicates if the CRC validated the previously written data. This technique has the drawback of requiring additional software intervention by having to reprogram the DMA controller between each sector write operation. The 82078 supports this verify technique but also provides a VERIFY command that does not require the use of the DMA controller.

To verify a write data transfer or format track operation using the VERIFY command, the software simply issues the command with the same format as a READ DATA command but without the support of the DMA controller. The 82078 will then perform a disk read operation without a host data transfer. The CRC will be calculated for each sector read and compared against the value stored on the disk. When the VERIFY command is complete, the status register reports detected CRC errors.

9.8 Powerdown State and Recovery

The two power management modes coupled with the internal oscillator power management forms an important consideration for programming the 82078. The recovery of 82078 and the time it takes to achieve complete recovery depends on how 82078 is powered down and how it is awakened. The following sections describe all the programming concerns and subtleties involved in using power management features of the 82078.

9.8.1 OSCILLATOR POWER MANAGEMENT

Section 4.1 covers the power management scheme involved in powering down of both an internal and an external oscillator. Both types of oscillators face drop out effects and require recovery times on the order of tens of milliseconds (this may be objectionable to some application software). This means that if the oscillator is powered down then it is imperative for the software to assure enough time for the oscillator to recover to a stable state. Oscillator power management must be controlled by the system software especially to maintain software transparency. In cases where the system goes into a standby mode (by user request or system time-out), the power management software can turn off the oscillator to conserve power. This can also be controlled in hardware using the Powerdown (PD) pin. Complete recovery from an oscillator powerdown state requires the software to turn on the oscillator sufficiently ahead of awakening the 82078.

9.8.2 PART POWER MANAGEMENT

The part powerdown and wake up modes are covered in Section 4.2 in detail. This section is meant to address the programming concerns for the part (excluding the oscillator) during these modes.

9.8.2.1 Powerdown Modes

For both types of powerdown modes—DSR powerdown and auto powerdown, if reset is used to exit the part from powerdown then the internal microcontroller will go through a standard sequence: register initialization followed after some delay by an interrupt.

Software transparency in auto powerdown mode is preserved by MSR retaining the value of 80H which indicates that the part is ready to receive a command. This feature allows the part to powerdown while maintaining its responsiveness to any application software.

The PD and IDLE status bits can be monitored via the Status Register B (SRB, enhanced AT/EISA mode) and in the Digital Input Register (DIR, PS/2 and Model 30). Since the IDLE pin stays high when the 82078 is in idle state, the IDLEMSK bit can be used to set the pin low again (as part of a power management routine).

9.8.2.2 Wake Up Modes

Wake up from DSR powerdown results in the part being internally reset and all present status being lost. During DSR powerdown the RQM bit in the MSR is set. A software or hardware reset will wake up the part.

The case for wake up from auto powerdown is different. The BIOS and application software are very sensitive to delays involved in writing the first command bytes to the 82078. Most programs have short error time-outs in these cases. Such programs would not tolerate any floppy disk controller that was unable to receive the first byte of a command at any time. The following describes how 82078 uniquely sustains its software transparency during wake up sequences.

Prior to writing a command to 82078, it is first necessary to read the MSR to ensure that the 82078 is ready (RQM bit must be set) to receive the command. When the part detects a MSR read, it assumes that another command will follow and begins the wake up process. While the part is waking up it does not change the state of the MSR (MSR = 80H) and is able to receive the command in the FIFO. At this point one of the two following scenarios can occur.

No other command is sent subsequent to the MSR read. The part wakes up and initializes the minimum power up timer. Upon the expiration of this timer the part is once again put in powerdown state.

Another command follows the MSR read. If the command is sent during the part's recovery from powerdown, the part remembers the command, clears the RQM bit (to prevent further bytes being written) and acts on the command once it is fully awake.

If the MSR was not checked prior to writing of a command, the part will proceed as stated above with the RQM bit cleared and the command byte held until the internal microcontroller is ready. Writing the motor enable bits in DOR active will initiate the wake up sequence with RQM set high, ready to receive any command.

As it is clear from the above discussion, the immediate access to the floppy disk controller for the first command byte is vital to software transparency. The recovery of the part from powerdown may involve a delay after the first command byte has been issued. However, all programs have tolerance for the delay after the first command byte is issued. In a powered up chip, it is possible for the microcontroller to be in its "polling loop". As a result the tolerance for this delay provides an excellent window for recovery of the part.

10.0 DESIGN APPLICATIONS

10.1 Operating the 82078SL in a 3.3V Design

The design for 3.3V is the same as for 5.0V with two exceptions: The SEL3V# pin must be held low to select 3.3V operation, and the VCCF pin can be either 3.3V or 5.0V (VCCF can only be 5.0V when SEL3V# is high). The VCCF pin allows the controller to be operated in mixed (3.3V/5.0V) mode. For example, if the system operates at 3.3V and the floppy disk drive operates at 5.0V, the 82078 can be configured to operate at 3.3V with 5.0V available to the drive interface. See Figure 10-1 for a schematic.

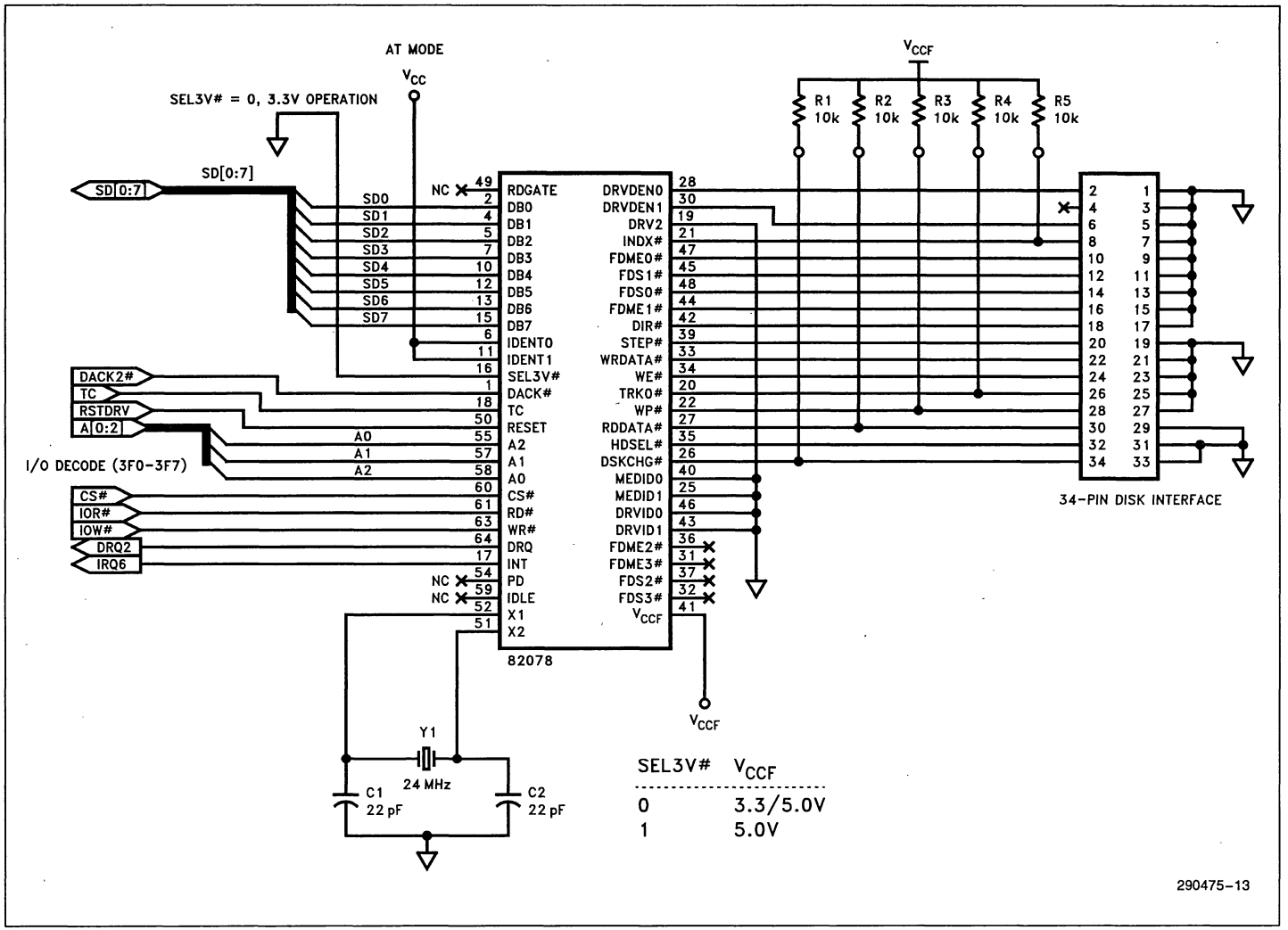


Figure 10-1. 82078SL 3.3V Design

10.2 Selectable Boot Drive

Generally a standard personal computer is configured with a 1.2 Mb 5.25″ disk drive and a 1.44 or 2.88 Mb 3.5″ disk drive. Usually the drive connects as "A:" and is the boot drive. At times the user may want to configure "B:" as the boot drive. Currently some BIOS' use a special implementation in software to accomplish this. The 82078 now offers this capability more efficiently by configuring the boot drives.

The 82078 allows for virtual drive designations. This is a result of allowing multiplexing the boot drive select and motor enable lines. This is shown in the Figure 10-2.

The DRIVE SEL1 and the DRIVE SEL2 bits in the DOR register decode internally to generate the signals DSn. The MEn signals generate directly from the DOR register. The DSn and MEn signals get mapped to actual FDSn and FDMEn pins based on the BOOTSELn bits (selected in the TDR register). The exact mapping of BOOTSEL vs. the FDSn and FDMEn pins is shown in the following table.

BOOTSEL1	BOOTSEL0	Mapping:
0	0	DS0 → FDS0, ME0 → FDME0 DS1 → FDS1, ME1 → FDME1 DS2 → FDS2, ME2 → FDME2
0	1	DS0 → FDS1, ME0 → FDME1 DS1 → FDS0, ME1 → FDME0 DS2 → FDS2, ME2 → FDME2
1	0	DS0 → FDS2, ME0 → FDME2 DS1 → FDS1, ME1 → FDME1 DS2 → FDS0, ME2 → FDME0
1	1	Reserved

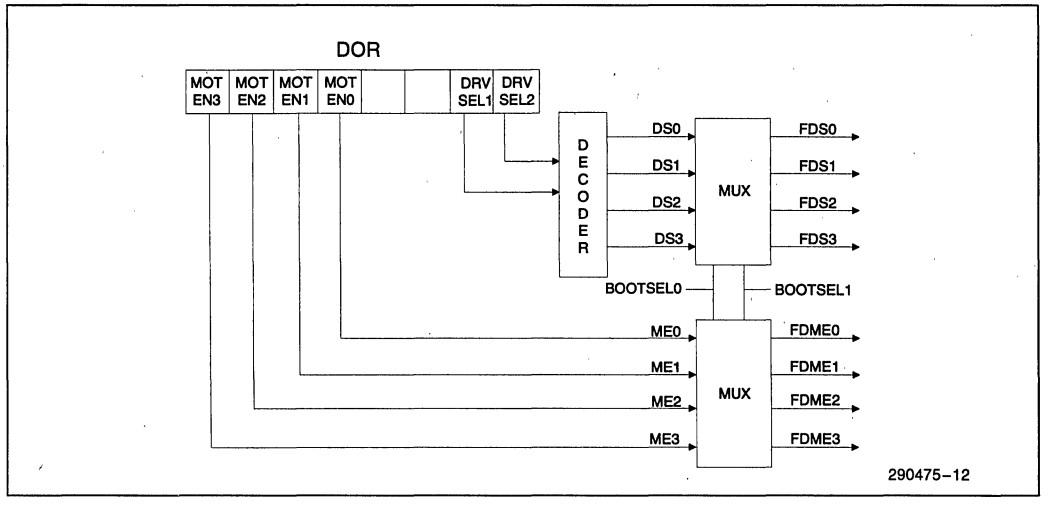


Figure 10-2. Virtual Drive Configuration

The BOOTSELn bits allow users to multiplex the output drive signals allowing different drives to be the boot drive. The DSn and MEn bits are considered virtual designations since the DSn and MEn signals get remapped to different corresponding physical FDSn and FDMEn pins. In other words, once the BOOTSELn bits are configured for a non-default selection, all future references made to the controller will be assumed as virtual designations. For example, if BOOTSEL1, BOOTSEL0 = 10 then DOR[1:0] = 00 refers to drive 2 and FDS2, FDME2 lines will be activated. Also, if TAPESEL[1:0] = 10, then tape mode is selected whenever FDS0, FDME0 are selected. Note, due to the virtual designations TAPESEL[1:0] = 00 would never enable tape mode due to boot drive restrictions.

10.3 How to Disable the Native Floppy Controller on the Motherboard

There are occasions when the floppy controller designed onto the motherboard of a system needs to be disabled in order to operate another floppy controller on the expansion bus. This can be done without changing the BIOS or remapping the address of the floppy controller (provided there is a jumper, or another way to disable the chip select on the native controller).

Upon reset, the DOR register in the 82078 is set to 00H. If the CS# is left enabled during the POST, the DOR is set to 0CH, this enables the DMA GATE# bit in the DOR. When this bit is set the 82078 treats a DACK# and a RD# or WR# as an internal chip select (CS#). Bus contention will occur between the native controller and the auxiliary controller if the DMA GATE# bit becomes active, even if the CS# signal is not present.

The proper way to disable the native floppy controller is to disable the CS# before the system is turned on. This will prevent the native controller from getting initialized. Another option is to map the native controller to a secondary address space, then disable the DMA GATE# via the DOR disabling the DMA GATE#. This assumes that the native controller is switchable to a secondary address space.

10.4 Replacing the 82077SL with a 82078 in a 5.0V Design

The 82078 easily replaces the 5.0V 82077SL with minimum design changes. With a few exceptions, most of the signals are named as they were in the 82077SL. Some pins were eliminated and other renamed to accommodate a reduced pin count and smaller package.

The connections to the AT bus are the same as the 82077SL with the following exceptions: MFM and IDENT have been replaced by IDENT1 and IDENT0. The PLL0 pin was removed. Configure the tape drive mode on the 82078 via the Tape Drive Register (TDR).

The Drive Interface on the 82078 is also similar to the 82077SL except as noted: DRVDEN0 and DRVDEN1 on the 82078 take the place of DENSEL, DRATE0, and DRATE1 on the 82077SL. The Drive Specification Command configures the polarity of these pins, thus selecting the density type of the drive. The Motor Enable pins (ME0-3) and the Drive Select pins (DS0-3) are renamed FDME(0-3) and FDS(0-3) respectively on the 82078. 10K pull-up resistors can be used on the disk interface. See Figure 10-3 for a schematic of the connection.

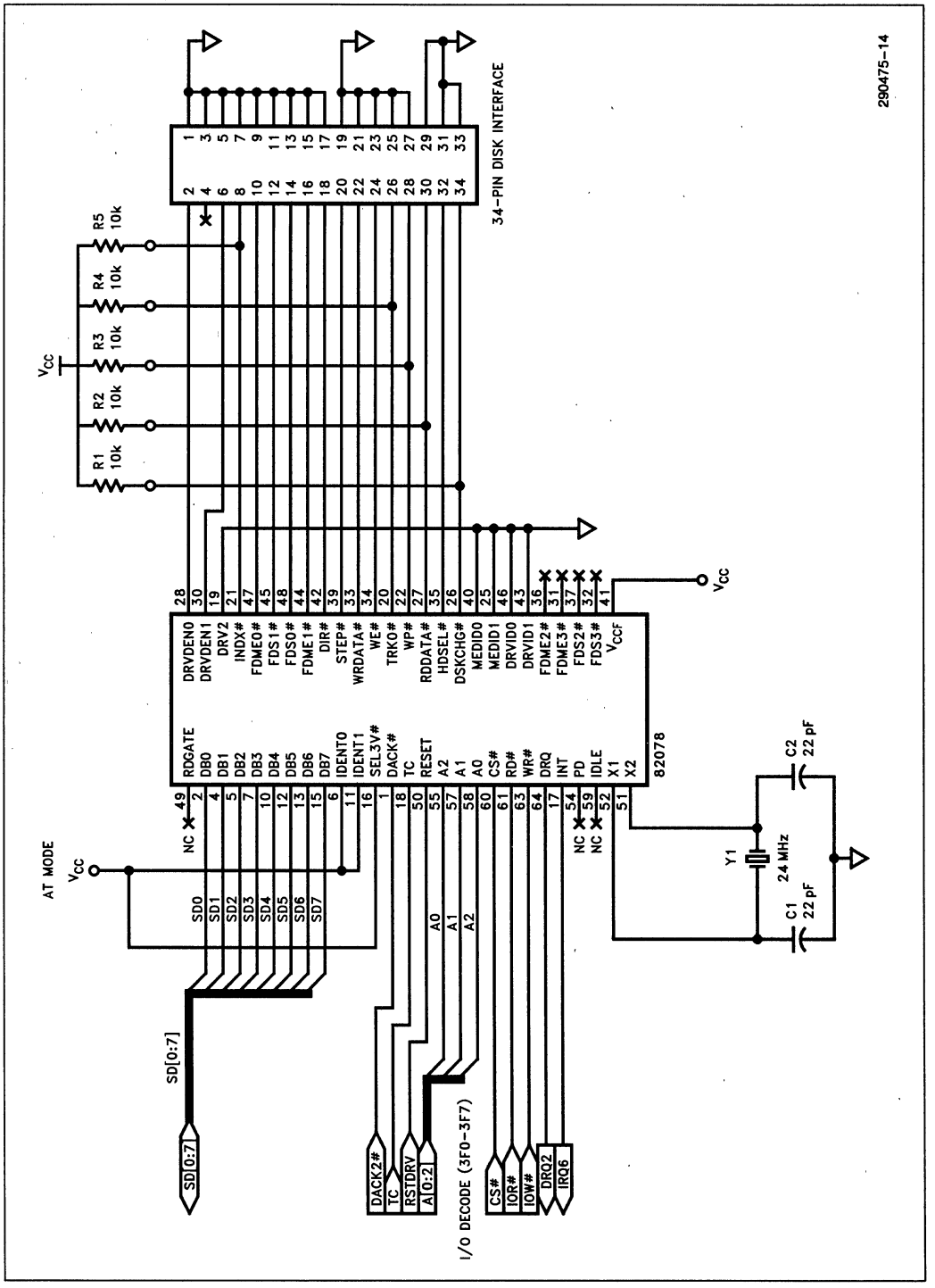


Figure 10-3. 82077SL Conversion to 82078-1

Pin Changes on the 64 Pin Part:

- INVERT# is removed
- 4 NC's (no connects) are removed
- MFM, IDENT pins on the 82077SL have been changed to IDENT1 and IDENT0 respectively.
- PLL0 pin, which allowed for H/W configuration of tape drive mode is no longer available. Tape mode can be configured via the TDR register.
- DENSEL, DRATE1, DRATE0 pins have been substituted by DRVDEN0, DRVDEN1. The Drive Specification command can be used to configure these pins for various requirements of drives available on the market.
- RDGATE has been added and can be used for diagnostics of the PLL.
- MEDID1, MEDID0 are new, they return media type information to the TDR register.
- DRVID1, DRVID0 return drive type information to the TDR register.
- SEL3V# selects between either 3.3V or 5V mode. Connecting the pin LOW selects 3.3V mode.
- 5 VSS pins, 2 VCC pins, 2 VSSP pins, 1 VCCF pin, and 1 AVCC and 1 AVSS pin.
- VCCF can be used to interface a 5.0V or a 3.3V drive to the 82078 (when SEL3V# is low).
- The Hardware RESET pulse width has changed from 170 times the oscillator period to 100 ns plus 25 times the oscillator period.

11.0 D.C. SPECIFICATIONS

11.1 Absolute Maximum Ratings

Storage Temperature	−65°C to +150°C
Supply Voltage	−0.5V to +8.0V
Voltage on Any Input	GND −2V to 6.5V
Voltage on Any Output	GND −0.5V to V_{CC} +0.5V
Power Dissipation	1W

11.2 D.C. Characteristics T_A = 0°C to 70°C, V_{SS} = AV_{SS} = 0V

64 PIN D.C. CHARACTERISTICS

Symbol	Parameter	V_{CC} = +5V ±10%(7)			V_{CC} = 3.3V ± 0.3V		
		Min(V)	Max(V)	Test Conditions	Min(V)	Max(V)	Test Conditions
V_{ILC}	Input Low Voltage, X1	−0.5	0.8		−0.3	0.8	
V_{IHC}	Input High Voltage, X1	3.9	V_{CC} + 0.5		2.4	V_{CC} + 0.3	
V_{IL}	Input Low Voltage (all pins except X1)	−0.5	0.8		−0.3	0.8	
V_{IH}	Input High Voltage (all pins except X1)	2.0	V_{CC} + 0.5		2.0	V_{CC} + 0.3	
V_{OL}(8)	System Interface		0.45	I_{OL} = 12 mA		0.45	I_{OL} = 6 mA
	FDD Interface outputs		0.45	I_{OL} = 24 mA		0.45	I_{OL} = 12 mA
	Status Outputs (Note 6)		0.45	I_{OL} = 4 mA		0.45	I_{OL} = 4 mA
V_{OH}	All outputs	3.0		I_{OH} = −4.0 mA	2.4		I_{OH} = −2.0 mA
	All outputs	V_{CC} − 0.4		I_{OH} = −100 µA	V_{CC} − 0.2		I_{OH} = −100 µA

64 PIN D.C. CHARACTERISTICS (I_{CC})

Symbol	Parameter	V_{CC} = +5V ±10%(7)			V_{CC} = 3.3V ± 0.3V		
		Typ	Max(A)	Test Conditions	Typ	Max(A)	Test Conditions
I_{CC1}	1 Mbps Data Rate V_{IL} = V_{SS}, V_{IH} = V_{CC}	15.4 mA	25 mA	(Notes 1, 2, 5)	8.4 mA	16 mA	(Notes 1, 2)
I_{CC2}	1 Mbps Data Rate V_{IL} = 0.45V, V_{IH} = 2.4V	20.8 mA	30 mA	(Notes 1, 2, 5)	8.6 mA	16 mA	(Notes 1, 2)
I_{CC3}	500 Kbps Data Rate V_{IL} = V_{SS}, V_{IH} = V_{CC}	11.8 mA	20 mA	(Notes 1, 2)	6.2 mA	14 mA	(Notes 1, 2)
I_{CC4}	500 Kbps Data Rate V_{IL} = 0.45V, V_{IH} = 2.4V	17.6 mA	25 mA	(Notes 1, 2)	6.2 mA	14 mA	(Notes 1, 2)
I_{CCSB}	I_{CC} in Powerdown	0 µA	60 µA	(Notes 3, 4)	0 µA	60 µA	(Notes 3, 4)

64 PIN D.C. CHARACTERISTICS (I_{CC}) (Continued)

Symbol	Parameter	V_{CC} = +5V ±10%(7)			V_{CC} = 3.3V ± 0.3V		
		Typ	Max(A)	Test Conditions	Typ	Max(A)	Test Conditions
I_{IL}	Input Load Current (all input pins)		10 μA −10 μA	$V_{IN} = V_{CC}$ $V_{IN} = 0V$		10 μA −10 μA	$V_{IN} = V_{CC}$ $V_{IN} = 0V$
I_{OFL}	Data Bus Output Float Leakage		±10 μA	0.45 < V_{OUT} < V_{CC}		±10 μA ±10 μA	0.45 < V_{OUT} < V_{CC}

NOTES:
1. Only the data bus inputs may float.
2. Tested while reading a sync field of "00". Outputs not connected to D.C. loads.
3. $V_{IL} = V_{SS}$, $V_{IH} = V_{CC}$; Outputs not connected to D.C. loads.
4. Typical value with the oscillator off.
5. I_{CC} for 2 Mbps Data Rate: Max 40 mA (TTL), 35 mA (CMOS) at 5.5V, typical 29.2 mA (TTL) and 24.4 (CMOS).
6. Status outputs are PD, IDLE, and RDGATE.
7. V_{CC} and V_{CCF} for the 82078-1 is +5V ±5%.
8. V_{OL} change effective for both 44-pin and 64-pin package offerings.

2

64 PIN MIXED MODE D.C. CHARACTERISTICS

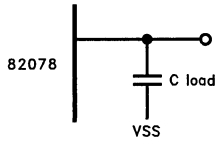
Symbol	Parameter	V_{CC} = 3.3V ±0.3V, V_{CCF} = +5V ±10%(7)		
		Min(V)	Max(V)	Test Conditions
V_{ILC}	Input Low Voltage, X1	−0.3	0.8	
V_{IHC}	Input High Voltage, X1	2.4	V_{CC} + 0.3	
V_{IL}	Input Low Voltage (system pins except X1) (floppy drive interface pins)	−0.3 −0.5	0.8 0.8	
V_{IH}	Input High Voltage (system interface pins except X1) (floppy drive interface pins)	2.0 2.0	V_{CC} + 0.3 V_{CC} + 0.5	
V_{OL}	System Interface		0.4	I_{OL} = 6 mA
	FDD Interface outputs		0.4	I_{OL} = 24 mA
	Status Pins: IDLE, PD, RDGATE		0.4	I_{OL} = 4 mA
V_{OH}	All system outputs	2.4		I_{OH} = −2.0 mA
	All FDD interface outputs	3.0		I_{OH} = −4.0 mA
	All system outputs	V_{CC} − 0.2		I_{OH} = −100 μA
	All FDD interface outputs	V_{CC} − 0.4		I_{OH} = −100 μA

CAPACITANCE

C_{IN}	Input Capacitance	10	pF	F = 1 MHz, T_A = 25°C
C_{IN1}	Clock Input Capacitance	20	pF	Sampled, not 100% Tested
$C_{I/O}$	Input/Output Capacitance	20	pF	

NOTE:
All pins except pins under test are tied to AC ground.

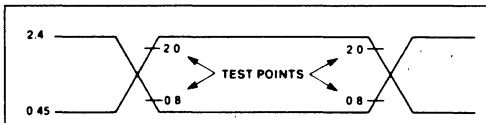
LOAD CIRCUIT



290475-15

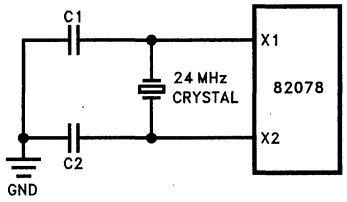
C_{load} = 50 for all logic outputs, 100 pF for the data bus.

A.C. TESTING INPUT, OUTPUT WAVEFORM

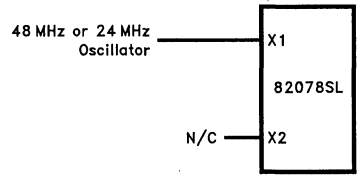


290475-16

11.3 Oscillator



290475-17



290475-18

The 24 MHz clock can be supplied either by a crystal or a MOS level square wave. All internal timings are referenced to this clock or a scaled count which is data rate dependent.

The crystal oscillator must be allowed to run for 10 ms after V_{CC} has reached 4.5V or exiting the POWERDOWN mode to guarantee that it is stable.

Frequency:	24 MHz ±0.1%
Mode:	Parallel Resonant Fundamental Mode
Series Resistance:	Less than 40Ω
Shunt Capacitance:	Less than 5 pF

12.0 A.C. SPECIFICATIONS

T_A = 0°C to 70°C, V_{CC} = +5V ±10%[17], +3.3V ±0.3V, V_{SS} = AV_{SS} = 0V

Symbol	Parameter	Min	Max	Unit
CLOCK TIMINGS				
t1	Clock Rise Time		10	ns
	Clock Fall Time		10	ns
t2	Clock High Time[7]	16	26	ns
t3	Clock Low Time[7]	16	26	ns
t4	Clock Period	41.66	41.66	ns
t5	Internal Clock Period[3]			
HOST READ CYCLES				
t7	Address Setup to RD#	5		ns
t8	RD# Pulse Width	90		ns
t9	Address Hold from RD#	0		ns
t10	Data Valid from RD#[12]		80	ns
t11	Command Inactive	60		ns
t12	Output Float Delay		35	ns
t13	INT Delay from RD#[16]		t5 + 125	ns
t14	Data Hold from RD#	5		ns
HOST WRITE CYCLES				
t15	Address Setup to WR#	5		ns
t16	WR# Pulse Width	90		ns
t17	Address Hold from WR#	0		ns
t18	Command Inactive	60		ns
t19	Data Setup to WR#	70		ns
t20	Data Hold from WR#	0		ns
t21	INT Delay from WR#[16]		t5 + 125	ns
DMA CYCLES				
t22	DRQ Cycle Period[1]	6.5		µs
t23	DACK# to DRQ Inactive		75	ns
t23a	DRQ to DACK# Inactive	(Note 15)		ns
t24	RD# to DRQ Inactive[4]		100	ns
t25	DACK# Setup to RD#, WR#	5		ns
t26	DACK# Hold from RD#, WR#	0		ns
t27	DRQ to RD#, WR# Active[1]	0	6	µs
t28	Terminal Count Width[10]	50		ns
t29	TC to DRQ Inactive		150	ns
RESET				
t30	"Hardware" Reset Width[5]	1.13		µs
t30a	"Software" Reset Width[5]	(Note 11)		ns
t31	Reset to Control Inactive		2	µs

A.C. SPECIFICATIONS

T_A = 0°C to 70°C, V_{CC} = +5V ±10%[17], +3.3V ±0.3V, V_{SS} = AV_{SS} = 0V (Continued)

Symbol	Parameter	Min	Max	Unit
WRITE DATA TIMING				
t32	Write Data Width[6]			ns
DRIVE CONTROL				
t35	DIR# Setup to STEP#[14]	1.0		µs
t36	DIR# Hold from STEP#	10		µs
t37	STEP# Active Time (High)	2.5		µs
t38	STEP# Cycle Time[2]			µs
t39	INDEX# Pulse Width	5		t5
t41	WE# to HDSEL# Change	(Note 13)		ms
READ DATA TIMING				
t40	Read Data Pulse Width	50		ns
t44	82078-1		2M	bits/sec
	82078SL		1M	bits/sec
t44	Data Rate Period = 1/f44			
tLOCK	Lockup Time		64	t44

NOTES:

1. This timing is for FIFO threshold = 1. When FIFO threshold is N bytes, the value should be multiplied by N and subtract 1.5 µs. The value shown is for 1 Mbps, scales linearly with data rate.
2. This value can range from 0.5 ms to 8.0 ms and is dependent upon data rate and the Specify command value.
3. Many timings are a function of the selected data rate. The nominal values for the internal clock period (t5) for the various data rates are:

2 Mbps	1.5x oscillator period = 62.5 ns
1 Mbps	3x oscillator period = 125 ns
500 Kbps	6x oscillator period = 250 ns
300 Kbps	10x oscillator period = 420 ns
250 Kbps	12x oscillator period = 500 ns

4. If DACK# transitions before RD#, then this specification is ignored. If there is no transition on DACK#, then this becomes the DRQ inactive delay.
5. Reset requires a stable oscillator to meet the minimum active period.

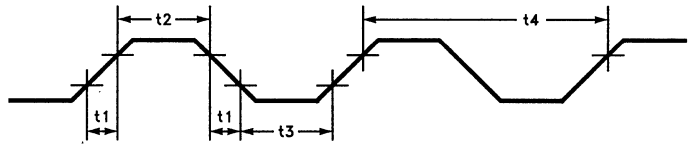
6. Based on the internal clock period (t5). For various data rates, the Write Data Width minimum values are:
   - 2 Mbps 2.5x oscillator period −50 ns = 75 ns
   - 1 Mbps 5x oscillator period −50 ns = 150 ns
   - 500 Kbps 10x oscillator period −50 ns = 360 ns
   - 300 Kbps 16x oscillator period −50 ns = 615 ns
   - 250 Kbps 19x oscillator period −50 ns = 740 ns
7. Test points for clock high time are 3.5V. Due to transitional times, clock high time max and clock low time max cannot be met simultaneously. Clock high time min and clock low time max can not be met simultaneously.
8. Based on internal clock period (t5).
9. Jitter tolerance is defined as:
   (Maximum bit shift from nominal position ÷ ¼ period of nominal data rate) × 100% is a measure of the allowable bit jitter that may be present and still be correctly detected. The data separator jitter tolerance is measured under dynamic conditions that jitters the bit stream according to a reverse precompensation algorithm.
10. TC width is defined as the time that both TC and DACK# are active. Note that TC and DACK# must overlap at least 50 ns.
11. The minimum reset active period for a software reset is dependent on the data rate, after the 82078 has been properly reset using the t30 spec. The minimum software reset period then becomes:
    - 2 Mbps 1.5 x t4 = 62.5 ns
    - 1 Mbps 3 x t4 = 125 ns
    - 500 Kbps 6 x t4 = 250 ns
    - 300 Kbps 10 x t4 = 420 ns
    - 250 Kbps 12 x t4 = 500 ns
12. Status Register's status bits which are not latched may be updated during a Host read operation.
13. The minimum MFM values for WE to HDSEL change (t41) for the various data rates are:
    - 2 Mbps 0.5 ms + [4 x GPL]
    - 1 Mbps 0.5 ms + [8 x GPL]
    - 500 Kbps 1.0 ms + [16 x GPL]
    - 300 Kbps 1.6 ms + [26.66 x GPL]
    - 250 Kbps 2.0 ms + [32 x GPL]

    GPL is the size of gap 3 defined in the sixth byte of a Write Command.
14. This timing is a function of the selected data rate as follows:
    - 2 Mbps 0.5 μs Min
    - 1 Mbps 1.0 μs Min
    - 500 Kbps 2.0 μs Min
    - 300 Kbps 3.3 μs Min
    - 250 Kbps 4.0 μs Min
15. This timing is a function of the internal clock period (t5) and is given as (⅔) t5. The values of t5 are shown in Note 3.
16. The timings t13 and t21 are specified for INT signal in the polling mode only. These timings in case of the result phase of the read and write commands are microcode dependent.

Part Specification	3.3V	5.0V	2 Mbps Data Rate
82078SL	X	X	
82078-1		X	X

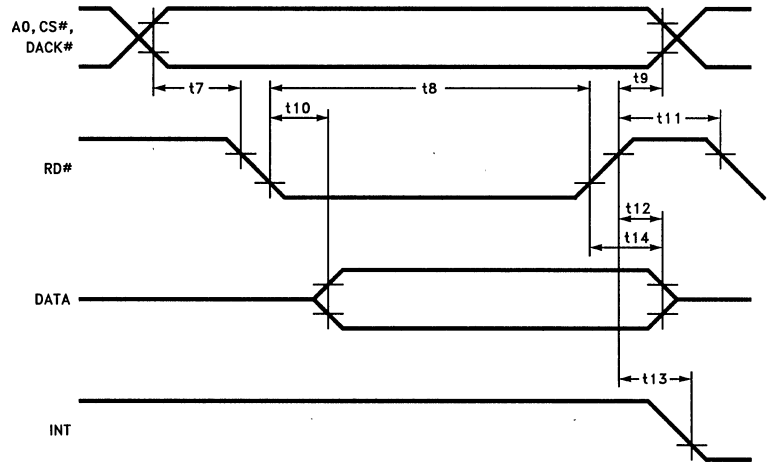
17. For 82078-1 only, V_{CC} and V_{CCF} requirements are +5V ±5%.

Clock Timings



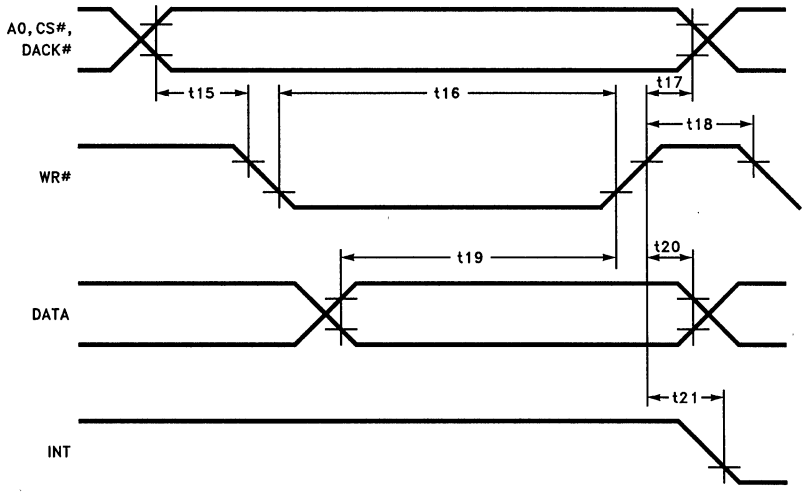
290475-19

Host Read Cycles



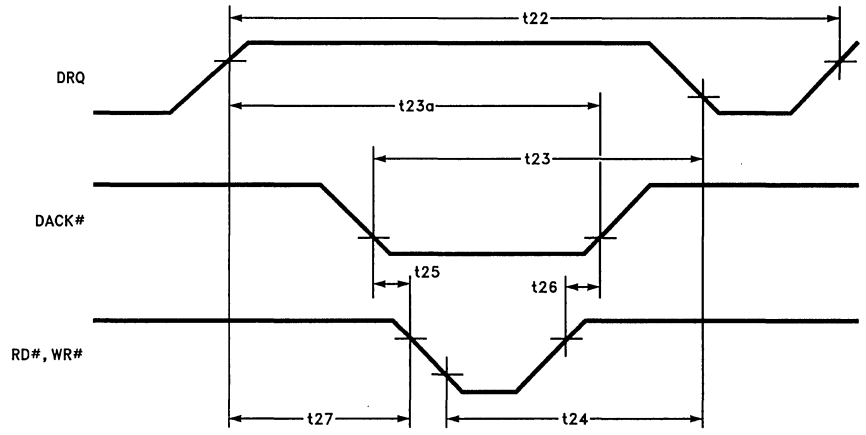
290475-20

Host Write Cycles



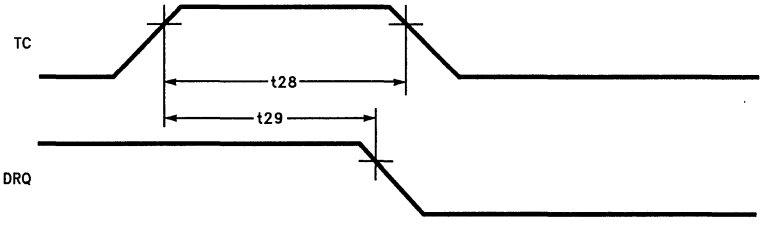
290475-21

DMA Cycles



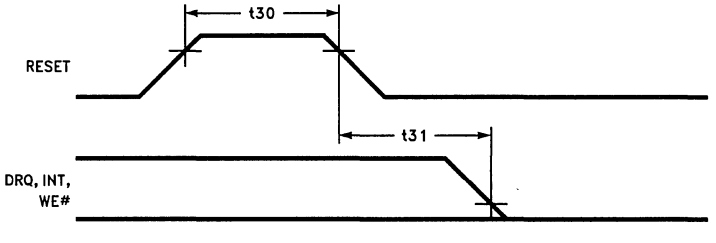
290475-22

Terminal Count



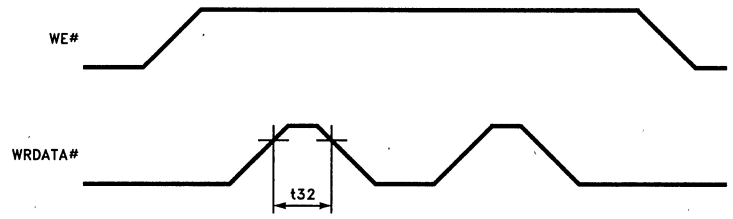
290475-23

Reset



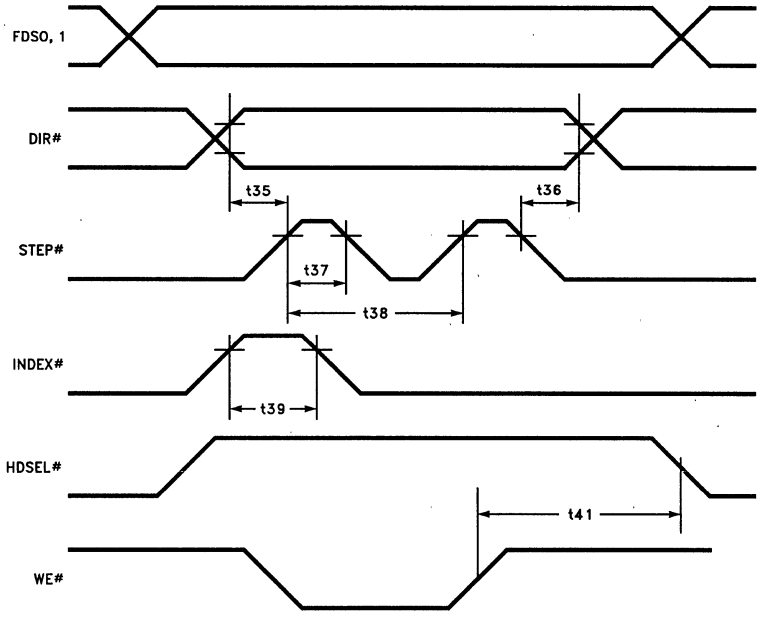
290475-24

Write Data Timing



290475-25

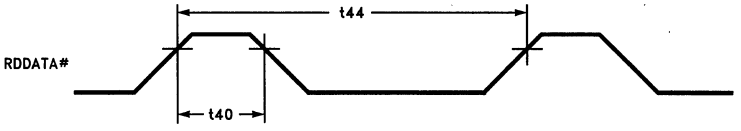
Drive Control



290475-26

NOTE:
For overlapped seeks, only one step pulse per drive selection is issued. Non-overlapped seeks will issue all programmed step pulses.

Internal PLL

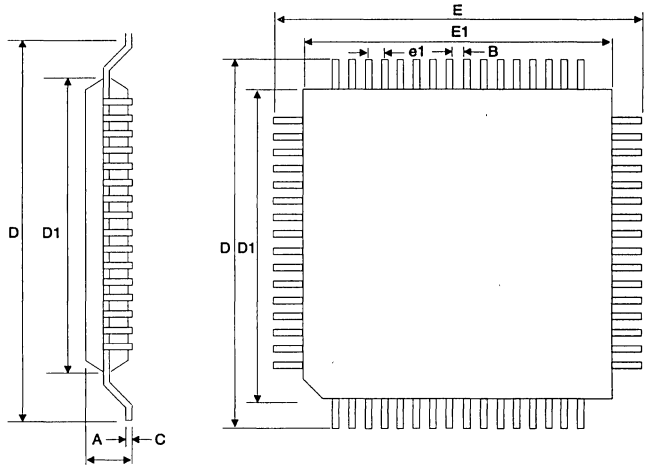


290475-27

12.1 Package Outline for the 64 QFP Part

The 82078 addresses the current need of the smaller and thinner packages, for the current market. The size of the part is becoming increasingly important in the portable computer market. The QFP part considerably reduces the real estate consumed. The package outline, with the appropriate dimensions is given below:

64-Lead Quad Flatpack Package



290475-28

Description	Symbol	64 QFP Package	
		Nominal (mm)	Tolerance (mm)
Overall Height	A	2.35	±0.20
Stand Off	A1	0.15	±0.10
Lead Width	B	0.30	±0.10
Lead Thickness	C	0.15	±0.05
Terminal	D	15.3	±0.40
Long Side	D1	12.0	±0.10
Terminal	E	15.3	±0.40
Short Side	E1	12.0	±0.10
Lead Spacing	e1	0.65	±0.12
Lead Count	N	64	

13.0 REVISION HISTORY FOR THE 82078 64 PIN

The following list represents the key differences between version 002 and version 003 of the 82078 64 pin data sheet.

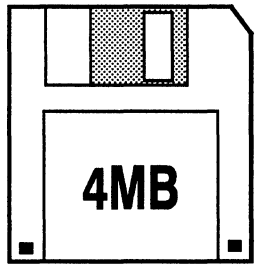
Section 5.2.3	Redundant information removed.
Section 5.2.4	Redundant information removed.
Section 8.0	Description of IDENT0 and IDENT1 changed to clarify their function.
Section 11.2	New Vol specification added for status pins.
Table 6-2	Data in table reordered to be consistent.
AC Specifications	V_{CC} has changed for 82078-1 only.

intel®

AP-358

APPLICATION NOTE

Intel 82077SL for Super Dense Floppies



292093-1

KATEN A. SHAH
APPLICATION ENGINEER

September 1992

Order Number: 292093-002

Intel 82077SL for Super Dense Floppies

CONTENTS

CONTENTS

INTRODUCTION

The evolution of the floppy has been marked in little over a decade by a significant increase in capacity accompanied by a noticeable decrease in the form factor from the early 8 inch floppy disks to the present day 3.5 inch floppy disks. This decade will also be remarkable as OEMs adopt "Super" dense floppies.

The most commonly seen floppies today are invariably one of the form factors – the 5.25″ or the 3.5″. Each form factor has several associated capacity ranges. The 5.25″ floppies available are: 180 KB (single density), 360 KB (double density) and 1.2 MB (high density). The 3.5″ floppies available are: 720 KB (double density) and 1.44 MB (high density). The emerging super dense floppies will evolve on the installed base of 3.5″ floppies. The latest member of this set is the 2.88 MB (extra density) floppy, pioneered by Toshiba. The cornerstone of market acceptance of newer drives is compatibility to the older family. The 2.88 MB (formatted) floppy drive allows the user to format, read from and write to the lower density diskettes.

As programs and data files get bigger, the demand for higher capacity floppies becomes obvious. There are several 3.5″ higher density drives available from various vendors with capacities well into the 20 MB range. NEC has introduced a 13 MB drive and companies such as Insite have introduced 20 MB drives. Both drives require servo-mechanisms to accurately position the head over the right track. NEC's drive has the standard floppy drive interface whereas Insite's interface is SCSI based. The market for these floppy drives will remain a niche unless they receive more OEM support.

Initiated by Toshiba's research and innovation of the higher density 4 MB floppy disk media, the market is headed towards the super dense floppy drive. After IBM's endorsement of the 4 MB (unformatted) floppy disk drives on their PS/2 model 57 and PS/2 model 90, several OEMs have shown a growing interest in "super" dense floppy disk drives. The latest DOS 5.0 supports the new 4 MB floppy media and BIOS vendors like Pheonix, AMI, Award, Quadtel, System Soft, and Microid all support the newer 4 MB floppy media.

PURPOSE

An important consideration to implement the 4 MB floppy drive is the floppy disk controller. Intel's highly integrated floppy disk controller, 82077AA/SL, has led the market in supporting the 4 MB floppy drive. Two ingredients are necessary to fully support these drives: 1 Mbps transfer rate and the perpendicular recording mode. This paper deals with a discussion of what the perpendicular mode is and how can a 4 MB floppy disk drive be implemented in a system using the 82077AA/SL.

PERPENDICULAR RECORDING MODE

Toshiba has taken the 2 MB floppy and doubled the storage capacity by doubling the number of bits per track. Toshiba achieved this by an innovative magnetic recording mode, called the vertical or the perpendicular recording mode. This mode utilizes magnetization perpendicular to the recording medium plane. This is in contrast to the current mode of longitudinal recording which uses the magnetization parallel to the recording plane. By making the bits stand vertical as opposed to on their side, recording density is effectively doubled, Figure 1. The new perpendicular mode of recording not only produces sharp magnetization transitions necessary at higher recording densities, but is also more stable.

The 4 MB disks utilize barium ferrite coated substrates to achieve perpendicular mode of magnetization. Current disks use cobalt iron oxide (Co-g-Fe_2O_3) coating for longitudinal recording. The barium ferrite ensures good head to medium contact, stable output and durability in terms of long use. High coercivity is required to attain high recording density for a longitudinal recording medium (coercivity specification of a disk refers to the magnetic field strength required to make an accurate record on the disk). A conventional head could not be used in this case; however, the barium ferrite disk has low coercivity and the conventional ferrite head can be used. The new combination heads include a pre-erase mechanism, i.e., the ferrite ring heads containing erase elements followed by the read/write head. These erase elements have deep overwrite penetration and ensure complete erasure for writing new data. The distance between the erase elements and the read/write head is about 200mm. This distance is important from the floppy disk controller point of view and will be discussed in later sections.

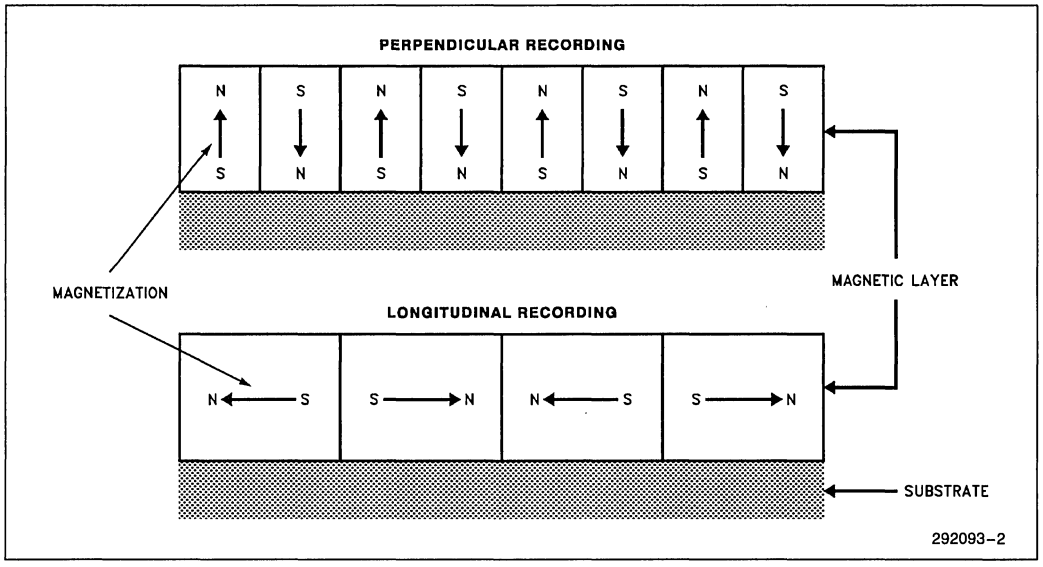


Figure 1. Perpendicular vs Longitudinal Recording

PERPENDICULAR DRIVE FORMAT AND SPECIFICATION

Figures 2a and 2b show the IBM drive format for both double density and perpendicular modes of recording. The main difference in recording format is the length of Gap2 between the ID field and the Data field. The main reason for the increased Gap2 length is the pre-erase head preceding the read/write head on the newer 4 MB floppy drives. The size of the data field is maintained at 512 KBytes standard. The increase in the capacity is implemented by increasing the number of sectors from 18 to 36. Table 1 shows the specifications of the various capacity 3.5" drives.

PERPENDICULAR MODE COMMAND

The current 82077AA/SL parts contain the "enhanced" perpendicular mode command as shown in Figure 3. This is a two byte command with the first byte being the command code (0x12H). The 2nd byte contains the parameters required to enable perpendicular mode recording. The former command (in the older 82077 parts) included only the WGATE and GAP bits. This command is compatible to the older mode where only the two LSBs are written. The enhanced mode allows system designers to designate specific drives as perpendicular recording drives. The second byte will be referenced as the PR[0:7] byte for ease of discussion. The following discusses the use of the enhanced perpendicular recording mode.

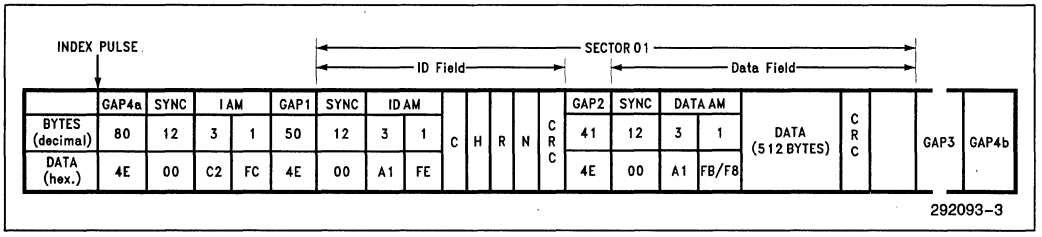


Figure 2a. Conventional IBM 1 MB and 2 MB Format (MFM)

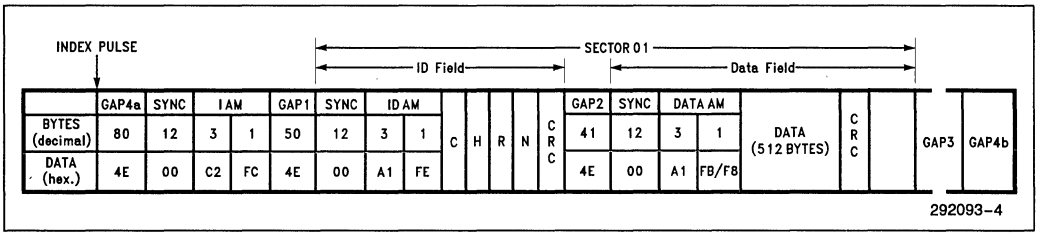


Figure 2b. Perpendicular 4 MB Format (MFM)

Phase	R/W	Data Bus								Remarks
		D7	D6	D5	D4	D3	D2	D1	D0	
PERPENDICULAR MODE COMMAND										
Command	W	0	0	0	1	0	0	1	0	Command Code
	W	OW	0	D3	D2	D1	D0	GAP	WGATE	PR

Figure 3. Perpendicular Mode Command

Table 1. Specifications of FDDs

Various Parameters Used in the Different Kinds of FDDs.		5.25" 360 KB	5.25" 1.2 MB	3.5" 720 KB	3.5" 1.44 MB	3.5" 2.88 MB
Number of Cylinders		40	80	80	80	80
Sectors/Track		9	15	9	18	36
Formatted Capacity		354 KB	1.2 MB	720 KB	1.44 MB	2.88MB
Unformatted Capacity		360 KB	1.6 MB	1 MB	2 MB	4 MB
Rotation Speed (rpm)	XT AT	300 360	360	300	300	300
Track Density (tpi)		48	96	135	135	135
Recording Density (bpi)		5876	9870	8717	17432	34868
Data Transfer Rate (Mbps)	XT AT	0.25 0.30	0.5	0.25	0.5	1
Gap Length for Read/Write		42	42	27	27	56
Gap Length for Format		80	80	84	84	83
Sector Size		512 KB	512 KB	512 KB	512 KB	512 KB
Density Notation		DD/DS	HD/DS	DD/DS	HD/DS	ED/DS

The following describes the various functions of the programmed bits in the PR:

OW — If this bit is not set high, all PR[2:5] are ignored. In other words, if OW = 0, only GAP and WGATE are considered. In order to select a drive as perpendicular, it is necessary to set OW = 1 and select the Dn bit.

Dn — This refers to the drive specification bits and corresponds to PR[2:5]. These bits are considered only if OW = 1. During the READ/WRITE/FORMAT command, the drive selected in these commands is compared to Dn. If the bits match then perpendicular mode will be enabled for that drive. For example, if D0 is set then drive 0 will be configured for perpendicular mode.

GAP — This alters the Gap2 length as required by the perpendicular mode format.

WGATE — Write gate alters timing of WE to allow for pre-erase loads in perpendicular drives.

The VCOEN timing and the length of the Gap2 field (explained above) can be altered to accommodate the unique requirements of the 4 MB floppy drives by GAP and WGATE bits of the PR. Table 2 describes the effects of the GAP and WGATE bits for the perpendicular command.

82077AA/SL's PERPENDICULAR MODE SUPPORT

The 82077AA and 82077SL both support 4 MB recording mode. The 82077SL has power management features included as well. Both AA and SL product lines have three versions each out of which two of the versions support the 4 MB floppy drives. The 82077AA-1, 82077AA, 82077SL, and 82077SL-1 all support the 4 MB floppy drives. A single command puts the 82077AA/SL into the perpendicular mode. This mode also requires the data rate to be set at 1 Mbps. The FIFO that is unique to Intel's 82077AA/SL parts may become necessary to remove the host interface bottleneck due to the higher data rate. The 4 MB floppy disk drives are downward compatible to 1 MB and 2 MB floppy diskettes. The following discussion explains the implications of the new 4 MB combination head and the functionality of the perpendicular mode command.

Table 2. Effects of GAP and WGATE Bits

GAP	WGATE	Mode	VCO Low Time after Index Pulse	Length of Gap2 Format Field	Portion of Gap2 Written by Write Data Operation	Gap2 VCO Low Time for Read Operations
0	0	Conventional	33 Bytes	22 Bytes	0 Bytes	24 Bytes
0	1	Perpendicular (Data Rate = 500 kbps)	33 Bytes	22 Bytes	19 Bytes	24 Bytes
1	0	Conventional	33 Bytes	22 Bytes	0 Bytes	24 Bytes
1	1	Perpendicular	33 Bytes	41 Bytes	38 Bytes	43 Bytes

The implementation of 4 MB drives requires understanding the Gap2 (see Figures 2a and 2b) and VCO timing requirements unique to these drives. These new requirements are dictated by the design of the "combination head" in these drives. Rewriting of disks in the 4 MB drives requires a pre-erase gap to erase the magnetic flux on the disk preceding the writing by the read/write gap. The read/write gap in the 4 MB drive does not have sufficient penetration (as shown in Figure 4a) to overwrite the existing data. In the conventional drives, the read/write gap had sufficient depth and could effectively overwrite the older data as depicted in Figure 4b. It must be noted that it is necessary to write the conventional 2 MB media in the 4 MB drive at 500 Kbps perpendicular mode. This ensures proper erasure of existing data and reliable write of the new data. The pre-erase gap in the 4 MB floppy drives is activated only during format and write commands. Both the pre-erase gap and read/write gap are activated at the same time.

As shown in Figure 4a, the pre-erase gap precedes the read/write gap by 200mm. This distance translated to bytes is about 38 bytes at a data rate of 1 Mbps and 19 bytes at 500 Kbps. Whenever the read/write gap is enabled by the Write Gate signal the pre-erase gap is activated at the same time.

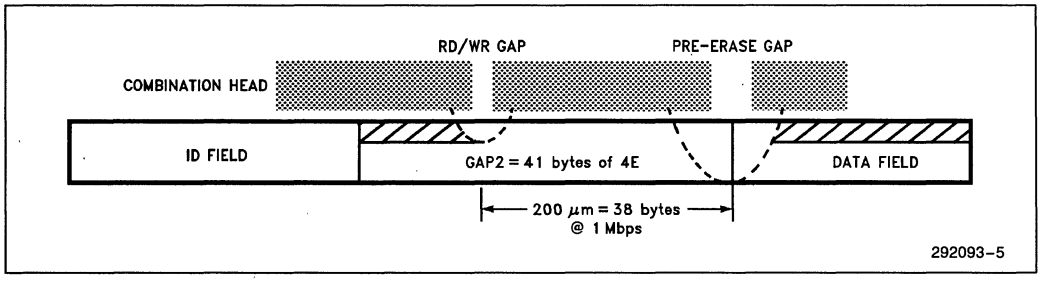


Figure 4a. Head Design for the 4 MB Perpendicular Mode

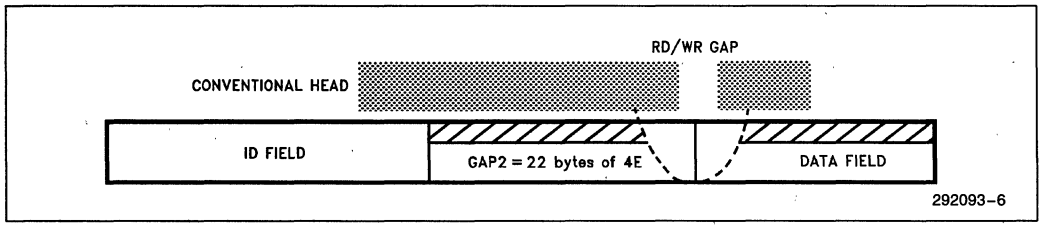


Figure 4b. Head Design for the Conventional 2 MB Mode

In conventional drives, the Write Gate is asserted at the beginning of the sync field, i.e., when the read/write is at the beginning of the data field. The controller then writes the new sync field, data address mark, data field and CRC (see Figure 2a). With the combination head, the read/write gap must be activated in the Gap2 field to ensure proper write of the new sync field. To accommodate both the distance between the pre-erase gap and read/write gap and the head activation and deactivation time, the Gap2 field is expanded to a length of 41 bytes at 1 Mbps (see Figure 2b). Since the bit density is proportional to the data rate, 19 bytes will be written in the Gap2 field at 500 Kbps data rate in the perpendicular mode.

On the read back by the 82077AA/SL, the controller must begin the synchronization at the beginning of the sync field. For conventional mode, the internal PLL VCO is enabled (VCOEN) approximately 24 bytes from the start of the Gap2 field. However, at 1 Mbps perpendicular mode the VCOEN goes active after 43 bytes to accomodate the increased Gap2 field size. For each case, a 2 byte cushion is maintained from the beginning of the sync field to avoid write splices caused by motor speed variation.

It should be noted that none of the alterations in Gap2 size, VCO timing or Write Gate timing affect the normal program flow. Once the perpendicular command is invoked, 82077AA/SL behaviour from the user standpoint is unchanged.

PROGRAMMING PERPENDICULAR MODE

Figures 5a and 5b show a flowchart on how the perpendicular recording mode is implemented on the 82077AA/SL. The perpendicular mode command can be issued during initialization. As shown in Figure 5a the perpendicular command stores the PR value internally. This value is used during the data transfer commands for configuration in order to deal with the perpendicular drives. Table 2 shows how the Gap2 length, VCOEN timing or Write Gate timing is affected. The OW bit is also tested for in this part of the loop. The enhanced perpendicular mode is enabled by setting the OW = 1, setting the Dn bits corresponding to the installed perpendicular drive high and leaving PR[0:1] = '00'.

As shown in Figure 5b, the Gap2 length is initially set to the conventional length of 22 bytes. Next the PR[0:1] bits (GAP, WGATE) are checked if they are set to '00'. If the PR[0:1] bits are set to '10' then, perpendicular mode is disabled and conventional mode is retained. If the PR[0:1] = '01' or '11' the VCOEN is

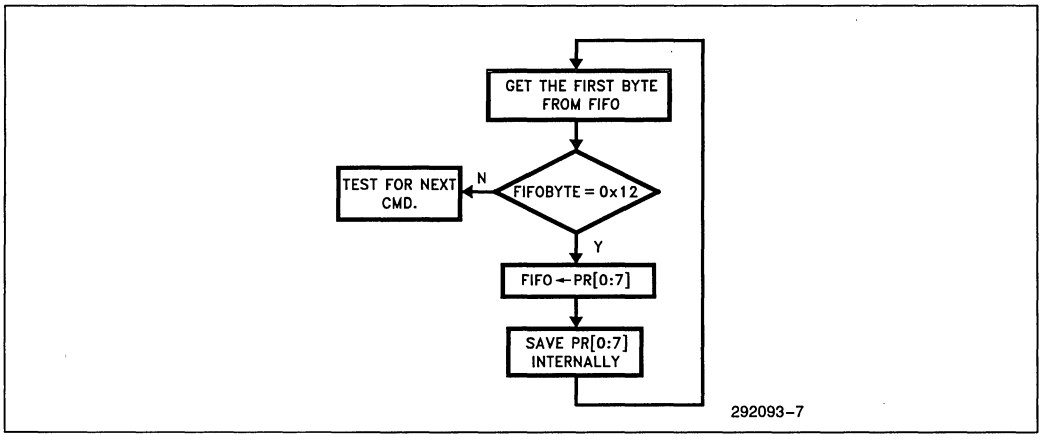


Figure 5a. Perpendicular Command Handling

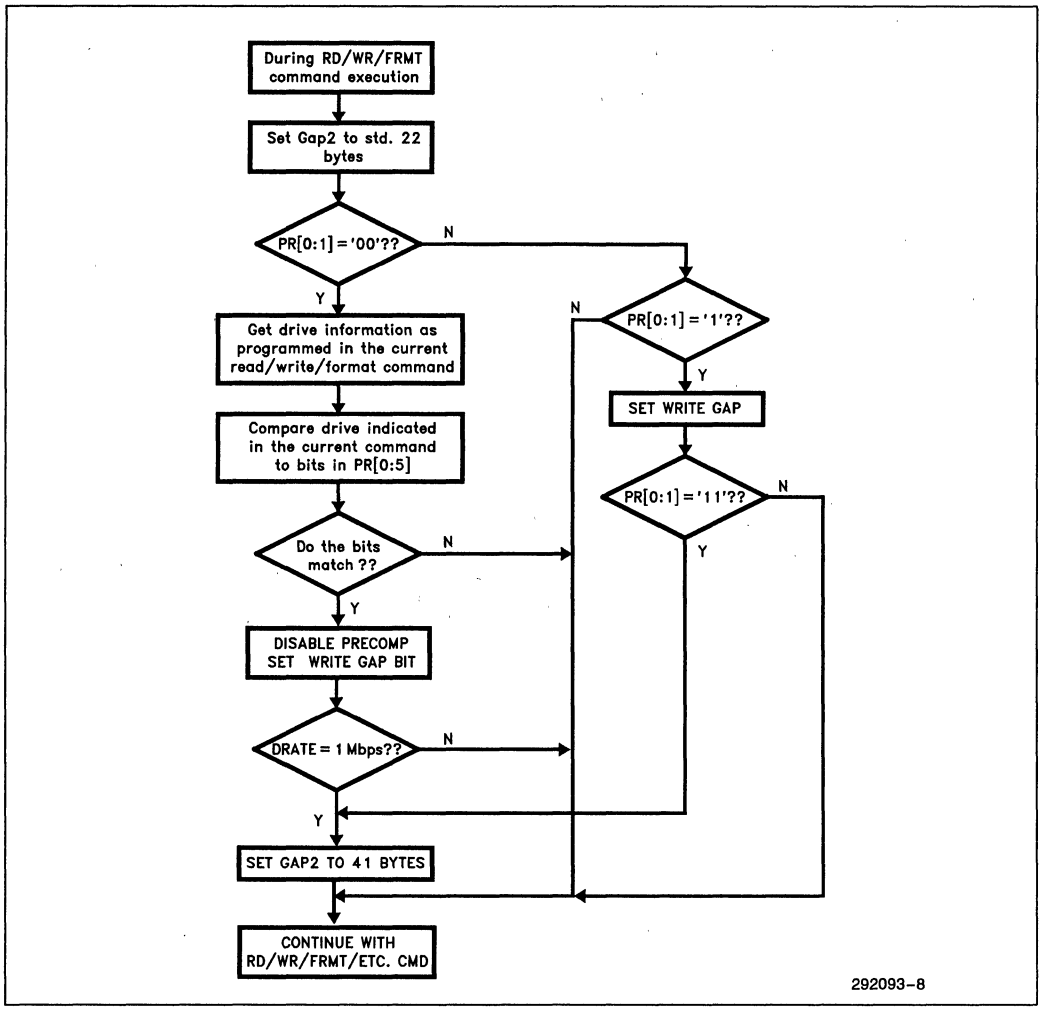


Figure 5b. During Data Transfer Commands

set to activate 43 bytes or 24 bytes from the start of the Gap2 field, depending on the value as shown in Table 2. After this, PR[0:1] = '11' is checked; if not true (programmed '01') the program is exited with only the VCOEN timing being set for perpendicular mode. If true, however, the Gap2 length is set up for perpendicular mode (note: this is done independent of the data rate). It must be noted that if the PR[0:1] bits are set to '11' then it is up to the user to disable precompensation before accessing perpendicular drives. The other branch of the flowchart refers to setting of PR[0:1] to '00'. In this case, the perpendicular command will have the following effect:

1. If any of the Dn bits in PR[2:5] programmed high, then precompensation is automatically disabled (0 ns is selected for the specified drive regardless of the data rate) and VCOEN is set to activate appropriately. All the bits that are set low will enable the 82077 to be configured for conventional mode, i.e., exit the program without modifications (shown Figure 5b).
2. Next the data rate is checked for 1 Mbps. If the data rate is at 1 Mbps, then Gap2 length is set to 41 bytes, otherwise, the program is exited without setting up the Gap2 to 41 bytes.

It must be noted that if PR[2:5] are to be recognized in the command the OW bit must be set high. If this bit is low, setting of Dn bits will have no effect. Setting the OW bit will enable the storage of the Dn bit. Also setting PR[0:1] to any other value than '00' will override anything written in the Dn bits. In other words, setting PR[0:1] to a value other than '00' enables the effect of that for all drives. It must be noted that if PR[0:1] bits are set to a value other than '00' then it is recommended not to use the enhanced command mode, i.e., all other bits should be zero. Consider the following examples:

a. PR[0:7] = 0x84; This is the way to use the command in the enhanced mode. In this case, the OW = 1 and D0 is set high. During the data transfer command, if D0 is selected it will be automatically configured for perpendicular mode. If D1 is accessed, however, it will be configured for conventional mode. Similarly, if PR[0:7] = 0x88 then D1 is configured for perpendicular mode and D0 is configured for conventional mode. Software resets do not clear this mode.
b. PR[0:7] = 0x03; This is the way to use the command in the old mode. If the user decides to use this mode, then it must be noted that the command has to be issued before every data transfer command. Also when used this way, all the drives are configured for perpendicular mode. The user must also remember to disable precompensation and set the data rate to 1 Mbps while accessing the perpendicular drive in the system. Any software reset clears the command.
c. PR[0:7] = 0x87; In this case, the OW = 1, D0 = 1 and PR[0:1] = 11. This may be called a mixed mode and should be refrained from usage. This is similar to setting PR[0:7] = 0x03, because setting PR[0:1] high overrides automatic configuration. In this case the user has to be aware that precompensation must be disabled and the data rate must be set to 1 Mbps while accessing drive 0. After software reset, bits GAP and WGATE will be cleared, but OW and D0 will retain their previously set values. In other words, after software reset, the part will see PR[0:7] = 0x84. Evidently, this would cause problems and, therefore, it is recommended this mode *not* be used.
d. PR[0:7] = 0x80; In this case, the OW = 1, Dn = 0 and PR[0:1] = 00. This has the effect of clearing the perpendicular mode command without doing a hardware reset. Another way to do this would be to set PR[0:7] = 0x02; this can then be used to temporarily disable perpendicular mode configuration without affecting the previously programmed Dn values. Software reset following this will reenable the previously programmed enhanced mode command.

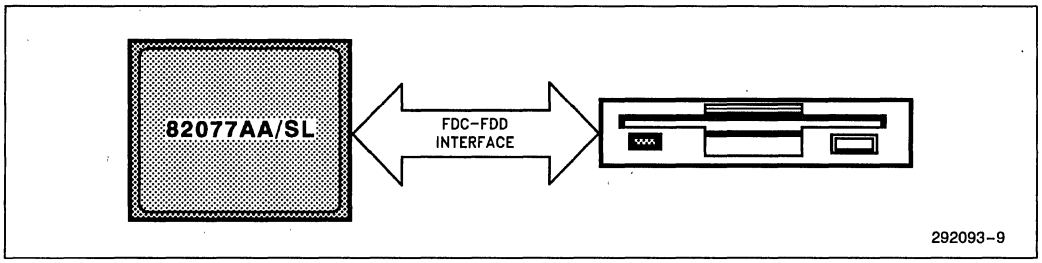
Using the enhanced perpendicular command removes the requirement of issuing the perpendicular command for each data transfer command and manually setting the perpendicular configuration.

"Software" RESETs (via DOR or DSR registers) will only clear the PR[0:1] values to '0'. Dn bits will retain their previously programmed values. "Hardware" RESETs will clear all the programmed bits including OW and Dn bits to '0'. The status of these bits can be determined by issuing the dumpreg command and checking the 8th result byte. This byte will contain the programmed values of the Dn and PR[0:1] bits as shown in Figure 6. The OW bit is *not* returned in this result byte.

Phase	R/W	Data Bus								Remarks
		D7	D6	D5	D4	D3	D2	D1	D0	
DUMPREG COMMAND										
Command	R	Eighth Result Byte								
		LOCK	0	D3	D2	D1	D0	GAP	WGATE	

Figure 6. Dumpreg Command

INTERFACE BETWEEN 82077AA/SL AND THE DRIVE



There is currently no industry-wide standard for the FDC to FDD interface. There are numerous floppy drive vendors, each with their own modes and interface pins to enable 4 MB perpendicular mode. The drive interface not only varies from manufacturer to manufacturer but also within a manufacturer's product line. The differences on the interface mainly originate from configuring the floppy drive into the 4 MB mode. Depending on the drive, the differences can create problems of daisy-chaining a 4 MB drive with the standard 1 MB and 2 MB drives. Of course, for laptops this is not a problem since most of them use a single floppy drive. Lack of an industry standard makes it necessary to look at each drive and build a interface for that particular drive.

The following is a brief discussion about some of the floppy drives available in the market and how these can be interfaced with the 82077AA/SL. It is important to note that although a manufacturer's name may be given in connection with the interface described, Intel does not guarantee that the interface discussed will apply to all the drives from that manufacturer. The main goal is introduce to the reader how to interface the 82077AA/SL with a 4 MB floppy drive.

Previously, for the conventional 1 MB and 2 MB AT mode drives, a single Density Select input was used by floppy drives to select between high density and low density drives. A high on this input enabled high density operation (500 Kbps) whereas a low enabled low density operation (300 Kbps/250 Kbps). This signal was asserted high or low by the floppy disk controller depending on the data rate programmed. For the 4 MB operation, there are two inputs defined by the floppy drive manufacturers. The polarity of these inputs enables the selected density operation. Implementing this requires at least 1 new pin to be defined on the FDC-FDD interface. Most floppy vendors have elected to take pin 2 (originally density select) and redefine the polarity to conform to one of these new density select inputs and another pin to be the other density select input. However, the new density select on pin 2 is not compatible to the old density select input in many of the floppy drives. This precludes the user from daisy chaining 4 MB drives with conventional drives. Another problem is that the second density select pin varies on its location on the FDC-FDD interface from drive to drive.

The way that the BIOS determines what type of diskette is in what type of drive is by trial and error. The system tries to read the diskette at 250 Kbps; if it fails then it will set the data rate to higher value and retry. The BIOS does this until the right data rate is selected. This method will still be implemented for the 4 MB drives by some BIOS vendors. However, the 4 MB drives available today also have two media sense ID pins that relate to the user what type of media is present in the floppy drive. This information will also require two pins on the FDC-FDD interface. The location of these pins is once again variable from drive to drive.

Some manufacturers have circumvented the entire standardization problem by including an auto configuration in the drive. In these cases, the type of floppy put into the drive is sensed by the hole (each 4/2/1 MB diskette has a hole in different locations identifying it) on the diskette. Then the drive automatically sets itself up for this mode. The BIOS must obviously set up the floppy disk controller for the correct data rate which could be done if the media sense ID was read and decoded as to the data rate. Due to lack of extra pins on the even side of the floppy connector the newer locations of some of the functions are migrating to the odd pins (previously all grounded). Some drive manufacturers have even made this configurable via jumpers. For instance, the new TEAC drives have a huge potpourri of configurations that would satisfy the appetite of some of the most finicky system interfaces.

The 82077AA/SL currently has two output pins DRATE0 and DRATE1 (pins 28 and 29 respectively) which directly reflect the data rate programmed in the DSR and CCR registers. These two pins can be used to select the correct density on the drive. These two can also be used with the combination of DENSEL to select the correct data rate. At the present time the 82077AA/SL does not support media sense ID. However, the user could easily make it readable directly by BIOS. The following is a discussion on what combination of DRATE0, DRATE1, and DENSEL could be used to interface to some of the currently available floppy drives.

2

1. TEAC 235J-600/Toshiba PD-211/Sony (Old Version)

These were among the first 4 MB drives available in the market. Each of them has a mode select input on pins 2 and 6. The polarity required for each different data rate is as shown below:

Data Rate	Capacity	DRATE1	DRATE0	MODSEL0 pin 2	MODSEL1 pin 6
1 Mbps	4 MB	1	1	1	0
500 Kbps	2 MB	0	0	0	1
300 Kbps/ 1 Mbps	4 MB	0	1	1	1
250 Kbps	1 MB	1	0	0	0

It is clear from the above that DRATE0 = MODSEL0 and MODSEL1 = DRATE1#. This would mean taking the drate signals onto pins 2 and 6 of the FDC-FDD interface. Unfortunately this solution requires an inverting gate. TEAC has recently, however, come out with a new version called TEAC 235J-3653. On this drive there are a number of possible configurations into which the drive can be put into, however, only the best way to interface to the 82077AA/SL will be discussed. The requirements are as shown below. This shows that HDIN = DENSEL (original signal for conventional drives) and EDIN = DRATE0. As suggested in the TEAC spec for method 1, the straps connected are MSC, HI2 (sets HDIN on pin 2), DC34 and EI6 (sets EDIN on pin 6). Pins 4, 29, and 33 are left open. Since pin 2 has the same polarity as the conventional drive requirement and the secondary input is connected via pin 6 (no connect on the conventional drives) daisy chaining this TEAC drive with a conventional drive does not cause any incompatibility. Figure 7 shows how the TEAC can be connected to the 82077AA/SL. It also shows daisy chaining of the TEAC drive with a conventional drive.

Data Rate	Capacity	DENSEL	DRATE1	DRATE0	HDIN pin 2	EDIN pin 6
1 Mbps	4 MB	1	1	1	X	1
500 Kbps	2 MB	1	0	0	1	0
300 Kbps/ 1 Mbps	4 MB	0	0	1	X	1
250 Kbps	1 MB	0	1	0	0	0

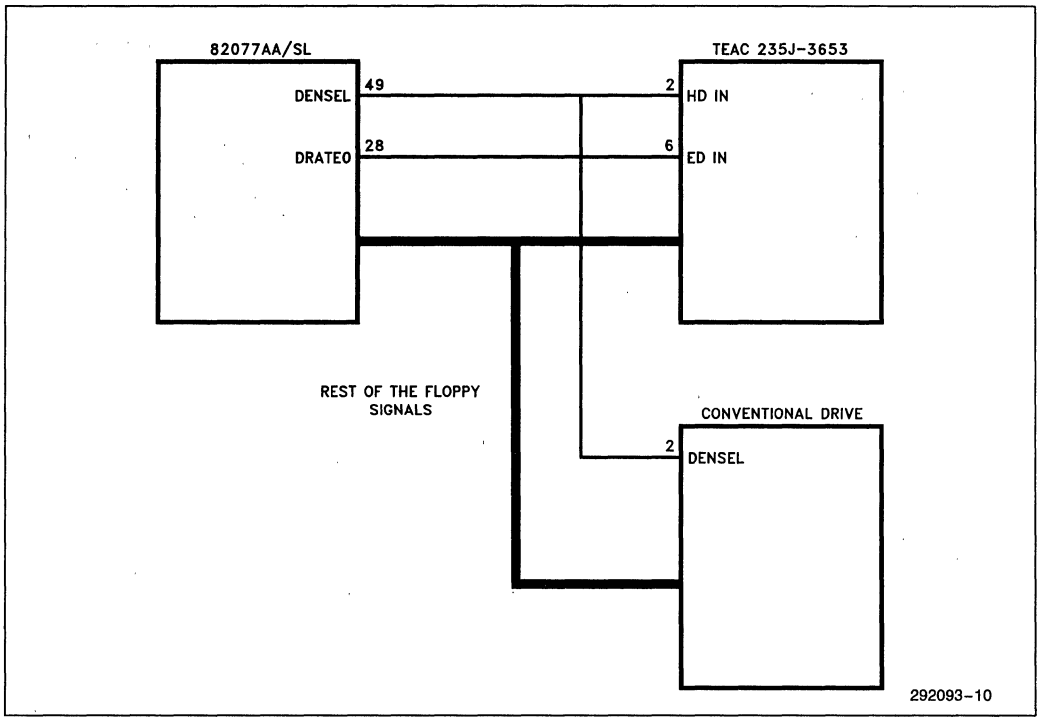


Figure 7. Interfacing 82077AA/SL to TEAC 235J-3653

2. Panasonic JU-259A (New Version)

This is Panasonic's new drive and has the HDIN signal on pin 2 and EDIN signal on pin 6. The requirements are shown below. This type of interface allows for daisy chaining the Panasonic drive with a conventional drive. The DENSEL signal can be connected to pin 2 and the DRATE0 should be connected to pin 6.

Data Rate	Capacity	DENSEL	DRATE1	DRATE0	HDIN pin 2	EDIN pin 6
1 Mbps	4 MB	1	1	1	1	1
500 Kbps	2 MB	1	0	0	1	0
300 Kbps/ 1 Mbps	4 MB	0	0	1	0	1
250 Kbps	1 MB	0	1	0	0	0

3. Mitsubishi MF356C (Model 252UG/788UG)

There are two models of this drive. The 252UG has DENSEL1 on pin 2 and DENSEL0 on pin 33, whereas the 788UG has DENSEL0 located on pin 2 and DENSEL1 located on pin 6. Via jumpers, it is possible to configure the drives to different polarity for the density select line. The following table shows the configuration for the 252UG in which jumper setting is 2MS = I/F and 4 MS = I/F.

Data Rate	Capacity	DENSEL	DRATE1	DRATE0	DENSEL1 pin 2	DENSEL0 pin 33
1 Mbps	4 MB	1	1	1	1	1
500 Kbps	2 MB	1	0	0	1	0
300 Kbps/ 1 Mbps	4 MB	0	0	1	0	1
250 Kbps	1 MB	0	1	0	0	0

The correct connection requirement is: DENSEL (from 82077AA/SL) = DENSEL1 and DRATE0 = DENSEL0. Although there are other configurations, this provides the best one, since daisy chaining is possible without any problem.

4. Epson SMD-1060

This drive has 3 different modes of operation. Mode B is the best and is similar to Mitsubishi's drives as described above. In this mode, HDI signal is connected to pin 2 and EDI is connected to pin 33. Mode B is enabled by inserting jumpers across 3-4 and 7-8 (SS01 B block) and 1-2 and 3-4 (SS03 block) for the drive with the power separated type (i.e., a connector for the floppy signals and another one for power supply) of 34-pin connector.

Data Rate	Capacity	DENSEL	DRATE1	DRATE0	HDI pin 2	EDI pin 33
1 Mbps	4 MB	1	1	1	1	1
500 Kbps	2 MB	1	0	0	1	0
300 Kbps/ 1 Mbps	4 MB	0	0	1	0	1
250 Kbps	1 MB	0	1	0	0	0

As demonstrated by the table, HDI = DENSEL and EDI = DRATE0. These connections would ensure daisy chaining capability without any problems.

5. Sony MP-F40W-14/15

The dash 14 and 15 are two drives from Sony that handle 4 MB requirements. The MP-F40W-14 has the DENSITY SELECT 1, DENSITY SELECT 0 on pins 2 and 33 respectively, whereas the MP-F40W-15 has the DENSITY SELECT 1, DENSITY SELECT 0 on pins 2 and 6 respectively. As it is obvious from the table below, daisy chaining is easily done if the 82077AA/SL is connected in the PS/2 mode (by tying IDENT low) with either type of drive, the only difference being the location of DENSITY SELECT 0.

Data Rate	Capacity	DENSEL PS/2 mode (IDENT = 0)	DRATE1	DRATE0	DENSITY SELECT1 pin 2	DENSITY SELECT0 pin 6/33
1 Mbps	4 MB	0	1	1	0	1
500 Kbps	2 MB	0	0	0	0	0
300 Kbps/ 1 Mbps	4 MB	1	0	1	1	1
250 Kbps	1 MB	1	1	0	1	0

If the drive is used in the PS/2 mode, then DENSITY SELECT1 = DENSEL and DENSITY SELECT0 = DRATE0. To use the drive in AT mode, DENSITY SELECT1 = DRATE1 and DENSITY SELECT0 = DRATE0, as shown below. However, daisy chaining is not possible.

Data Rate	Capacity	DENSEL PS/2 mode (IDENT = 0)	DRATE1	DRATE0	DENSITY SELECT1 pin 2	DENSITY SELECT0 pin 6/33
1 Mbps	4 MB	0	1	1	1	1
500 Kbps	2 MB	0	0	0	0	0
300 Kbps/ 1 Mbps	4 MB	1	0	1	0	1
250 Kbps	1 MB	1	1	0	1	0

6. Toshiba ND3571

Toshiba MB drive has the HD mode selection on pin 6 and ED mode selection on pin 2. This causes daisy chaining problems with conventional drives as shown in the figure below:

Data Rate	Capacity	DENSEL	DRATE1	DRATE0	ED Mode pin 2	HD Mode pin 6
1 Mbps	4 MB	1	1	1	1	1
500 Kbps	2 MB	1	0	0	0	1
300 Kbps/ 1 Mbps	4 MB	0	0	1	1	0
250 Kbps	1 MB	0	1	0	0	0

The DENSEL from the 82077 is connected to pin 6 and DRATE0 is connected to pin 2.

82077SL 4 MB DESIGN

This section presents a design application of a PC/AT compatible floppy disk controller. The 82077SL integrates the entire PC/AT controller design with the exception of the address decode on a single chip. The schematic for this solution is shown in Figure 8. The chip select for the 82077SL is generated by a 85C220 μPLD that is programmed to decode addresses 03F0H through 03F7H when AEN is low. The programming equations for the μPLD is in the Intel's .ADF format and can be processed using the IPLSII compiler (available from Intel).

A floppy disk interface is provided by on-chip output buffers with a 40 mA sink capability. The outputs from the disk drive are terminated at the floppy disk controller with a 1 KΩ resistor pack. The 82077SL disk interface inputs contain a Schmitt trigger input structure for higher noise immunity. The host interface is a similar direct connection with on-chip 12 mA sink capable buffers on DB0–7, INT and DRQ.

The schematic shows eleven jumpers numbered J1 through J11. The table below describes the functions of these jumpers as well as their normal connections. The normal connections allow the BIOS to work without modification. In the normal mode, the 82077SL responds to DRQ2 and DACK2# as well as IRQ6. Depending on the type of drive interfaced to this board, the DENOUT0 and DENOUT1 signals can be tied. With the setting to 2–3 on J8 and J9, the default setting is DENSEL on DRVDEN0 and DRATE0 on DRVDEN1. PIN6/33 SELECT is used to set for pin 6 as the EDIN input. The J11 should always be closed. It can be used to measure the current consumption of 82077SL. J7 selects between the primary and secondary address spaces. There are two resistor packs used for pullups on input signals from the floppy drive interface. These resistors are rated at 1K. Please note that if using older 5.25″ drives, the pullup on some of them is 150Ω. Most modem 5.25″ drives use a 1K value. In order to ensure the correct value please refer to the floppy drive specification manual.

For further information, please contact your local Intel sales office.

Jumper	Description	Normal Connection
J1	DRQ1: DMA request 1 used with DACK1# to allow for DMA transfers	Open
J2	DRQ2: DMA request 2 used with DACK2# to allow for DMA transfers	Closed
J3	DACK1: DMA acknowledge 1 used with DRQ1 to allow for DMA transfers	Open
J4	DACK2: DMA acknowledge 2 used with DRQ2 to allow for DMA transfers	Closed
J5	IRQ5: Interrupt line 5 used to generate floppy interrupts	Open
J6	IRQ6: Interrupt line 6 used to generate floppy interrupts	Closed
J7	DRV2: Address selection (between 3FX and 37X address ranges)	Open
J8	DENOUT0: Used with DENOUT1 to select the values of DRVDEN1,0	2–3
J9	DENOUT1: Used with DENOUT0 to select the values of DRVDEN1,0	2–3
J10	PIN6/33 SELECT: Used to select between pin 6 and pin 33 for EDIN input	1–2 or 2–3
J11	V_{BB}/V_{CC}: Connection between two power layers	Closed

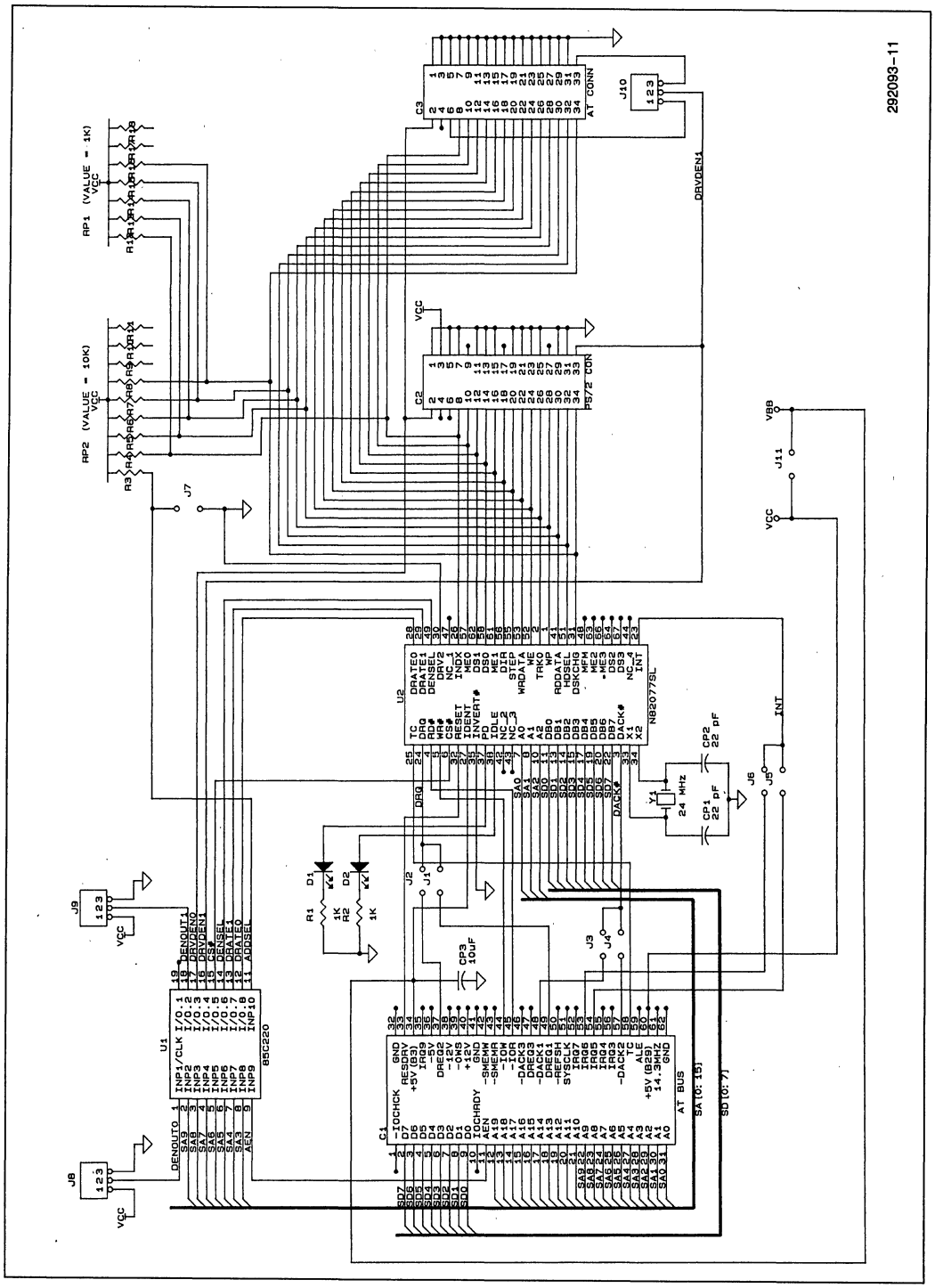


Figure 8. 82077SL Evaluation Board

```
Designer: K. Shah
Company: Intel Corp.
Dept:    IMD Marketing
Date:    April '92
Rev.#:
% The μPLD used in the 82077SL Evaluation board design, Rev.#1.0. %
85C220 dip package

OPTIONS: TURBO = ON

PART:    85C220

INPUTS:
        SA9@2, % System Address Inputs %
        SA8@3,
        SA7@4,
        SA6@5,
        SA5@6,
        SA4@7,
        SA3@8,
        AEN@9,

        DENOUT0@1, % Maps the DRVDEN0 and DRVDEN1 to appropriate polarity table %
        DENOUT1@18, % Maps the DRVDEN0 and DRVDEN1 to appropriate polarity table %

        ADDSEL@11, % Selects between primary and secondary address spaces %

        DRATE0@12, % DRATE0 signal from the 82077SL %
        DRATE1@13, % DRATE1 signal from the 82077SL %
        DENSEL@14 % DENSEL signal from the 82077SL %

OUTPUTS:
        CS_@15, % 82077SL chip select signal %

        DRVDEN1@16, % Drive density signal connected to EDIN of the drive %
        DRVDEN0@17 % Drive density signal connected to HDIN of the drive %

NETWORK:
        % Inputs %

        SA9 = INP(SA9)
        SA8 = INP(SA8)
        SA7 = INP(SA7)
        SA6 = INP(SA6)
        SA5 = INP(SA5)
        SA4 = INP(SA4)
        SA3 = INP(SA3)
        AEN = INP(AEN)
        ADDSEL = INP(ADDSEL)
        DRATE0 = INP(DRATE0)
        DRATE1 = INP(DRATE1)
        DENSEL = INP(DENSEL)
        DENOUT0 = INP(DENOUT0)
        DENOUT1 = INP(DENOUT1)

        % Outputs %

        CS_ = CONF(CSeq, VCC)

        DRVDEN0 = CONF(DEN0eq, VCC)
        DRVDEN1 = CONF(DEN1eq, VCC)
```

```
EQUATIONS:

       % CS_is activated for 3F0-3F7 and 370-377 address spaces %
       CSeq = (AEN' * SA9 * SA8 * SA7' * SA6 * SA5 * SA4 * SA3' * ADDSEL'
              + AEN' * SA9 * SA8 * SA7 * SA6 * SA5 * SA4 * SA3' * ADDSEL)';

       % These are the signals generated on DRVDEN0 and DRVDEN1 for the FDC-FDD
         interface
       DENOUT1  DENOUT0  DRVDEN0  DRVDEN1
            0     0       DENSEL   DRATE0
            0     1       DENSEL'  DRATE0
            1     0       DRATE1   DRATE0
            1     1       DRATE0   DRATE1
       %

       DEN0eq = DENSEL * (DENOUT0' * DENOUT1') + DENSEL' * (DENOUT0 * DENOUT1')
              + DRATE1 * (DENOUT0' * DENOUT1) + DRATE0 * (DENOUT0 * DENOUT1);
       DEN1eq = DRATE1 * (DENOUT0 * DENOUT1) + DRATE0 * (DENOUT0' + DENOUT1');

END$
```

82077SL Application Note Revision Summary

The following changes have been made since revision 001:

Table 2 kBps was corrected to kbps.

Page 12 3. Mitsubishi MF356C description modified to read: "There are two models of this drive. The 252UG has DENSEL1 on pin 2 and DENSEL0 on pin 33, whereas the 788UG has DENSEL0 located on pin 2 and DENSEL1 located on pin 6. Via jumpers, it is possible to configure the drives to different polarity for the density select lines. The following table shows the configuration for the 252UG in which jumper setting is 2 MS = I/F and 4 MS = I/F."

Figure 8 Arrow added to diagram.

Page 17 Columns corrected to line up properly.

intel®

3

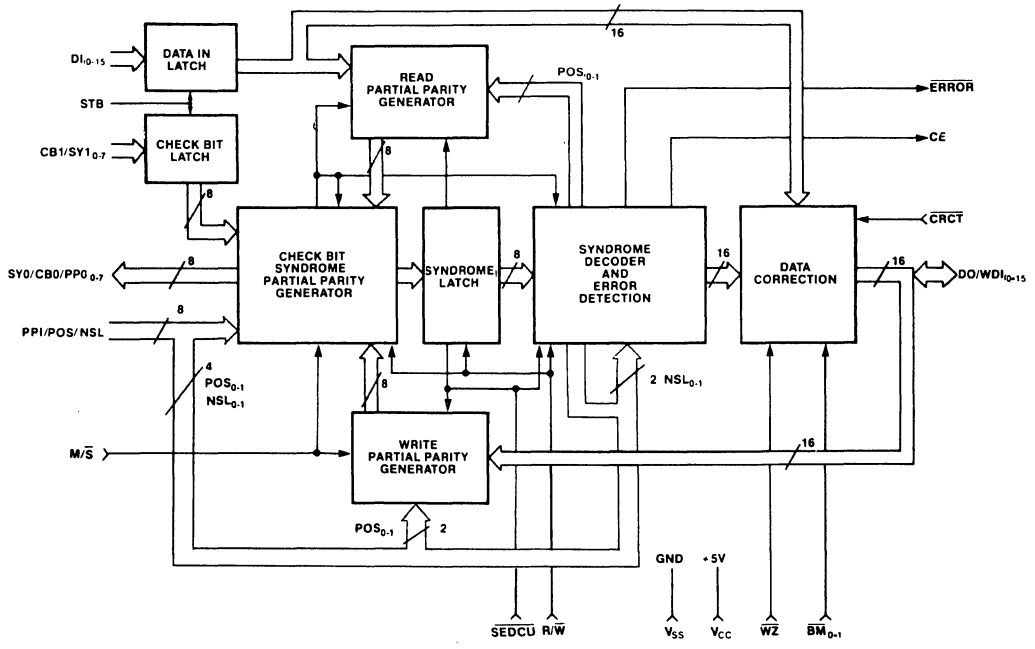
Memory Controllers

8206
ERROR DETECTION AND CORRECTION UNIT

- **Detects All Single Bit, and Double Bit and Most Multiple Bit Errors**
- **Corrects All Single Bit Errors**
3 Selections	8206-1	8206
Detection	35 ns	42 ns
Correction	55 ns	67 ns
- **Syndrome Outputs for Error Logging**
- **Automatic Error Scrubbing with 8207**
- **Expandable to Handle 80 Bit Memories**
- **Separate Input and Output Busses—No Timing Strobes Required**
- **Supports Read With and Without Correction, Writes, Partial (Byte) Writes, and Read-Modify-Writes**
- **HMOS III Technology for Low Power**
- **68 Pin Leadless JEDEC Package**
- **68 Pin Grid Array Package**

The HMOS 8206 Error Detection and Correction Unit is a high-speed device that provides error detection and correction for memory systems (static and dynamic) requiring high reliability and performance. Each 8206 handles 8 or 16 data bits and up to 8 check bits. 8206's can be cascaded to provide correction and detection for up to 80 bits of data. Other 8206 features include the ability to handle byte writes, memory initialization, and error logging.

3



205220-1

Figure 1. 8206 Block Diagram

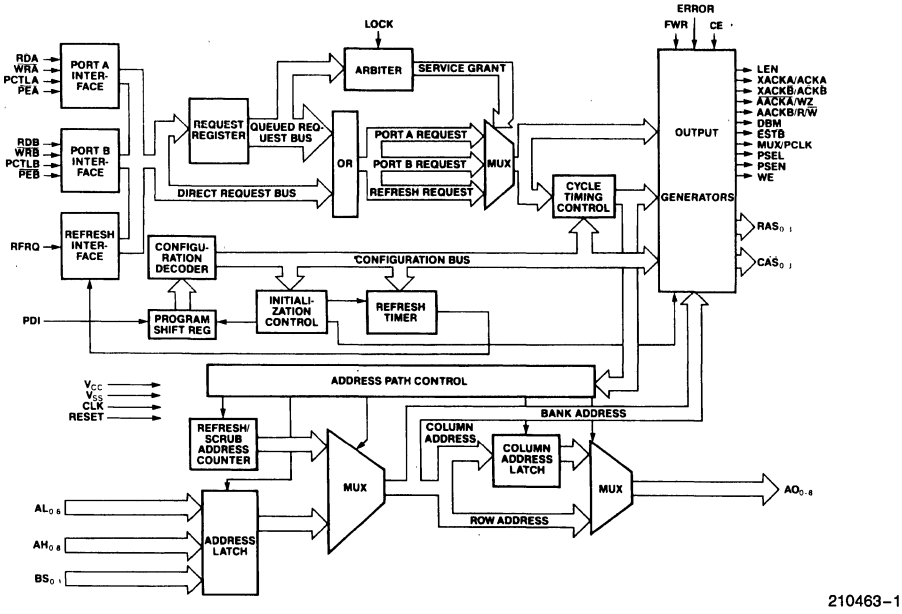
The complete document for this product is available from Intel's Literature Center at 1-800-548-4725.

September 1987
Order Number: 205220-008

8207
DUAL-PORT DYNAMIC RAM CONTROLLER

- **Provides All Signals Necessary to Control 16K, 64K and 256K Dynamic RAMs**
- **Directly Addresses and Drives up to 2 Megabytes without External Drivers**
- **Supports Single and Dual-Port Configurations**
- **Automatic RAM Initialization in All Modes**
- **Four Programmable Refresh Modes**
- **Transparent Memory Scrubbing in ECC Mode**
- **Fast Cycle Support for 8 MHz 80286 with 8207-16**
- **Slow Cycle Support for 8 MHz, 10 MHz 8086/88, 80186/188 with 8207-8, 8207-10**
- **Provides Signals to Directly Control the 8206 Error Detection and Correction Unit**
- **Supports Synchronous or Asynchronous Operation on Either Port**
- **68 Lead JEDEC Type A Leadless Chip Carrier (LCC) and Pin Grid Array (PGA), Both in Ceramic.**

The Intel 8207 Dual-Port Dynamic RAM Controller is a high-performance, systems-oriented, Dynamic RAM controller that is designed to easily interface 16K, 64K and 256K Dynamic RAMs to Intel and other microprocessor systems. A dual-port interface allows two different busses to independently access memory. When configured with an 8206 Error Detection and Correction Unit the 8207 supplies the necessary logic for designing large error-corrected memory arrays. This combination provides automatic memory initialization and transparent memory error scrubbing.



210463-1

Figure 1. 8207 Block Diagram

The complete document for this product is available from Intel's Literature Center at 1-800-548-4725.

September 1987
Order Number: 210463-007

82C08
CHMOS DYNAMIC RAM CONTROLLER

- **0 Wait State with INTEL μProcessors**
- **iAPX 286 (10, 8 MHz) } 82C08-20 20 MHz**
  82C08-16 16 MHz
  iAPX 186/88 } 82C08-10 10 MHz
  86/88 } 82C08-8 8 MHz
- **Supports 64K and 256K DRAMs (256K x 1 and 256K x 4 Organizations)**
- **Power Down Mode with Programmable Memory Refresh using Battery Backup**
- **Directly Addresses and Drives up to 1 Megabyte without External Drivers**
- **Microprocessor Data Transfer and Advance Acknowledge Signals**
- **Five Programmable Refresh Modes**
- **Automatic RAM Warm-up**
- **Pin-Compatible with 8208**
- **48 Lead Plastic DIP; 68 Lead PLCC**
  (See Intel Packaging; Order Number: 231369-001)
- **Compatible with Normal Modes of Static Column and Ripplemode DRAMs**

The Intel 82C08 Dynamic RAM Controller is a CMOS, high performance, systems oriented, Dynamic RAM controller that is designed to easily interface 64K and 256K Dynamic RAMs to Intel and other microprocessors. The 82C08 also has a power down mode where only the refresh logic is activated using battery backup.

3

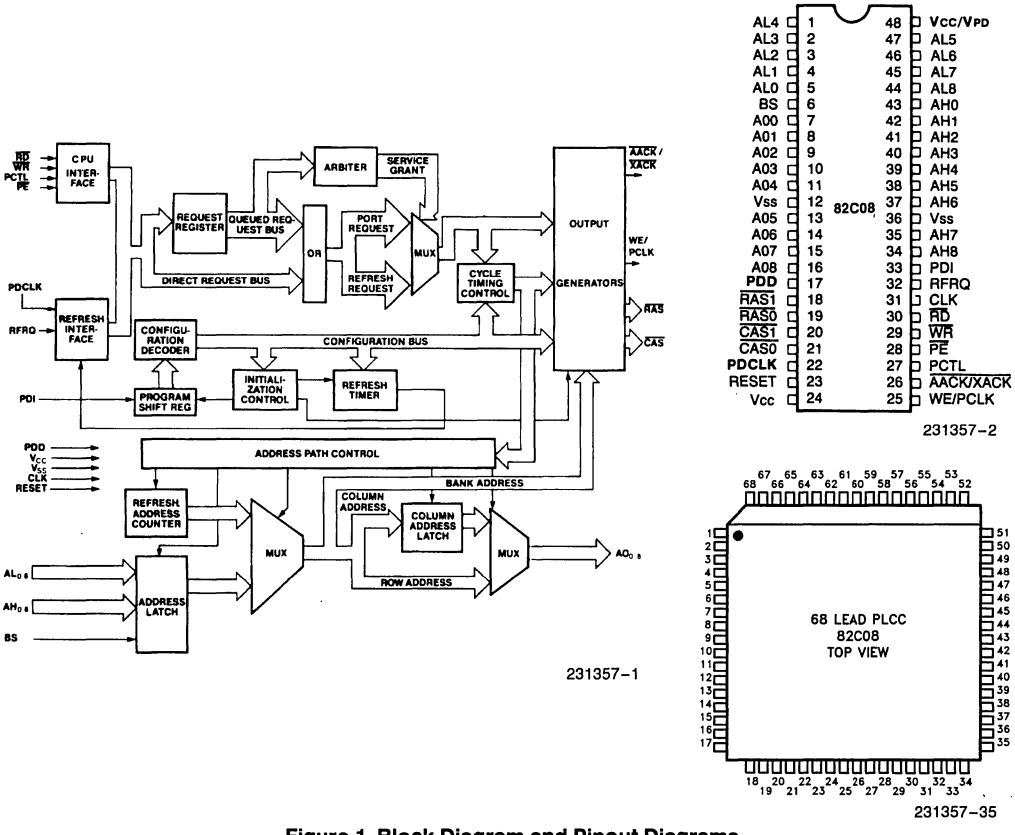


Figure 1. Block Diagram and Pinout Diagrams

The complete document for this product is available from Intel's Literature Center at 1-800-548-4725.

September 1990
Order Number: 231357-008

intel®

4

UPI Keyboard Controllers

intel®

Microprocessor Peripherals
UPI-41A/41AH/42/42AH
User's Manual

4

March 1994

Order Number: 231318-006

Microprocessor Peripherals
UPI-41A/41AH/42/42AH User's Manual

CONTENTS

CONTENTS

CHAPTER 1
INTRODUCTION

Accompanying the introduction of microprocessors such as the 8088, 8086, 80186 and 80286 there has been a rapid proliferation of intelligent peripheral devices. These special purpose peripherals extend CPU performance and flexibility in a number of important ways.

Table 1-1. Intelligent Peripheral Devices

8255 (GPIO)	Programmable Peripheral Interface
8251A(USART)	Programmable Communication Interface
8253 (TIMER)	Programmable Interval Timer
8257 (DMA)	Programmable DMA Controller
8259	Programmable Interrupt Controller
82077AA	Programmable Floppy Disk Controller
8273 (SDLC)	Programmable Synchronous Data Link Controller
8274	Programmable Multiprotocol-Serial Communications Controller
8275/8276 (CRT)	Programmable CRT Controllers
8279 (PKD)	Programmable Keyboard/Display Controller
8291A, 8292, 8293	Programmable GPIB System Talker, Listener, Controller

Intelligent devices like the 82077AA floppy disk controller and 8273 synchronous data link controller (see Table 1-1) can preprocess serial data and perform control tasks which off-load the main system processor. Higher overall system throughput is achieved and software complexity is greatly reduced. The intelligent peripheral chips simplify master processor control tasks by performing many functions externally in peripheral hardware rather than internally in main processor software.

Intelligent peripherals also provide system flexibility. They contain on-chip mode registers which are programmed by the master processor during system initialization. These control registers allow the peripheral to be configured into many different operation modes. The user-defined program for the peripheral is stored in main system memory and is transferred to the peripheral's registers whenever a mode change is required. Of course, this type of flexibility requires software overhead in the master system which tends to limit the benefit derived from the peripheral chip.

In the past, intelligent peripherals were designed to handle very specialized tasks. Separate chips were designed for communication disciplines, parallel I/O, keyboard encoding, interval timing, CRT control, etc. Yet, in spite of the large number of devices available and the increased flexibility built into these chips, there is still a large number of microcomputer peripheral control tasks which are not satisfied.

With the introduction of the Universal Peripheral Interface (UPI) microcomputer, Intel has taken the intelligent peripheral concept a step further by providing an intelligent controller that is fully user programmable. It is a complete single-chip microcomputer which can connect directly to a master processor data bus. It has the same advantages of intelligence and flexibility which previous peripheral chips offered. In addition, UPIs are user-programmable: it has 1K/2K bytes of ROM or EPROM memory for program storage plus 64/128/256 bytes of RAM memory UPI-41A, 41AH/42, 42AH respectively for data storage or initialization from the master processor. The UPI device allows a designer to fully specify his control algorithm in the peripheral chip without relying on the master processor. Devices like printer controllers and keyboard scanners can be completely self-contained, relying on the master processor only for data transfer.

The UPI family currently consists of seven components:

- 8741A microcomputer with 1K EPROM memory
- 8741AH microcomputer with 1K OTP EPROM memory
- 8041AH microcomputer with 1K ROM memory
- 8742 microcomputer with 2K EPROM memory
- 8742AH microcomputer with 2K "OTP" EPROM memory
- 8042AH microcomputer with 2K ROM memory
- 8243 I/O expander device

The UPI-41A/41AH/42/42AH family of microcomputers are functionally equivalent except for the type and amount of program memory available with each. In addition, the UPI-41AH/42AH family has a Signature Row outside the EPROM Array. The UPI-41AH/42AH family also has a Security Feature which renders the EPROM Array unreadable when set.

All UPI's have the following main features:

- 8-bit CPU
- 8-bit data bus interface registers
- Interval timer/event counter
- Two 8-bit TTL compatible I/O ports
- Resident clock oscillator circuits

The UPI family has the following differences:

Table 1-2

UPI-41A	UPI-42	UPI-41AH	UPI-42AH
1K x 8 EPROM	2K x 8 EPROM	1K x 8 ROM or 1K x 8 OTP	2K x 8 ROM or 2K x 8 OTP
64 x 8 RAM	128 x 8 RAM	128 x 8 RAM	256 x 8 RAM
		*Set Security Feature **Signature Row Feature 32 Bytes with: 1. Test Code/Checksum 2. Intel Signature 3. Security Byte 4. User Signature	
PROGRAMMING			
UPI-41A	**UPI-42**	**UPI-41AH/UPI-42AH**	
V_{DD} = 25V	21V	12.5V	
I_{DD} = 50 ms	50 mA	30 mA	
EA = 21.5V–24.5V	18V	12.5V	
V_{PH} = 21.5V–24.5V	18V	20.V–5.5V	
TPW = 50 ms	50 ms	1 ms	
PIN DESCRIPTION			
UPI-41A/UPI-42		**UPI-41AH/UPI-42AH**	
(T1) T1 functions as a test input which can be directly tested using conditional branching instructions. It functions as the event timer input under software control.		T1 functions as a test input that can be directly tested using conditional branching instructions. It works as the event timer input under software control. It is used during sync mode to reset the instruction state to S1 and synchronize the internal clock to phase 1.	
(SS) Single step input used with the sync output to step the program through each instruction.		Single step input used with the sync output to step the program through each instruction. This pin is used to put the device in sync mode by applying +12.5V to it.	
Port 1 (P10–P17): 8-bit, Quasi-Bidirectional I/O Lines.		Port 1 (P10–P17): 8-bit, Quasi-Bidirectional I/O Lines. P10–P17 access the Signature Row and Security Bit.	

NOTES:
*For a complete description of the Security Feature, refer to the UPI-41AH/42AH Datasheet.
**For a complete description of the Signature Row, refer to the UPI-41AH/42AH Datasheet.

HMOS processing has been applied to the UPI family to allow for additional performance and memory capability while reducing costs. The UPI-41A/41AH/42/42AH are all pin and software compatible. This allows growth in present designs to incorporate new features and add additional performance. For new designs, the additional memory and performance of the UPI-41A/41AH/42/42AH extends the UPI 'grow your own solution' concept to more complex motor control tasks, 80-column printers and process control applications as examples.

The 8243 device is an I/O multiplexer which allows expansion of I/O to over 100 lines (if seven devices are used). All three parts are fabricated with N-channel MOS technology and require a single, 5V supply for operation.

INTERFACE REGISTERS FOR MULTI-PROCESSOR CONFIGURATIONS

In the normal configuration, the UPI-41A/41AH/42/42AH interfaces to the system bus, just like any intelligent peripheral device (see Figure 1-1). The host processor and the UPI-41A/41AH/42/42AH form a loosely coupled multi-processor system, that is, communications between the two processors are direct. Common resources are three addressable registers located physically on the UPI-41A/41AH/42/42AH. These registers are the Data Bus Buffer Input (DBBIN), Data Bus Buffer Output (DBBOUT), and Status (STATUS) registers. The host processor may read data from DBBOUT or write commands and data into DBBIN. The status of DBBOUT and DBBIN plus user-defined status is supplied in STATUS. The host may read STATUS at any time. An interrupt to the UPI processor is automatically generated (if enabled) when DBBIN is loaded.

Because the UPI contains a complete microcomputer with program memory, data memory, and CPU it can function as a "Universal" controller. A designer can program the UPI to control printers, tape transports, or multiple serial communication channels. The UPI can also handle off-line arithmetic processing, or any number of other low speed control tasks.

POWERFUL 8-BIT PROCESSOR

The UPI contains a powerful, 8-bit CPU with as fast as 1.2 μsec cycle time and two single-level interrupts. Its instruction set includes over 90 instructions for easy software development. Most instructions are single byte and single cycle and none are more than two bytes long. The instruction set is optimized for bit manipulation and I/O operations. Special instructions are included to allow binary or BCD arithmetic operations, table look-up routines, loop counters, and N-way branch routines.

4

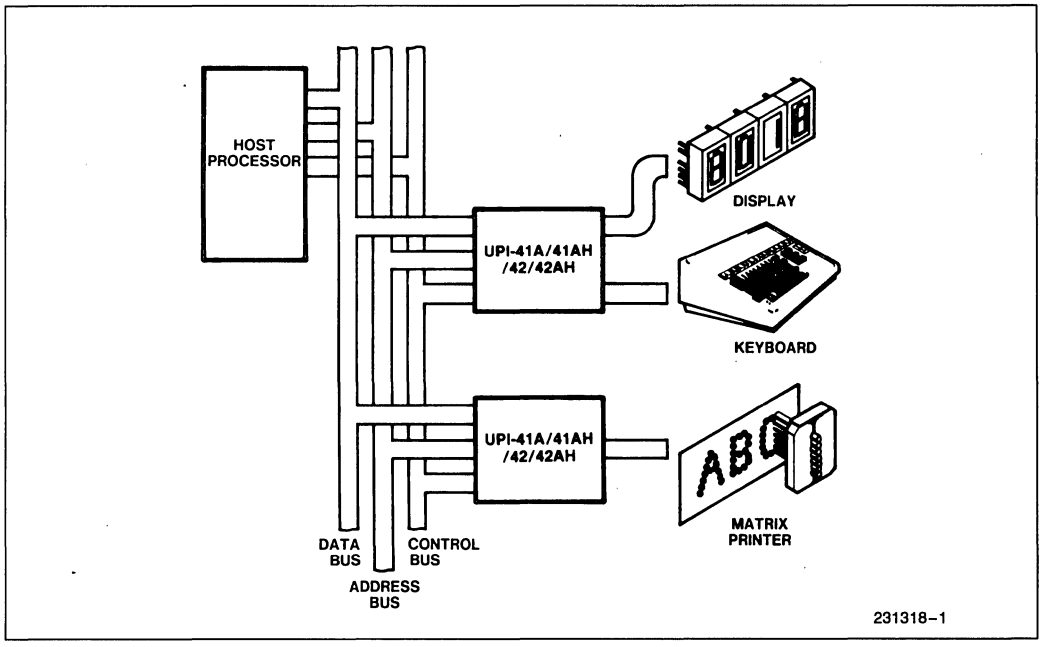


Figure 1-1. Interfacing Peripherals To Microcomputer Systems

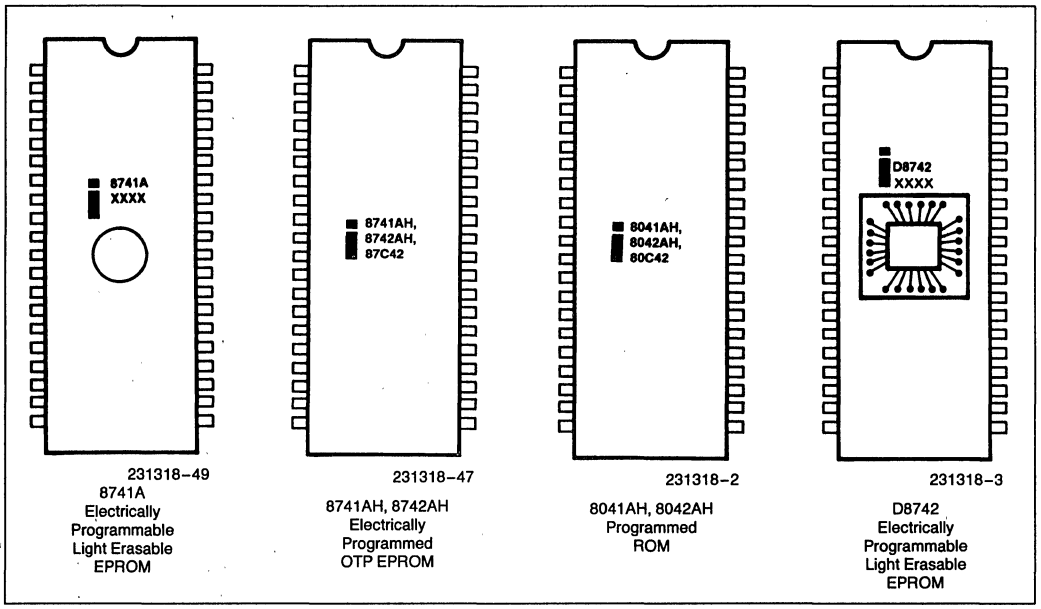


Figure 1-2. Pin Compatible ROM/EPROM Versions

SPECIAL INSTRUCTION SET FEATURES

- For Loop Counters:
  Decrement Register and Jump if not zero.
- For Bit Manipulation:
  AND to A (immediate data or Register)
  OR to A (immediate data or Register)
  XOR to A (immediate data or Register)
  AND to Output Ports (Accumulator)
  OR to Output Ports (Accumulator)
  Jump Conditionally on any bit in A
- For BDC Arithmetic:
  Decimal Adjust A
  Swap 4-bit Nibbles of A
  Exchange lower nibbles of A and Register
  Rotate A left or right with or without Carry
- For Lookup Tables:
  Load A from Page of ROM (Address in A)
  Load A from Current Page of ROM (Address in A)

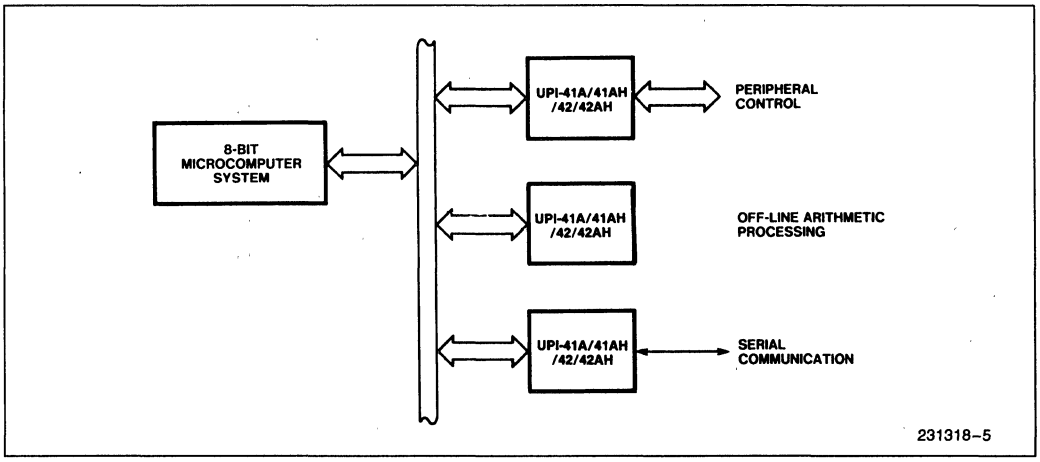


Figure 1-3. Interfaces and Protocols for Multiprocessor Systems

Features for Peripheral Control

The UPI 8-bit interval timer/event counter can be used to generate complex timing sequences for control applications or it can count external events such as switch closures and position encoder pulses. Software timing loops can be simplified or eliminated by the interval timer. If enabled, an interrupt to the CPU will occur when the timer overflows.

The UPI I/O complement contains two TTL-compatible 8-bit bidirectional I/O ports and two general-purpose test inputs. Each of the 16 port lines can individually function as either input or output under software control. Four of the port lines can also function as an interface for the 8243 I/O expander which provides four additional 4-bit ports that are directly addressable by UPI software. The 8243 expander allows low cost I/O expansion for large control applications while maintaining easy and efficient software port addressing.

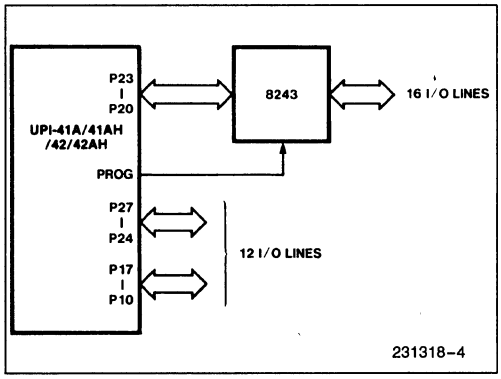


Figure 1-4. 8243 I/O Expander Interface

On-Chip Memory

The UPI's 64/128/256 bytes data memory include dual working register banks and an 8-level program counter stack. Switching between the register banks allows fast response to interrupts. The stack is used to store return addresses and processor status upon entering a subroutine.

The UPI program memory is available in three types to allow flexibility in moving from design to prototype to production with the same PC layout. The 8741A/8742 device with EPROM memory is very economical for initial system design and development. Its program memory can be electrically programmed using the Intel Universal PROM Programmer. When changes are needed, the entire program can be erased using UV lamp and reprogrammed in about 20 minutes. This means the 8741A/8742 can be used as a single chip "breadboard" for very complex interface and control problems. After the 8741A/8742 is programmed it can be tested in the actual production level PC board and the actual functional environment. Changes required during system debugging can be made in the 8741A/8742 program much more easily than they could be made in a random logic design. The system configuration and PC layout can remain fixed during the development process and the turn around time between changes can be reduced to a minimum.

At any point during the development cycle, the 8741A/8742 EPROM part can be replaced with the low cost UPI-41AH/42AH respectively with factory mask programmed memory or OTP EPROM. The transition from system development to mass production is made smoothly because the 8741A/8742, 8741AH and 8041AH, 8742AH and 8042AH parts are completely pin compatible. This feature allows extensive testing with the EPROM part, even into initial shipments to customers. Yet, the transition to low-cost ROMs or OTP EPROM is simplified to the point of being merely a package substitution.

PREPROGRAMMED UPI's

The 8242AH, 8292, and 8294 are 8042AH's that are programmed by Intel and sold as standard peripherals. Intel offers a complete line of factory programmed keyboard controllers. These devices contain firmware developed by Phoenix Technologies Ltd. and Award Software Inc. See Table 1-3 for a complete listing of Intels' entire keyboard controller product line. The 8292 is a GPIB controller, part of a three chip GPIB system. The 8294 is a Data Encryption Unit that implements the National Bureau of Standards data encryption algorithm. These parts illustrate the great flexibility offered by the UPI family.

Table 1-3. Keyboard Controller Family Product Selection Guide

UPI-42: The industry standard for desktop Keyboard Control.

Device	Package	ROM	OTP	Comments
8042	N, P	2K		ROM Device
8242	N, P			Phoenix firmware version 2.5
8242PC	N, P			Phoenix MultiKey/42 firmware, PS/2 style mouse support
8242WA	N, P			Award firmware version 3.57
8242WB	N, P			Award firmware version 4.14, PS/2 style mouse support
8742	N, P, D		2K	Available as OTP (N, P) or EPROM (D)

UPI-C42: A low power CHMOS version of the UPI-42. The UPI-C42 doubles the user programmable memory size, adds Auto A20 Gate support, includes Standby (**) and Suspend power down modes, and is available in a space saving 44-lead QFP pkg.

Device	Package	ROM	OTP	Comments
80C42	N, P, S	4K		ROM Device
82C42PC	N, P, S			Phoenix MultiKey/42 firmware, PS/2 style mouse support
82C42PD	N, P, S			Phoenix MultiKey/42L firmware, KBC and SCC for portable apps.
82C42PE	N, P, S			Phoenix MultiKey/42G firmware, Energy Efficient KBC solution
87C42	N, P, S		4K	One Time Programmable Version

UPI-L42: The low voltage 3.3V version of the UPI-C42.

Device	Package	ROM	OTP	Comments
80L42	N, P, S	4K		ROM Device
82L42PC	N, P, S			Phoenix MultiKey/42 firmware, PS/2 style mouse support
82L42PD	N, P, S			Phoenix MultiKey/42L firmware, KBC and SCC for portable apps.
87L42	N, P, S		4K	One Time Programmable Version

NOTES:
N = 44 lead PLCC, P = 40 lead PDIP, S = 44 lead QFP, D = 40 lead CERDIP
KBC = Key Board Control, SCC = Scan Code Control
(**) Standby feature not supported on current (B-1) stepping

DEVELOPMENT SUPPORT

The UPI microcomputer is fully supported by Intel with development tools like the UPP PROM programmer already mentioned. The combination of device features and Intel development support make the UPI an ideal component for low-speed peripheral control applications.

UPI DEVELOPMENT SUPPORT

- 8048/UPI-41A/41AH/42/42AH Assembler
- Universal PROM Programmer UPP Series
- Application Engineers
- Training Courses

CHAPTER 2
FUNCTIONAL DESCRIPTION

The UPI microcomputer is an intelligent peripheral controller designed to operate in iAPX-86, 88, MCS-85, MCS-80, MCS-51 and MCS-48 systems. The UPI's architecture, illustrated in Figure 2-1, is based on a low cost, single-chip microcomputer with program memory, data memory, CPU, I/O, event timer and clock oscillator in a single 40-pin package. Special interface registers are included which enable the UPI to function as a peripheral to an 8-bit master processor.

This chapter provides a basic description of the UPI microcomputer and its system interface registers. Unless otherwise noted the descriptions in this section apply to the 8741AH, 8742AH with OTP EPROM memory, the 8741A/8742 (with UV erasable program memory) and the 8041AH, 8042AH. These devices are so similar that they can be considered identical under most circumstances. All functions described in this chapter apply to the UPI-41A/41AH/42/42AH.

PIN DESCRIPTION

The UPI-41A/41AH/42/42AH are packaged in 40-pin Dual In-Line (DIP) packages. The pin configuration for both devices is shown in Figure 2-2. Figure 2-3 illustrates the UPI Logic Symbol.

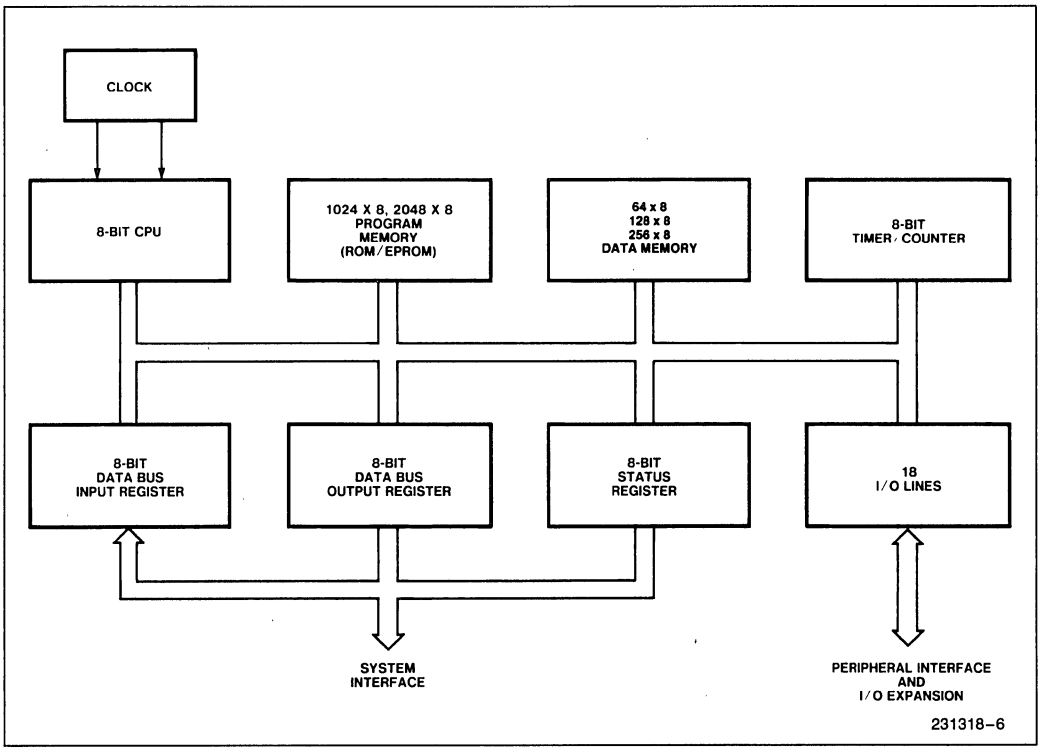


Figure 2-1. UPI-41A/41AH/42/42AH Single Chip Microcomputer

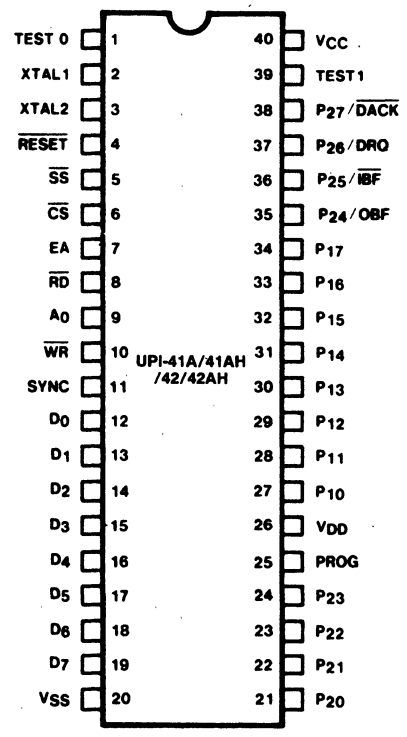


Figure 2-2. Pin Configuration

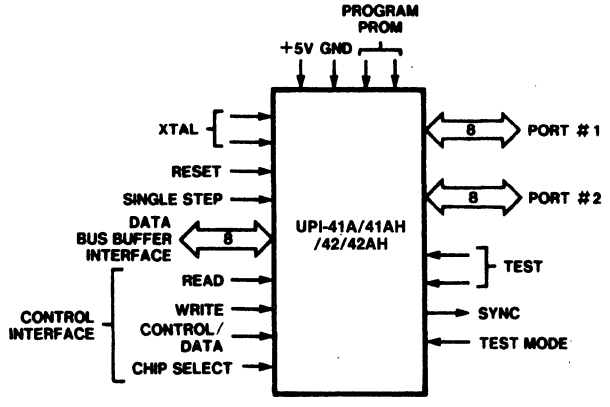


Figure 2-3. Logic Symbol

The following section summarizes the functions of each UPI pin. NOTE that several pins have two or more functions which are described in separate paragraphs.

Table 2-1. Pin Description

Symbol	Pin No.	Type	Name and Function
D_0–D_7 (BUS)	12–19	I/O	**DATA BUS:** Three-state, bidirectional DATA BUS BUFFER lines used to interface the UPI-41A/41AH/42/42AH microcomputer to an 8-bit master system data bus.
P_{10}–P_{17}	27-34	I/O	**PORT 1:** 8-bit, PORT 1 quasi-bidirectional I/O lines.
P_{20}–P_{27}	21-24 35–38	I/O	**PORT 2:** 8-bit, PORT 2 quasi-bidirectional I/O lines. The lower 4 bits (P_{20}–P_{23}) interface directly to the 8243 I/O expander device and contain address and data information during PORT 4–7 access. The upper 4 bits (P_{24}–P_{27}) can be programmed to provide interrupt Request and DMA Handshake capability. Software control can configure P_{24} as Output Buffer Full (OBF) interrupt, P_{25} as Input Buffer Full ($\overline{IBF}$) interrupt, P_{26} as DMA Request ($\overline{DRQ}$), and P_{27} as DMA ACKnowledge ($\overline{DACK}$).
$\overline{WR}$	10	I	**WRITE:** I/O write input which enables the master CPU to write data and command words to the UPI INPUT DATA BUS BUFFER.
$\overline{RD}$	8	I	**READ:** I/O read input which enables the master CPU to read data and status words from the OUTPUT DATA BUS BUFFER or status register.
$\overline{CS}$	6	I	**CHIP SELECT:** Chip select input used to select one UPI-41A/41AH/42/42AH microcomputer out of several connected to a common data bus.
A_0	9	I	**COMMAND/DATA SELECT:** Address input used by the master processor to indicate whether byte transfer is data (A_0 = 0) or command (A_0 = 1).
TEST 0, TEST 1	1 39	I	**TEST INPUTS:** Input pins can be directly tested using conditional branch instructions. **FREQUENCY REFERENCE:** TEST 1 (T_1) also functions as the event timer input (under software control). TEST0 (T_0) is used during PROM programming and verification in the UPI-41A/41AH/42/42AH.
XTAL 1, XTAL 2	2 3	I	**INPUTS:** Inputs for a crystal, LC or an external timing signal to determine the internal oscillator frequency.
SYNC	11	O	**OUTPUT CLOCK:** Output signal which occurs once per UPI instruction cycle. SYNC can be used as a strobe for external circuitry; it is also used to synchronize single step operation.
EA	7	I	**EXTERNAL ACCESS:** External access input which allows emulation, testing and PROM/ROM verification.
PROG	25	I/O	**PROGRAM:** Multifunction pin used as the program pulse input during PROM programming. During I/O expander access the PROG pin acts as an address/data strobe to the 8243.
$\overline{RESET}$	4	I	**RESET:** Input used to reset status flip-flops and to set the program counter to zero. $\overline{RESET}$ is also used during PROM programming and verification.
$\overline{SS}$	5	I	**SINGLE STEP:** Single step input used in conjunction with the SYNC output to step the program through each instruction.
V_{CC}	40		**POWER:** +5V main power supply pin.
V_{DD}	26		**POWER:** +5V during normal operation. +25V for UPI-41A, 21V for UPI-42 programming operation, +12V for programming, UPI-41AH/42AH. Low power standby pin in ROM version.
V_{SS}	20		**GROUND:** Circuit ground potential.

The following sections provide a detailed functional description of the UPI microcomputer. Figure 2-4 illustrates the functional blocks within the UPI device.

CPU SECTION

The CPU section of the UPI-41A/41AH/42/42AH microcomputer performs basic data manipulations and controls data flow throughout the single chip computer via the internal 8-bit data bus. The CPU section includes the following functional blocks shown in Figure 2-4:

- Arithmetic Logic Unit (ALU)
- Instruction Decoder
- Accumulator
- Flags

Arithmetic Logic Units (ALU)

The ALU is capable of performing the following operations:

- ADD with or without carry
- AND, OR, and EXCLUSIVE OR
- Increment, Decrement
- Bit complement
- Rotate left or right
- Swap
- BCD decimal adjust

In a typical operation data from the accumulator is combined in the ALU with data from some other source on the UPI-41A/41AH/42/42AH internal bus (such as a register or an I/O port). The result of an ALU operation can be transferred to the internal bus or back to the accumulator.

If an operation such as an ADD or ROTATE requires more than 8 bits, the CARRY flag is used as an indicator. Likewise, during decimal adjust and other BCD operations the AUXILIARY CARRY flag can be set and acted upon. These flags are part of the Program Status Word (PSW).

Instruction Decoder

During an instruction fetch, the operation code (opcode) portion of each program instruction is stored and decoded by the instruction decoder. The decoder generates outputs used along with various timing signals to control the functions performed in the ALU. Also, the instruction decoder controls the source and destination of ALU data.

Accumulator

The accumulator is the single most important register in the processor. It is the primary source of data to the ALU and is often the destination for results as well. Data to and from the I/O ports and memory normally passes through the accumulator.

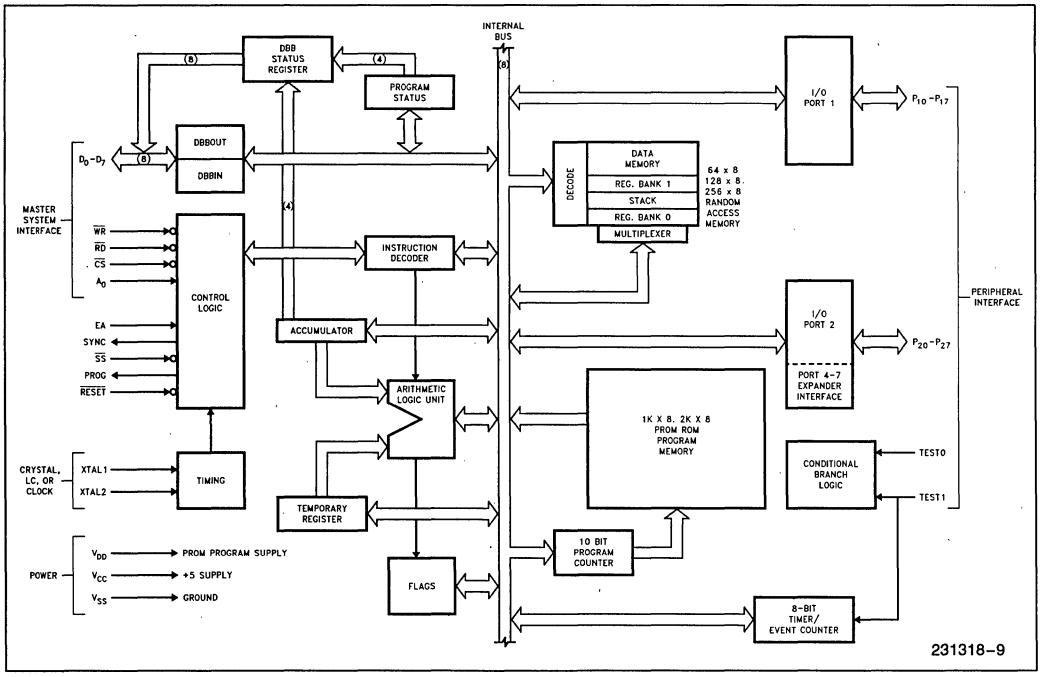


Figure 2-4. UPI-41A/41AH/42/42AH Block Diagram

PROGRAM MEMORY

The UPI-41A/41AH/42/42AH microcomputer has 1024, 2048 8-bit words of resident, read-only memory for program storage. Each of these memory locations is directly addressable by a 10-bit program counter. Depending on the type of application and the number of program changes anticipated, three types of program memory are available:

- 8041AH, 8042AH with mask programmed ROM Memory
- 8741AH, 8742AH with electrically programmable OTP EPROM Memory
- 8741A and 8742 with electrically programmable EPROM Memory

A program memory map is illustrated in Figure 2-5. Memory is divided into 256 location 'pages' and three locations are reserved for special use:

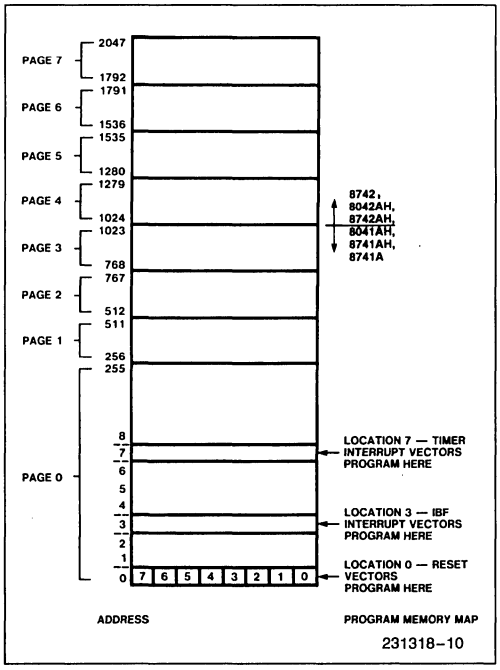


Figure 2-5. Program Memory Map

INTERRUPT VECTORS

1) **Location 0**
   Following a $\overline{\text{RESET}}$ input to the processor, the next instruction is automatically fetched from location 0.
2) **Location 3**
   An interrupt generated by an Input Buffer Full (IBF) condition (when the IBF interrupt is enabled) causes the next instruction to be fetched from location 3.
3) **Location 7**
   A timer overflow interrupt (when enabled) will cause the next instruction to be fetched from location 7.

Following a system $\overline{\text{RESET}}$, program execution begins at location 0. Instructions in program memory are normally executed sequentially. Program control can be transferred out of the main line of code by an input buffer full (IBF) interrupt or a timer interrupt, or when a jump or call instruction is encountered. An IBF interrupt (if enabled) will automatically transfer control to location 3 while a timer interrupt will transfer control to location 7.

All conditional JUMP instructions and the indirect JUMP instruction are limited in range to the current 256-location page (that is, they alter PC bits 0–7 only). If a conditional JUMP or indirect JUMP begins in location 255 of a page, it must reference a destination on the following page.

Program memory can be used to store constants as well as program instructions. The UPI-41AH, 42AH instruction set contains an instruction (MOVP3) designed specifically for efficient transfer of look-up table information from page 3 of memory.

DATA MEMORY

The UPI-41A has 64 8-bit words of Random Access Memory, the UPI-41AH has 128 8-bit words of Random Access Memory; the UPI-42 has 128 8-bit words of RAM; and the UPI-42AH has 256 8-bit words of RAM. This memory contains two working register banks, an 8-level program counter stack and a scratch pad memory, as shown in Figure 2-6. The amount of scratch pad memory available is variable depending on the number of addresses nested in the stack and the number of working registers being used.

Addressing Data Memory

The first eight locations in RAM are designated as working registers R_0–R_7. These locations (or registers) can be addressed directly by specifying a register number in the instruction. Since these locations are easily addressed, they are generally used to store frequently

accessed intermediate results. Other locations in data memory are addressed indirectly by using R_0 or R_1 to specify the desired address.

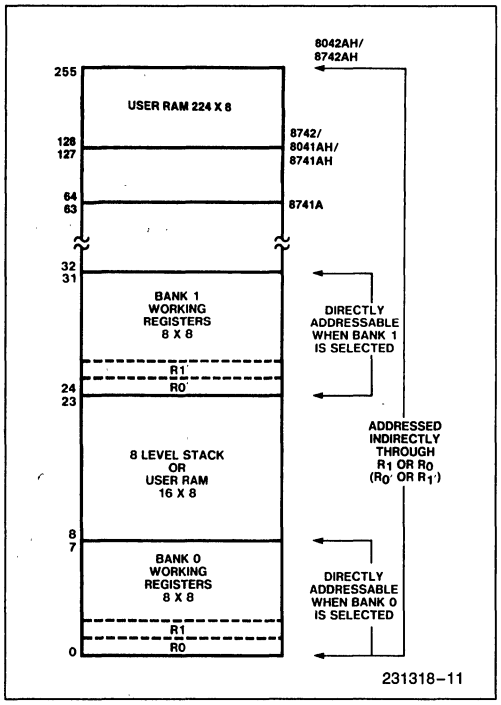


Figure 2-6. Data Memory Map

Working Registers

Dual banks of eight working registers are included in the UPI-41A/41AH/42/42AH data memory. Locations 0–7 make up register bank 0 and locations 24–13 form register bank 1. A $\overline{\text{RESET}}$ signal automatically selects register bank 0. When bank 0 is selected, references to R_0–R_7 in UPI-41A/41AH/42/42AH instructions operate on locations 0–7 in data memory. A "select register bank" instruction is used to selected between the banks during program execution. If the instruction SEL RB1 (Select Register Bank 1) is executed, then program references to R_0–R_7 will operate on locations 24–31. As stated previously, registers 0 and 1 in the active register bank are used as indirect address registers for all locations in data memory.

Register bank 1 is normally reserved for handling interrupt service routines, thereby preserving the contents of the main program registers. The SEL RB1 instruction can be issued at the beginning of an interrupt service routine. Then, upon return to the main program, an RETR (return & restore status) instruction will automatically restore the previously selected bank. During interrupt processing, registers in bank 0 can be accessed indirectly using $R_{0'}$ and $R_{1'}$.

If register bank 1 is not used, registers 24–31 can still serve as additional scratch pad memory.

Program Counter Stack

RAM locations 8–23 are used as an 8-level program counter stack. When program control is temporarily passed from the main program to a subroutine or interrupt service routine, the 10-bit program counter and bits 4–7 of the program status word (PSW) are stored in two stack locations. When control is returned to the main program via an RETR instruction, the program counter and PSW bits 4–7 are restored. Returning via an RET instruction does not restore the PSW bits, however. The program counter stack is addressed by three stack pointer bits in the PSW (bits 0–2). Operation of the program counter stack and the program status word is explained in detail in the following sections.

The stack allows up to eight levels of subroutine 'nesting'; that is, a subroutine may call a second subroutine, which may call a third, etc., up to eight levels. Unused stack locations can be used as scratch pad memory. Each unused level of subroutine nesting provides two additional RAM locations for general use.

The following sections provide a detailed description of the Program Counter Stack and the Program Status Word.

PROGRAM COUNTER

The UPI-41A/41AH/42/42AH microcomputer has a 10-bit program counter (PC) which can directly address any of the 1024, 2048, or 4096 locations in program memory. The program counter always contains the address of the next instruction to be executed and is normally incremented sequentially for each instruction to be executed when each instruction fetches occurs.

When control is temporarily passed from the main program to a subroutine or an interrupt routine, however, the PC contents must be altered to point to the address of the desired routine. The stack is used to save the current PC contents so that, at the end of the routine, main program execution can continue. The program counter is initialized to zero by a $\overline{\text{RESET}}$ signal.

PROGRAM COUNTER STACK

The Program Counter Stack is composed of 16 locations in Data Memory as illustrated in Figure 2-7. These RAM locations (8 through 23) are used to store the 10-bit program counter and 4 bits of the program status word.

An interrupt or Call to a subroutine causes the contents of the program counter to be stored in one of the 8 register pairs of the program counter stack.

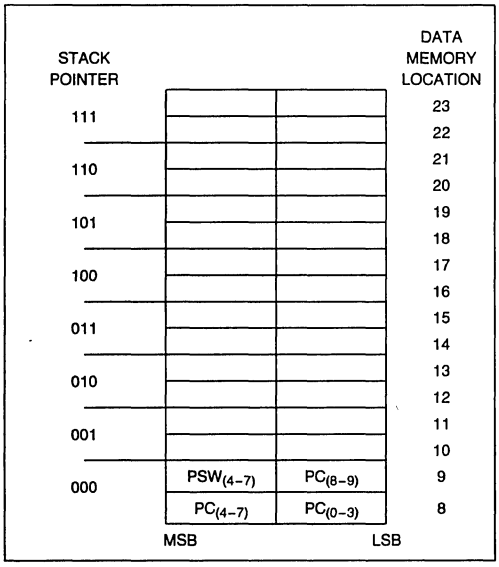


Figure 2-7. Program Counter Stack

A 3-bit Stack Pointer which is part of the Program Status Word (PSW) determines the stack pair to be used at a given time. The stack pointer is initialized by a $\overline{\text{RESET}}$ signal to 00H which corresponds to RAM locations 8 and 9.

The first call or interrupt results in the program counter and PSW contents being transferred to RAM locations 8 and 9 in the format shown in Figure 2-7. The stack pointer is automatically incremented by 1 to point to location is 10 and 11 in anticipation of another CALL.

Nesting of subroutines within subroutines can continue up to 8 levels without overflowing the stack. If overflow does occur the deepest address stored (locations 8 and 9) will be overwritten and lost since the stack pointer overflows from 07H to 00H. Likewise, the stack pointer will underflow from 00H to 07H.

The end of a subroutine is signaled by a return instruction, either RET or RETR. Each instruction will automatically decrement the Stack Pointer and transfer the contents of the proper RAM register pair to the Program Counter.

PROGRAM STATUS WORD

The 8-bit program status word illustrated in Figure 2-8 is used to store general information about program execution. In addition to the 3-bit Stack Pointer discussed previously, the PSW includes the following flags:

- CY — Carry
- AC — Auxiliary Carry
- F_0 — Flag 0
- BS — Register Bank Select

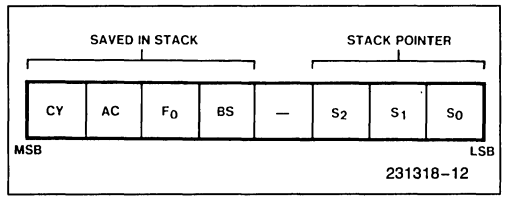


Figure 2-8. Program Status Word

The Program Status Word (PSW) is actually a collection of flip-flops located throughout the machine which are read or written as a whole. The PSW can be loaded to or from the accumulator by the MOV A, PSW or MOV PSW, A instructions. The ability to write directly to the PSW allows easy restoration of machine status after a power-down sequence.

The upper 4 bits of the PSW (bits 4, 5, 6, and 7) are stored in the PC Stack with every subroutine CALL or interrupt vector. Restoring the bits on a return is optional. The bits are restored if an RETR instruction is executed, but not if an RET is executed.

PSW bit definitions are as follows:

- Bits 0–2 Stack Pointer Bits S_0, S_1, S_2
- Bit 3 Not Used
- Bit 4 Working Register Bank
  0 = Bank 0
  1 = Bank 1
- Bit 5 Flag 0 bit (F_0)
  This is a general purpose flag which can be cleared or complemented and tested with conditional jump instructions. It may be used during data transfer to an external processor.
- Bit 6 Auxiliary Carry (AC)
  The flag status is determined by an ADD instruction and is used by the Decimal Adjustment instruction DAA
- Bit 7 Carry (CY)
  The flag indicates that a previous operation resulted in overflow of the accumulator.

CONDITIONAL BRANCH LOGIC

Conditional Branch Logic in the UPI-41AH, 42AH allows the status of various processor flags, inputs, and other hardware functions to directly affect program execution. The status is sampled in state 3 of the first cycle.

Table 2-2 lists the internal conditions which are testable and indicates the condition which will cause a jump. In all cases, the destination address must be within the page of program memory (256 locations) in which the jump instruction occurs.

OSCILLATOR AND TIMING CIRCUITS

The UPI-41A/41AH/42/42AH's internal timing generation is controlled by a self-contained oscillator and timing circuit. A choice of crystal, L-C or external clock can be used to derive the basic oscillator frequency.

The resident timing circuit consists of an oscillator, a state counter and a cycle counter as illustrated in Figure 2-9. Figure 2-10 shows instruction cycle timing.

Oscillator

The on-board oscillator is a series resonant circuit with a frequency range of 1 to 12.5 MHz depending on which UPI is used. Refer to Table 1.1. Pins XTAL 1 and XTAL 2 are input and output (respectively) of a high gain amplifier stage. A crystal or inductor and capacitor connected between XTAL 1 and XTAL 2 provide the feedback and proper phase shift for oscillation. Recommended connections for crystal or L-C are shown in Figure 2-11.

State Counter

The output of the oscillator is divided by 3 in the state counter to generate a signal which defines the state times of the machine.

Each instruction cycle consists of five states as illustrated in Figure 2-10 and Table 2-3. The overlap of address and execution operations illustrated in Figure 2-10 allows fast instruction execution.

Table 2-2. Conditional Branch Instructions

Device	Instruction Mnemonic		Jump Condition Jump if:
Accumulator	JZ	addr	All bits zero
	JNZ	addr	Any bit not zero
Accumulator bit	JBb	addr	Bit "b" = 1
Carry flag	JC	addr	Carry flag = 1
	JNC	addr	Carry flag = 0
User flag	JFO	addr	F_0 flag = 1
	JF1	addr	F_1 flag = 1
Timer flag	JTF	addr	Timer flag = 1
Test Input 0	JT0	addr	T_0 = 1
	JNT0	addr	T_0 = 0
Test Input 1	JT1	addr	T_1 = 1
	JNT1	addr	T_1 = 0
Input Buffer flag	JNIBF	addr	IBF flag = 0
Output Buffer flag	JOBF	addr	OBF flag = 1

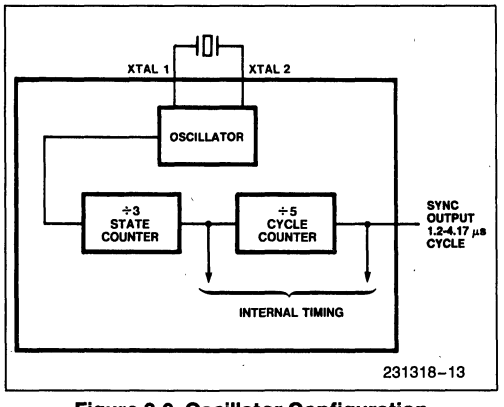


Figure 2-9. Oscillator Configuration

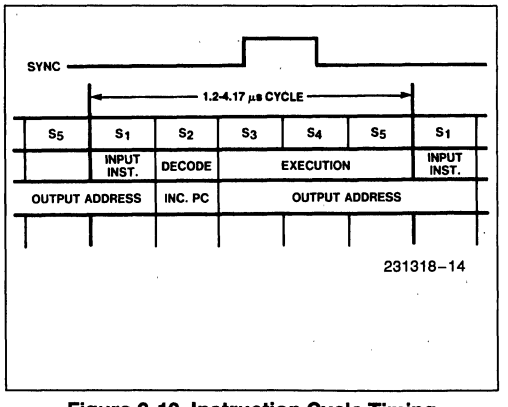


Figure 2-10. Instruction Cycle Timing

Table 2-3. Instruction Timing Diagram

Instruction	CYCLE 1					CYCLE 2				
	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
IN A,Pp	Fetch Instruction	Increment Program Counter	—	Increment Timer	—	—	Read Port	—	—	—
OUTL Pp,A	Fetch Instruction	Increment Program Counter	—	Increment Timer	Output To Port	—	—	—	—	—
ANL Pp, DATA	Fetch Instruction	Increment Program Counter	—	Increment Timer	Read Port	Fetch Immediate Data	—	Increment Program Counter	Output To Port	—
ORL Pp, DATA	Fetch Instruction	Increment Program Counter	—	Increment Timer	Read Port	Fetch Immediate Data	—	Increment Program Counter	Output To Port	—
MOVD A,Pp	Fetch Instruction	Increment Program Counter	Output Opcode/Address	Increment Timer	—	—	Read P2 Lower	—	—	—
MOVD Pp, A	Fetch Instruction	Increment Program Counter	Output Opcode/Address	Increment Timer	Output Data To P2 Lower	—	—	—	—	—
D Pp, A	Fetch Instruction	Increment Program Counter	Output Opcode/Address	Increment Timer	Output Data	—	—	—	—	—
ORLD Pp, A	Fetch Instruction	Increment Program Counter	Output Opcode/Address	Increment Timer	Output Data	—	—	—	—	—
J (Conditional)	Fetch Instruction	Increment Program Counter	Sample Condition	Increment Timer	—	Fetch Immediate Data	—	Update Program Counter	—	—
MOV STS, A	Fetch Instruction	Increment Program Counter	—	Increment Timer	Update Status Register					
IN A, DBB	Fetch Instruction	Increment Program Counter	—	Increment Timer	—					
OUT DBB, A	Fetch Instruction	Increment Program Counter	—	Increment Timer	Output To Port					
STRT T STRT CNT	Fetch Instruction	Increment Program Counter	—	—	Start Counter					
STOP TCNT	Fetch Instruction	Increment Program Counter	—	—	Stop Counter					
EN I	Fetch Instruction	Increment Program Counter	—	Enable Interrupt	—					
DIS I	Fetch Instruction	Increment Program Counter	—	Disable Interrupt	—					
EN DMA	Fetch Instruction	Increment Program Counter	—	DMA Enabled DRQ Cleared	—					
EN FLAGS	Fetch Instruction	Increment Program Counter	—	OBF, $\overline{IBF}$ Output Enabled	—					

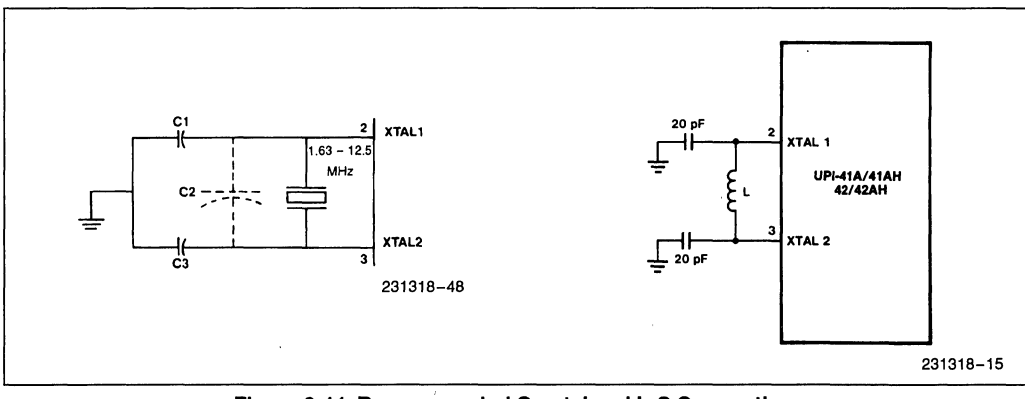


Figure 2-11. Recommended Crystal and L-C Connections

Cycle Counter

The output of the state counter is divided by 5 in the cycle counter to generate a signal which defines a machine cycle. This signal is call SYNC and is available continuously on the SYNC output pin. It can be used to synchronize external circuitry or as a general purpose clock output. It is also used for synchronizing single-step.

Frequency Reference

The external crystal provides high speed and accurate timing generation. A crystal frequency of 5.9904 MHz is useful for generation of standard communication frequencies by the UPI-41A/41AH/42/42AH. However, if an accurate frequency reference and maximum processor speed are not required, an inductor and capacitor may be used in place of the crystal as shown in Figure 2-11.

A recommended range of inductance and capacitance combinations is given below:

- L = 130 μH corresponds to 3 MHz
- L = 45 μH corresponds to 5 MHz

An external clock signal can also be used as a frequency reference to the UPI-41A/41AH/42/42AH; however, the levels are *not* TTL compatible. The signal must be in the 1–12.5 MHz frequency range depending on which UPI is used. Refer to Table 1-2. The signal must be connected to pins XTAL 1 and XTAL 2 by buffers with a suitable pull-up resistor to guarantee that a logic "1" is above 3.8 volts. The recommended connection is shown in Figure 2-12.

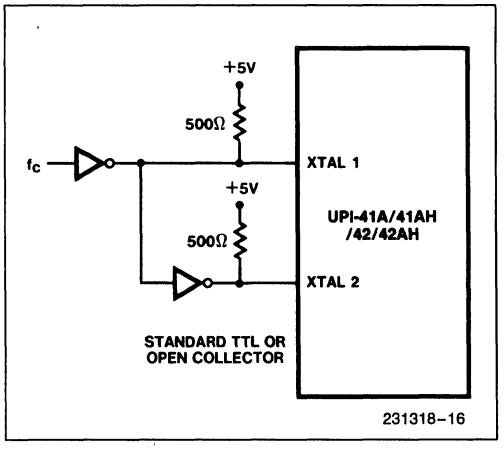


Figure 2-12. Recommended Connection For External Clock Signal

INTERVAL TIMER/EVENT COUNTER

The UPI-41A/41AH/42/42AH has a resident 8-bit timer/counter which has several software selectable modes of operation. As an interval timer, it can generate accurate delays from 80 microseconds to 20.48 milliseconds without placing undue burden on the processor. In the counter mode, external events such as switch closures or tachometer pulses can be counted and used to direct program flow.

Timer Configuration

Figure 2-13 illustrates the basic timer/counter configuration. An 8-bit register is used to count pulses from either the internal clock and prescaler or from an external source. The counter is presettable and readable with two MOV instructions which transfer the contents of the accumulator to the counter and vice-versa (i.e. MOV T, A and MOV A, T). The counter is stopped by a $\overline{\text{RESET}}$ or STOP TCNT instruction and remains stopped until restarted either as a timer (START T instruction) or as a counter (START CNT instruction). Once started, the counter will increment to its maximum count (FFH) and overflow to zero continuing its count until stopped by a STOP TCNT instruction or $\overline{\text{RESET}}$.

The increment from maximum count to zero (overflow) results in setting the Timer Flag (TF) and generating an interrupt request. The state of the overflow flag is testable with the conditional jump instruction, JTF. The flag is reset by executing a JTF or by a $\overline{\text{RESET}}$ signal.

The timer interrupt request is stored in a latch and ORed with the input buffer full interrupt request. The timer interrupt can be enabled or disabled independent of the IBF interrupt by the EN TCNTI and DIS TCTNI instructions. If enabled, the counter overflow will cause a subroutine call to location 7 where the timer service routine is stored. If the timer and Input Buffer Full interrupts occur simultaneously, the IBF source will be recognized and the call will be to location 3. Since the timer interrupt is latched, it will remain pending until the DBBIN register has been serviced and will immediately be recognized upon return from the service routine. A pending timer interrupt is reset by the initiation of a timer interrupt service routine.

Event Counter Mode

The STRT CNT instruction connects the TEST 1 input pin to the counter input and enables the counter. Note this instruction does not clear the counter. The counter is incremented on high to low transitions of the TEST 1 input. The TEST 1 input must remain high for a minimum of one state in order to be registered (250 ns at 12 MHz). The maximum count frequency is one count per three instruction cycles (267 kHz at 12 MHz). There is no minimum frequency limit.

Timer Mode

The STRT T instruction connects the internal clock to the counter input and enables the counter. The input clock is derived from the SYNC signal of the internal oscillator and the divide-by-32 prescaler. The configuration is illustrated in Figure 2-13. Note this instruction does not clear the timer register. Various delays and timing sequences between 40 μsec and 10.24 msec can easily be generated with a minimum of software timing loops (at 12 MHz).

Times longer than 10.24 msec can be accurately measured by accumulating multiple overflows in a register under software control. For time resolution less than 40 μsec, an external clock can be applied to the TEST 1 counter input (see Event Counter Mode). The minimum time resolution with an external clock is 3.75 μsec (267 kHz at 12 MHz).

TEST 1 Event Counter Input

The TEST 1 pin is multifunctional. It is automatically initialized as a test input by a $\overline{\text{RESET}}$ signal and can be tested using UPI-41A conditional branch instructions.

In the second mode of operation, illustrated in Figure 2-13, the TEST 1 pin is used as an input to the internal 8-bit event counter. The Start Counter (STRT CNT) instruction controls an internal switch which connects TEST 1 through an edge detector to the 8-bit internal counter. Note that this instruction does not inhibit the testing of TEST 1 via conditional Jump instructions.

In the counter mode the TEST 1 input is sampled once per instruction cycle. After a high level is detected, the next occurrence of a low level at TEST 1 will cause the counter to increment by one.

The event counter functions can be stopped by the Stop Timer/Counter (STOP TCNT) instruction. When this instruction is executed the TEST 1 pin becomes a test input and functions as previously described.

TEST INPUTS

There are two multifunction pins designated as Test Inputs, TEST 0 and TEST 1. In the normal mode of operation, status of each of these lines can be directly tested using the following conditional Jump instructions:

- JT0 Jump if TEST 0 = 1
- JNT0 Jump if TEST 0 = 0
- JT1 Jump if TEST 1 = 1
- JNT1 Jump if TEST 1 = 0

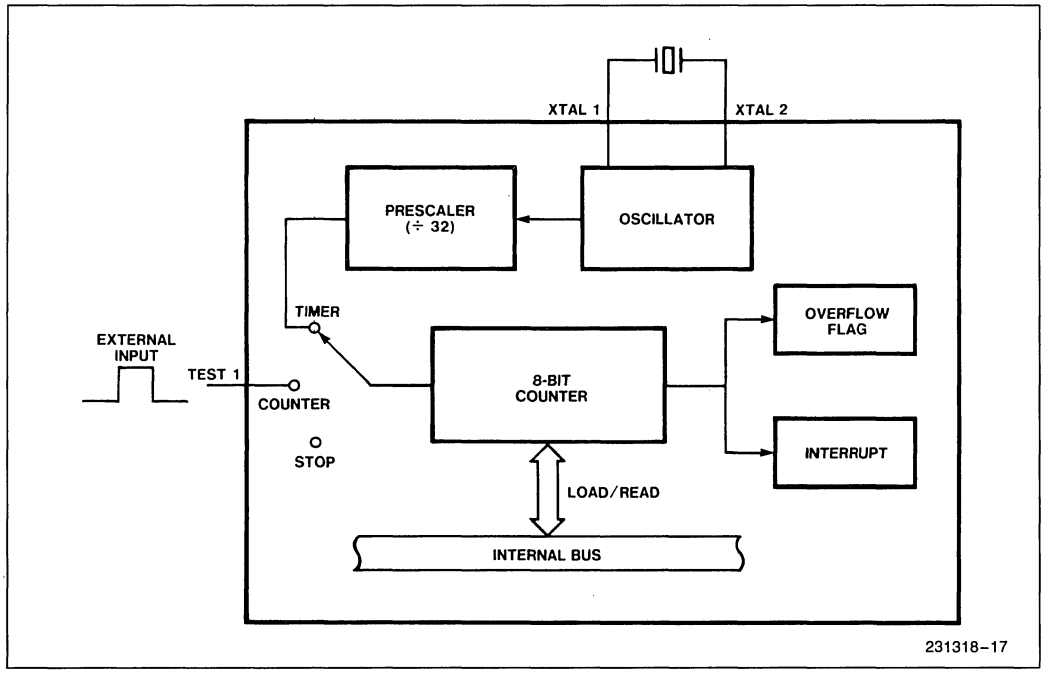


Figure 2-13. Timer Counter

The test imputs are TTL compatible. An external logic signal connected to one of the test inputs will be sampled at the time the appropriate conditional jump instruction is executed. The path of program execution will be altered depending on the state of the external signal when sampled.

INTERRUPTS

The UPI-41A/41AH/42/42AH has the following internal interrupts:

- Input Buffer Full (IBF) interrupt
- Timer Overflow interrupt

The IBF interrupt forces a CALL to location 3 in program memory; a timer-overflow interrupts forces a CALL to location 7. The IBF interrupt is enabled by the EN I instruction and disabled by the DIS I instruction. The timer-overflow interrupt is enabled and disabled by the EN TNCTI and DIS TCNTI instructions, respectively.

Figure 2-14 illustrates the internal interrupt logic. An IBF interrupt request is generated whenever $\overline{WR}$ and $\overline{CS}$ are both low, regardless of whether interrupts are enabled. The interrupt request is cleared upon entering the IBF service routine only. That is, the DIS I instruction does not clear a pending IBF interrupt.

Interrupt Timing Latency

When the IBF interrupt is enabled and an IBF interrupt request occurs, an interrupt sequence is intiated as soon as the currently executing instruction is completed. The following sequence occurs:

- A CALL to location 3 is forced.
- The program counter and bits 4–7 of the Program Status Word are stored in the stack.
- The stack pointer is incremented.

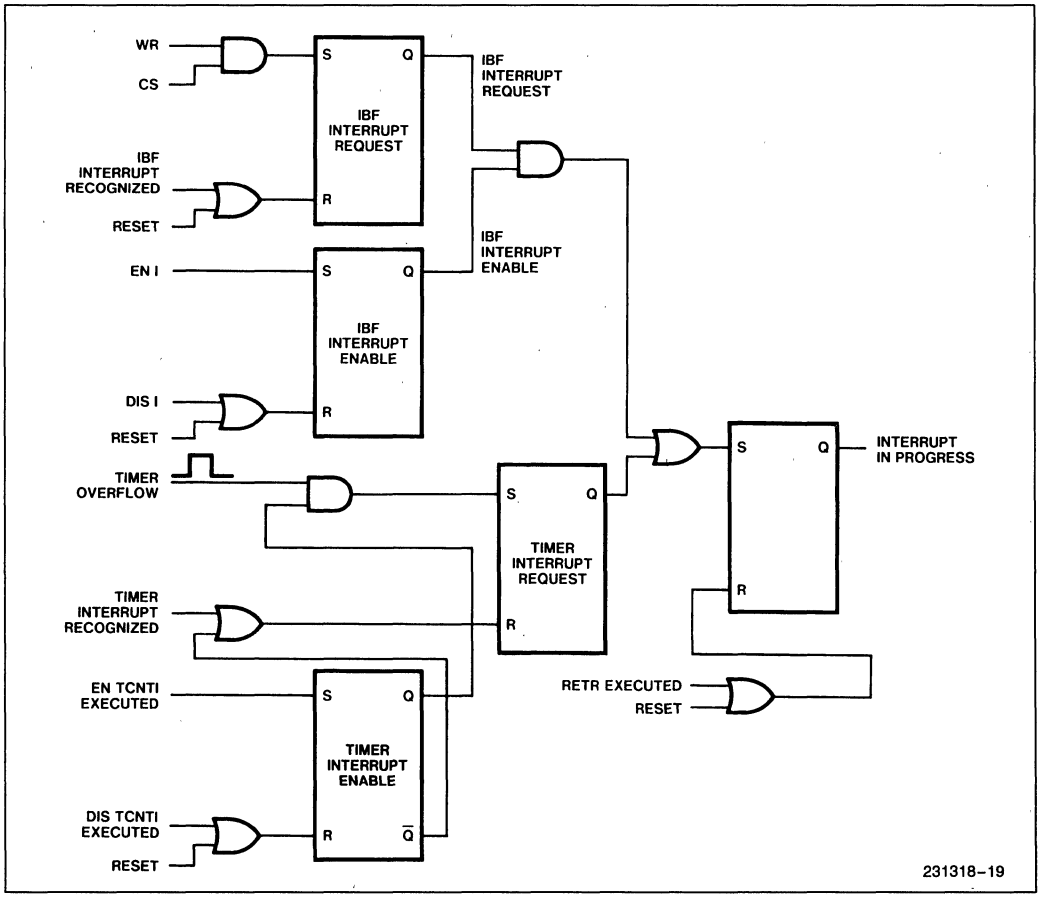


Figure 2-14. Interrupt Logic

Location 3 in program memory should contain an unconditional jump to the beginning of the IBF interrupt service routine elsewhere in program memory. At the end of the service routine, an RETR (Return and Restore Status) instruction is used to return control to the main program. This instruction will restore the program counter and PSW bits 4–7, providing automatic restoration of the previously active register bank as well. RETR also re-enables interrupts.

A timer-overflow interrupt is enabled by the EN TCNTI instruction and disabled by the DIS TCNTI instruction. If enabled, this interrupt occurs when the timer/counter register overflows. A CALL to location 7 is forced and the interrupt routine proceeds as described above.

The interrupt service latency is the sum of current instruction time, interrupt recognition time, and the internal call to the interrupt vector address. The worst case latency time for servicing an interrupt is 7 clock cycles. Best case latency is 4 clock cycles.

Interrupt Timing

Interrupt inputs may be enabled or disabled under program control using EN I, DIS I, EN TCNTI and DIS TCNTI instructions. Also, a $\overline{\text{RESET}}$ input will disable interrupts. An interrupt request must be removed before the RETR instruction is executed to return from the service routine, otherwise the processor will re-enter the service routine immediately. Thus, the $\overline{\text{WR}}$ and $\overline{\text{CS}}$ inputs should not be held low longer than the duration of the interrupt service routine.

The interrupt system is single level. Once an interrupt is detected, all further interrupt requests are latched but are not acted upon until execution of an RETR instruction re-enables the interrupt input logic. This occurs at the beginning of the second cycle of the RETR instruction. If an IBF interrupt and a timer-overflow interrupt occur simultaneously, the IBF interrupt will be recognized first and the timer-overflow interrupt will remain pending until the end of the interrupt service routine.

External Interrupts

An external interrupt can be created using the UPI-41A/41AH/42/42AH timer/counter in the event counter mode. The counter is first preset to FFH and the EN TCNTI instruction is executed. A timer-overflow interrupt is generated by the first high to low transition of the TEST 1 input pin. Also, if an IBF interrupt occurs during servicing of the timer/counter interrupt, it will remain pending until the end of the service routine.

Host Interrupts And DMA

If needed, two external interrupts to the host system can be created using the EN FLAGS instruction. This instruction allocates two I/O lines on PORT 2 (P_{24} and P_{25}). P_{24} is the Output Buffer Full interrupt request line to the host system; P_{25} is the Input Buffer empty interrupt request line. These interrupt outputs reflect the internal status of the OBF flag and the IBF inverted flag. Note, these outputs may be inhibited by writing a "0" to these pins. Reenabling interrupts is done by writing a "1" to these port pins. Interrupts are typically enabled after power on since the I/O ports are set in a "1" condition. The EN FLAG's effect is only cancelled by a device RESET.

DMA handshaking controls are available from two pins on PORT 2 of the UPI-41A/41AH/42/42AH microcomputer. These lines (P_{26} and P_{27}) are enabled by the EN DMA instruction. P_{26} becomes DMA request (DRQ) and P_{27} becomes DMA acknowledge ($\overline{\text{DACK}}$). The UPI program initiates a DMA request by writing a "1" to P_{26}. The DMA controller transfers the data into the DBBIN data register using DACK which acts as a chip select. The EN DMA instruction can only be cancelled by a chip RESET.

RESET

The $\overline{\text{RESET}}$ input provides a means for internal initialization of the processor. An automatic initialization pulse can be generated at power-on by simply connecting a 1 μfd capacitor between the $\overline{\text{RESET}}$ input and ground as shown in Figure 2-15. It has an internal pull-up resistor to charge the capacitor and a Schmitt-trigger circuit to generate a clean transition. A 2-stage synchronizer has been added to support reliable operation up to 12.5 MHz.

If automatic initialization is used, $\overline{\text{RESET}}$ should be held low for at least 10 milliseconds to allow the power supply to stabilize. If an external $\overline{\text{RESET}}$ signal is used, $\overline{\text{RESET}}$ may be held low for a minimum of 8 instruction cycles. Figure 2-15 illustrates a configuration using an external TTL gate to generate the $\overline{\text{RESET}}$ input. This configuration can be used to derive the $\overline{\text{RESET}}$ signal from the 8224 clock generator in an 8080 system.

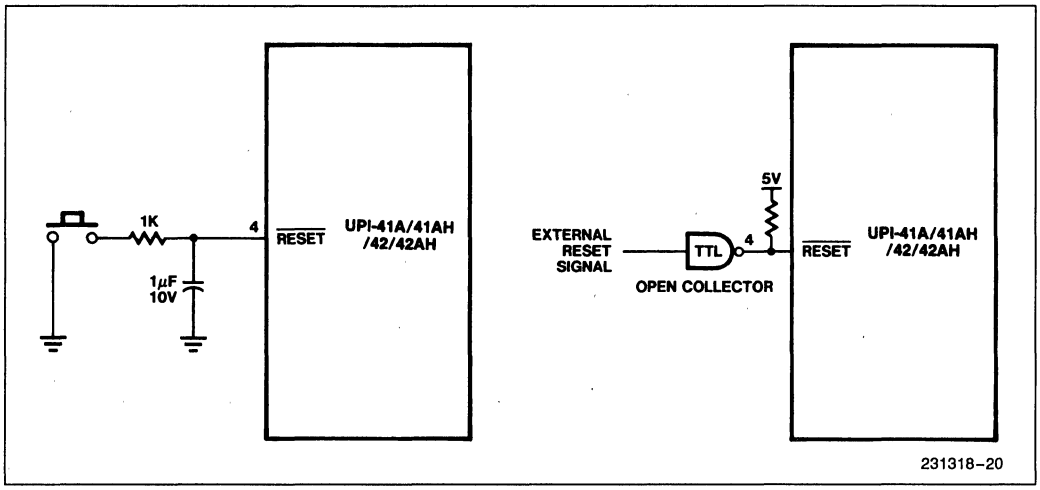


Figure 2-15. External Reset Configuration

The $\overline{\text{RESET}}$ input performs the following functions:

- Disables Interrupts
- Clears Program Counter to Zero
- Clears Stack Pointer
- Clears Status Register and Flags
- Clears Timer and Timer Flag
- Stops Timer
- Selects Register Bank 0
- Sets PORTS 1 and 2 to Input Mode

DATA BUS BUFFER

Two 8-bit data bus buffer registers, DBBIN and DBBOUT, serve as temporary buffers for commands and data flowing between it and the master processor. Externally, data is transmitted or received by the DBB registers upon execution of an INput or OUTput instruction by the master processor. Four control signals are used:

- A_0 Address input signifying control or data
- $\overline{\text{CS}}$ Chip Select
- $\overline{\text{RD}}$ Read Strobe
- $\overline{\text{WR}}$ Write Strobe

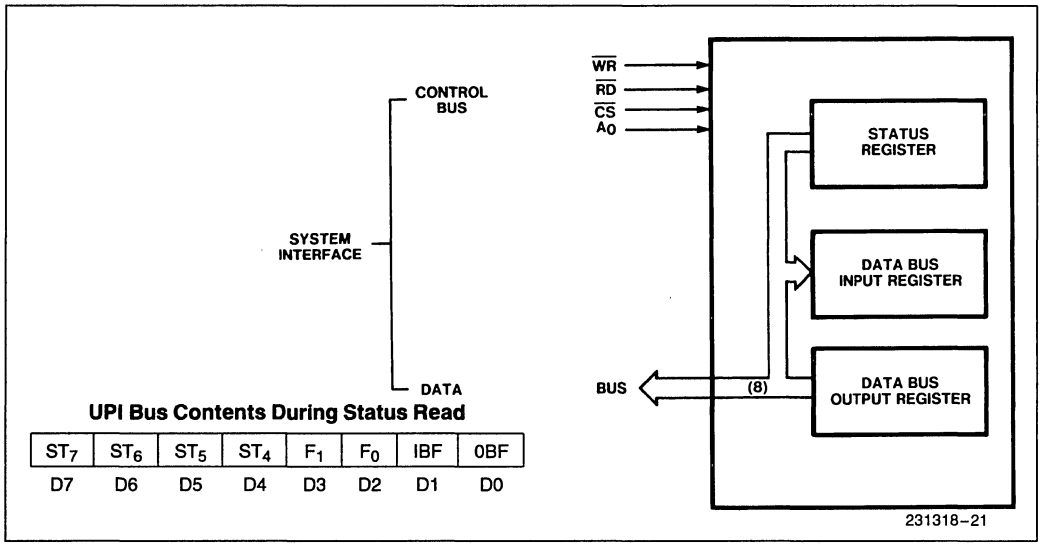
Transfer can be implemented with or without UPI program interference by enabling or disabling an internal UPI interrupt. Internally, data transfer between the DBB and the UPI accumulator is under software control and is completely asynchronous to the external processor timing. This allows the UPI software to handle peripheral control tasks independent of the main processor while still maintaining a data interface with the master system.

Configuration

Figure 2-16 illustrates the internal configuration of the DBB registers. Data is stored in two 8-bit buffer registers, DBBIN and DBBOUT. DBBIN and DBBOUT may be accessed by the external processor using the $\overline{\text{WR}}$ line and the $\overline{\text{RD}}$ line, respectively. The data bus is a bidirectional, three-state bus which can be connected directly to an 8-bit microprocessor system. Four control lines ($\overline{\text{WR}}$, $\overline{\text{RD}}$, $\overline{\text{CS}}$, A_0) are used by the external processor to transfer data to and from the DBBIN and DBBOUT registers.

An 8-bit register containing status flags is used to indicate the status of the DBB registers. The eight status flags are defined as follows:

- **OBF Output Buffer Full**
  This flag is automatically set when the UPI-Microcomputer loads the DBBOUT register and is cleared when the master processor reads the data register.
- **IBF Input Buffer Full**
  This flag is set when the master processor writes a character to the DBBIN register and is cleared when the UPI INputs the data register contents to its accumulator.



UPI Bus Contents During Status Read

ST_7	ST_6	ST_5	ST_4	F_1	F_0	IBF	OBF
D7	D6	D5	D4	D3	D2	D1	D0

Figure 2-16. Data Bus Buffer Configuration

- **F_0**
  This is a general purpose flag which can be cleared or toggled under UPI software control. The flag is used to transfer UPI status information to the master processor.
- **F_1 Command/Data**
  This flag is set to the condition of the A_0 input line when the master processor writes a character to the data register. The F_1 flag can also be cleared or toggled under UPI-Microcomputer program control.
- **ST4 through ST7**
  These bits are user defined status bits. They are defined by the MOV STS,A instruction.

SYSTEM INTERFACE

Figure 2-17 illustrates how a UPI-Microcomputer can be connected to a standard 8080-type bus system. Data lines D_0–D_7 form a three-state, bidirectional port which can be connected directly to the system data bus. The UPI bus interface has sufficient drive capability (400 μA) for small systems, however, a larger system may require buffers.

Four control signals are required to handle the data and status information transfer:

- $\overline{WR}$
  I/O WRITE signal used to transfer data from the system bus to the UPI DBBIN register and set the F_1 flag in the status register.
- $\overline{RD}$
  I/O READ signal used to transfer data from the DBBOUT register or status register to the system data bus.
- $\overline{CS}$
  CHIP SELECT signal used to enable one 8041AH out of several connected to a common bus.
- $\overline{A_0}$
  Address input used to select either the 8-bit status register or DBBOUT register during an I/O READ. Also, the signal is used to set the F_1 flag in the status register during an I/O WRITE.

The $\overline{WR}$ and $\overline{RD}$ signals are active low and are standard MCS-80 peripheral control signals used to synchronize data transfer between the system bus and peripheral devices.

The $\overline{CS}$ and A_0 signals are decoded from the address bus of the master system. In a system with few I/O devices a linear addressing configuration can be used where A_0 and A_1 lines are connected directly to A_0 and $\overline{CS}$ inputs (see Figure 2-17).

Data Read

Table 2-4 illustrates the relative timing of a DBBOUT Read. When $\overline{CS}$, A_0, and $\overline{RD}$ are low, the contents of the DBBOUT register is placed on the three-state Data lines D_0–D_7 and the OBF flag is cleared.

The master processor uses $\overline{CS}$, A_0, $\overline{WR}$, and $\overline{RD}$ to control data transfer between the DBBOUT register and the master system. The following operations are under master processor control:

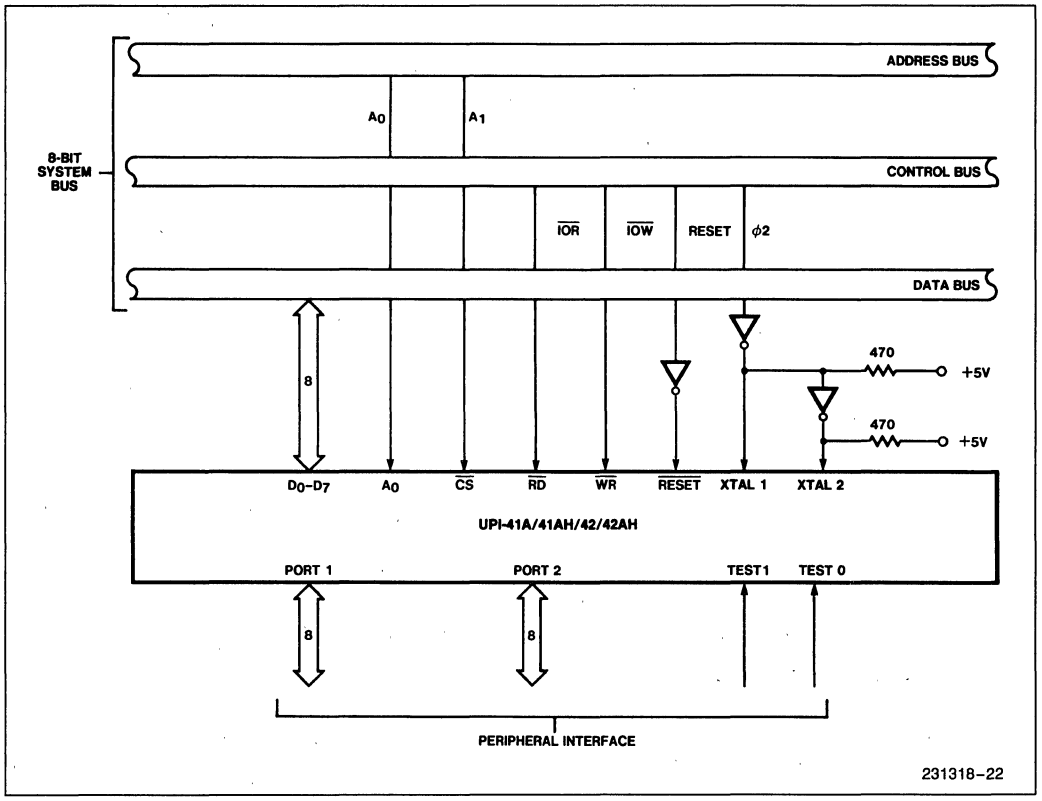


Figure 2-17. Interface to 8080 System Bus

Table 2-4. Data Transfer Controls

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A_0	
0	0	1	0	Read DBBOUT register
0	0	1	1	Read STATUS register
0	1	0	0	Write DBBIN data register
0	1	0	1	Write DBBIN command register
1	x	x	x	Disable DBB

Status Read

Table 2-4 shows the logic sequence required for a STATUS register read. When $\overline{CS}$ and $\overline{RD}$ are low with A_0 high, the contents of the 8-bit status register appears on Data lines D_0–D_7.

Data Write

Table 2-4 shows the sequence for writing information to the DBBIN register. When $\overline{CS}$ and $\overline{WR}$ are low, the contents of the system data bus is latched into DBBIN. Also, the IBF flag is set and an interrupt is generated, if enabled.

Command Write

During any write (Table 2-4), the state of the A_0 input is latched into the status register in the F_1 (command/data) flag location. This additional bit is used to signal whether DBBIN contents are command (A_0 = 1) or data (A_0 = 0) information.

INPUT/OUTPUT INTERFACE

The UPI-41A/41AH/42/42AH has 16 lines for input and output functions. These I/O lines are grouped as two 8-bit TTL compatible ports: PORTS 1 and 2. The port lines can individually function as either inputs or outputs under software control. In addition, the lower 4 lines of PORT 2 can be used to interface to an 8243 I/O expander device to increase I/O capacity to 28 or more lines. The additional lines are grouped as 4-bit ports: PORTS 4, 5, 6, and 7.

PORTS 1 and 2

PORTS 1 and 2 are each 8 bits wide and have the same I/O characteristics. Data written to these ports by an

OUTL Pp,A instruction is latched and remains unchanged until it is rewritten. Input data is sampled at the time the IN, A, Pp instruction is executed. Therefore, input data must be present at the PORT until read by an INput instruction. PORT 1 and 2 inputs are fully TTL compatible and outputs will drive one standard TTL load.

Circuit Configuration

The PORT 1 and 2 lines have a special output structure (shown in Figure 2-18) that allows each line to serve as an input, an output, or both, even though outputs are statically latched.

Each line has a permanent high impedance pull-up (50 KΩ) which is sufficient to provide source current for a TTL high level, yet can be pulled low by a standard TTL gate drive. Whenever a "1" is written to a line, a low impedance pull-up (250Ω) is switched in momentarily (500 ns) to provide a fast transition from 0 to 1. When a "0" is written to the line, a low impedance pull-down (300Ω) is active to provide TTL current sinking capability.

To use a particular PORT pin as an input, a logic "1" must first be written to that pin.

NOTE:
A $\overline{\text{RESET}}$ initializes all PORT pins to the high impedance logic "1" state.

An external TTL device connected to the pin has sufficient current sinking capability to pull-down the pin to the low state. An IN A, Pp instruction will sample the status of PORT pin and will input the proper logic level. With no external input connected, the IN A,Pp instruction inputs the previous output status.

This structure allows input and output information on the same pin and also allows any mix of input and output lines on the same port. However, when inputs and outputs are mixed on one PORT, a PORT write will cause the strong internal pull-ups to turn on at all inputs. If a switch or other low impedance device is connected to an input, a PORT write ("1" to an input) could cause current limits on internal lines to be exceeded. Figure 2-19 illustrates the recommended connection when inputs and outputs are mixed on one PORT.

The bidirectional port structure in combination with the UPI-41A/41AH/42/42AH logical AND and OR instructions provide an efficient means for handling single line inputs and outputs within an 8-bit processor.

PORTS 4, 5, 6, and 7

By using an 8243 I/O expander, 16 additional I/O lines can be connected to the UPI-41AH, 42AH and directly addressed as 4-bit I/O ports using UPI-41AH, 42AH

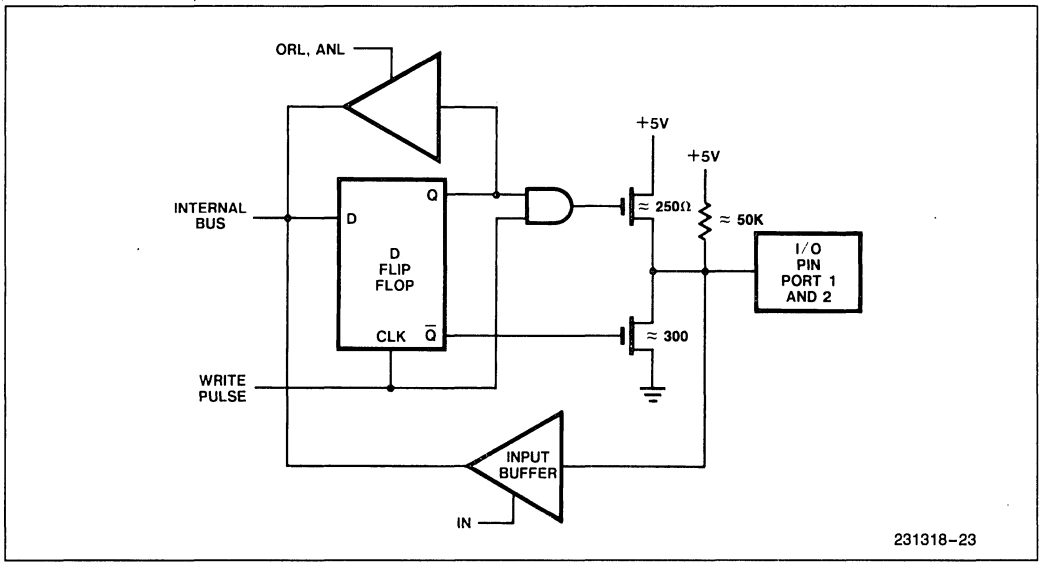


Figure 2-18. Quasi-Bidirectional Port Structure

instructions. This feature saves program space and design time, and improves the bit handling capability of the UPI-41A/41AH/42/42AH.

The lower half of PORT 2 provides an interface to the 8243 as illustrated in Figure 2-20. The PROG pin is used as a strobe to clock address and data information via the PORT 2 interface. The extra 16 I/O lines are referred to in UPI software as PORTS 4, 5, 6, and 7. Each PORT can be directly addressed and can be ANDed and ORed with an immediate data mask. Data can be moved directly to the accumulator from the expander PORTS (or vice-versa).

The 8243 I/O ports, PORTS 4, 5, 6, and 7, provide more drive capability than the UPI-41A/41AH/42/42AH bidirectional ports. The 8243 output is capable of driving about 5 standard TTL loads.

Multiple 8243's can be connected to the PORT 2 interface. In normal operation, only one of the 8243's would be active at the time an Input or Output command is executed. The upper half of PORT 2 is used to provide chip select signals to the 8043's. Figure 2-21 shows how four 8243's could be connected. Software is needed to select and set the proper PORT 2 pin before an INPUT or OUTPUT command to PORTS 4–7 is executed. In general, the software overhead required is very minor compared to the added flexibility of having a large number of I/O pins available.

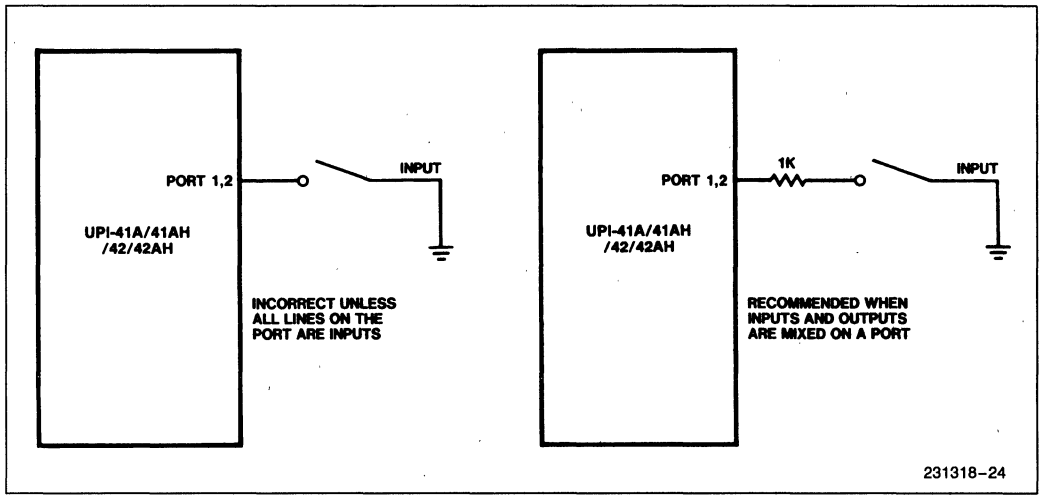


Figure 2-19. Recommended PORT Input Connections

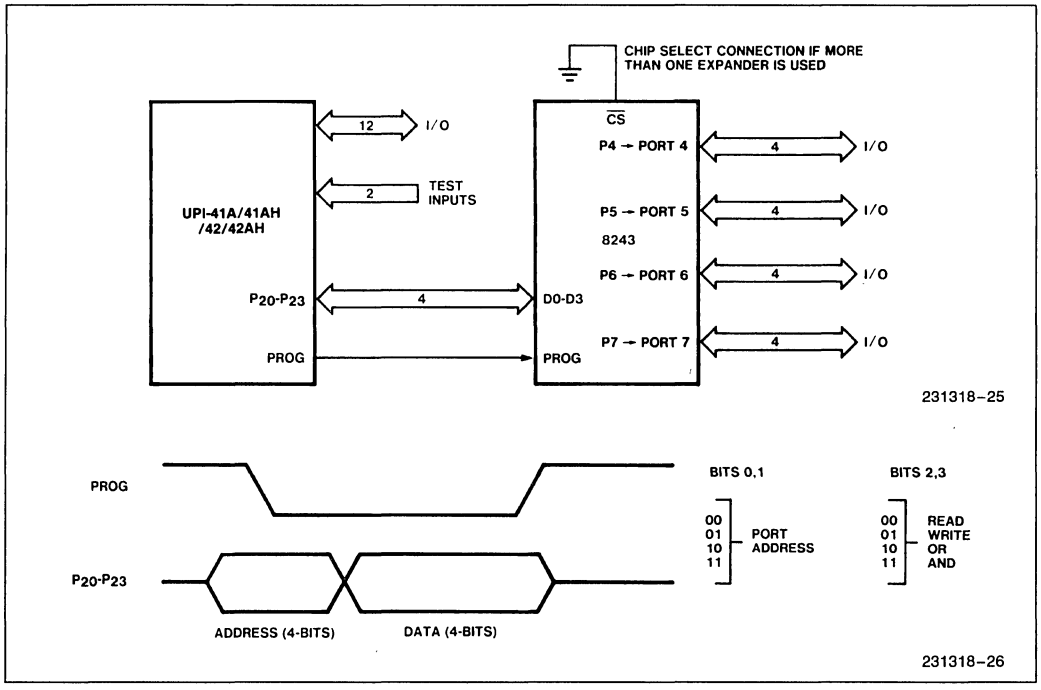


Figure 2-20. 8243 Expander Interface

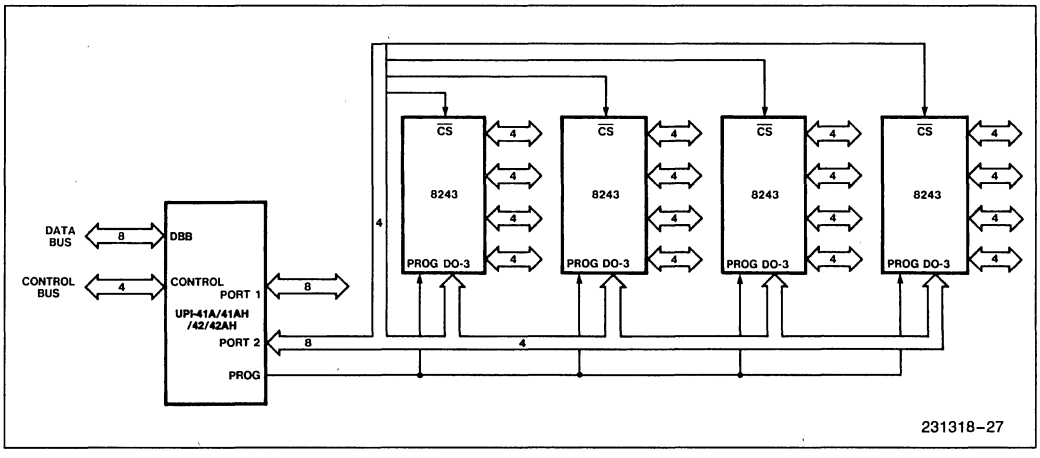


Figure 2-21. Multiple 8243 Expansion

CHAPTER 3
INSTRUCTION SET

The UPI-41A/41AH/42/42AH Instruction Set is op-code-compatible with the MCS-48 set except for the elimination of external program and data memory instructions and the addition of the data bus buffer instructions. It is very straightforward and efficient in its use of program memory. All instructions are either 1 or 2 bytes in length (over 70% are only 1 byte long) and over half of the instructions execute in one machine cycle. The remainder require only two cycles and include Branch, Immediate, and I/O operations.

The UPI-41A/41AH/42/42AH Instruction Set efficiently handles the single-bit operations required in control applications. Special instructions allow port bits to be set or cleared individually. Also, any accumulator bit can be directly tested via conditional branch instructions. Additional instructions are included to simplify loop counters, table look-up routines and N-way branch routines.

The UPI-41A/41AH/42/42AH Microcomputer handles arithmetic operations in both binary and BCD for efficient interface to peripherals such as keyboards and displays.

The instruction set can be divided into the following groups:

- Data Moves
- Accumulator Operations
- Flags
- Register Operations
- Branch Instructions
- Control
- Timer Operations
- Subroutines
- Input/Output Instructions

Data Moves (See Instruction Summary)

The 8-bit accumulator is the control point for all data transfers within the UPI-41A/41AH/42/42AH. Data can be transferred between the 8 registers of each working register bank and the accumulator directly (i.e., with a source or destination register specified by 3 bits in the instruction). The remaining locations in the RAM array are addressed either by R_0 or R_1 of the active register bank. Transfers to and from RAM require one cycle.

Constants stored in Program Memory can be loaded directly into the accumulator or the eight working registers. Data can also be transferred directly between the accumulator and the on-board timer/counter, the Status Register (STS), or the Program Status Word (PSW). Transfers to the STS register alter bits 4–7 only. Transfers to the PSW alter machine status accordingly and provide a means of restoring status after an interrupt or of altering the stack pointer if necessary.

Accumulator Operations

Immediate data, data memory, or the working registers can be added (with or without carry) to the accumulator. These sources can also be ANDed, ORed, or exclusive ORed to the accumulator. Data may be moved to or from the accumulator and working registers or data memory. The two values can also be exchanged in a single operation.

The lower 4 bits of the accumulator can be exchanged with the lower 4 bits of any of the internal RAM locations. This operation, along with an instruction which swaps the upper and lower 4-bit halves of the accumulator, provides easy handling of BCD numbers and other 4-bit quantities. To facilitate BCD arithmetic a Decimal Adjust instruction is also included. This instruction is used to correct the result of the binary addition of two 2-digit BCD numbers. Performing a decimal adjust on the result in the accumulator produces the desired BCD result.

The accumulator can be incremented, decremented, cleared, or complemented and can be rotated left or right 1 bit at a time with or without carry.

A subtract operation can be easily implemented in UPI software using three single-byte, single-cycle instructions. A value can be subtracted from the accumulator by using the following instructions:

- Complement the accumulator
- Add the value to the accumulator
- Complement the accumulator

Flags

There are four user accessible flags:

- Carry
- Auxiliary Carry
- F_0
- F_1

The Carry flag indicates overflow of the accumulator, while the Auxiliary Carry flag indicates overflow between BCD digits and is used during decimal adjust

operations. Both Carry and Auxiliary Carry are part of the Program Status Word (PSW) and are stored in the stack during subroutine calls. The F_0 and F_1 flags are general-purpose flags which can be cleared or complemented by UPI instructions. F_0 is accessible via the Program Status Word and is stored in the stack with the Carry flags. F_1 reflects the condition of the A_0 line, and caution must be used when setting or clearing it.

Register Operations

The working registers can be accessed via the accumulator as explained above, or they can be loaded with immediate data constants from program memory. In addition, they can be incremented or decremented directly, or they can be used as loop counters as explained in the section on branch instructions.

Additional Data Memory locations can be accessed with indirect instructions via R_0 and R_1.

Branch Instructions

The UPI-41A/41AH/42/42AH Instruction Set includes 17 jump instructions. The unconditional allows jumps anywhere in the 1K words of program memory. All other jump instructions are limited to the current page (256 words) of program memory.

Conditional jump instructions can test the following inputs and maching flags:

- TEST 0 input pin
- TEST 1 input pin
- Input Buffer Full flag
- Output Buffer Full flag
- Timer flag
- Accumulator zero
- Accumulator bit
- Carry flag
- F_0 flag
- F_1 flag

The conditions tested by these instructions are the instantaneous values at the time the conditional jump instruction is executed. For instance, the jump on accumulator zero instruction tests the accumulator itself, not an intermediate flag.

The decrement register and jump if not zero (DJNZ) instruction combines decrement and branch operations in a single instruction which is useful in implementing a loop counter. This instruction can designate any of the 8 working registers as a counter and can effect a branch to any address within the current page of execution.

A special indirect jump instruction (JMPP @A) allows the program to be vectored to any one of several different locations based on the contents of the accumulator. The contents of the accumulator point to a location in program memory which contains the jump address. As an example, this instruction could be used to vector to any one of several routines based on an ASCII character which has been loaded into the accumulator. In this way, ASCII inputs can be used to initiate various routines.

Control

The UPI-41A/41AH/42/42AH Instruction Set has six instructions for control of the DMA, interrupts, and selection of working registers banks.

The UPI-41A/41AH/42/42AH provides two instructions for control of the external microcomputer system. IBF and OBF flags can be routed to PORT 2 allowing interrupts of the external processor. DMA handshaking signals can also be enabled using lines from PORT 2.

The IBF interrupt can be enabled and disabled using two instructions. Also, the interrupt is automatically disabled following a RESET input or during an interrupt service routine.

The working register bank switch instructions allow the programmer to immediately substitute a second 8 register bank for the one in use. This effectively provides either 16 working registers or the means for quickly saving the contents of the first 8 registers in response to an interrupt. The user has the option of switching register banks when an interrupt occurs. However, if the banks are switched, the original bank will automatically be restored upon execution of a return and restore status (RETR) instruction at the end of the interrupt service routine.

Timer

The 8-bit on-board timer/counter can be loaded or read via the accumulator while the counter is stopped or while counting.

The counter can be started as a timer with an internal clock source or as an event counter or timer with an

external clock applied to the TEST 1 pin. The instruction executed determines which clock source is used. A single instruction stops the counter whether it is operating with an internal or an external clock source. In addition, two instructions allow the timer interrupt to be enabled or disabled.

Subroutines

Subroutines are entered by executing a call instruction. Calls can be made to any address in the 1K word program memory. Two separate return instructions determine whether or not status (i.e., the upper 4 bits of the PSW) is restored upon return from a subroutine.

Input/Output Instructions

Two 8-bit data bus buffer registers (DBBIN and DBBOUT) and an 8-bit status register (STS) enable the UPI-41A universal peripheral interface to communicate with the external microcomputer system. Data can be INputted from the DBBIN register to the accumulator. Data can be OUTputted from the accumulator to the DBBOUT register.

The STS register contains four user-definable bits (ST_4–ST_7) plus four reserved status bits (IBF, OBF, F_0 and F_1). The user-definable bits are set from the accumulator.

The UPI-41A/41AH/42/42AH peripheral interface has two 8-bit static I/O ports which can be loaded to and from the accumulator. Outputs are statically latched but inputs to the ports are sampled at the time an IN instruction is executed. In addition, immediate data from program memory can be ANDed and ORed directly to PORTS 1 and 2 with the result remaining on the port. This allows "masks" stored in program memory to be used to set or reset individual bits on the I/O ports. PORTS 1 and 2 are configured to allow input on a given pin by first writing a "1" to the pin.

Four additional 4-bit ports are available through the 8243 I/O expander device. The 8243 interfaces to the UPI-41A/41AH/42/42AH peripheral interface via four PORT 2 lines which form an expander bus. The 8243 ports have their own AND and OR instructions like the on-board ports, as well as move instructions to transfer data in or out. The expander AND or OR instructions, however, combine the contents of the accumulator with the selected port rather than with immediate data as is done with the on-board ports.

INSTRUCTION SET DESCRIPTION

The following section provides a detailed description of each UPI instruction and illustrates how the instructions are used.

For further information about programming the UPI, consult the *8048/8041AH Assembly Language Manual.*

Table 3-1. Symbols and Abbreviations Used

Symbol	Definition
A	Accumulator
C	Carry
DBBIN	Data Bus Buffer Input
DBBOUT	Data Bus Buffer Output
F_0, F_1	FLAG 0, FLAG 1 (C/D flag)
I	Interrupt
P	Mnemonic for "in-page" operation
PC	Program Counter
Pp	Port designator (p = 1, 2, or 4–7)
PSW	Program Status Word
Rr	Register designator (r = 0–7)
SP	Stack Pointer
STS	Status register
T	Timer
TF	Timer Flag
T_0, T_1	TEST 0, TEST 1
#	Immediate data prefix
@	Indirect address prefix
(())	Double parentheses show the effect of @, that is @RO is shown as ((RO)).
()	Contents of

Table 3-2. Instruction Set Summary

Mnemonic	Description	Bytes	Cycle
ACCUMULATOR			
ADD A, Rr	Add register to A	1	1
ADD A, @Rr	Add data memory to A	1	1
ADD A, #data	Add immediate to A	2	2
ADDC A, Rr	Add register to A with carry	1	1
ADDC A, @Rr	Add data memory to A with carry	1	1
ADDC A, #data	Add immediate to A with carry	2	2
ANL A, Rr	And register to A	1	1
ANL A, @Rr	And data memory to A	1	1
ANL A, #data	And immediate to A	2	2
ORL A, Rr	Or register to A	1	1
ORL A, @Rr	Or data memory to A	1	1
ORL A, #data	Or immediate to A	2	2
XRL A, Rr	Exclusive Or register to A	1	1
XRL A, @Rr	Exclusive Or data memory to A	1	1
XRL A, #data	Exclusive Or immediate to A	2	2
INC A	Increment A	1	1
DEC A	Decrement A	1	1
CLR A	Clear A	1	1
CPL A	Complement A	1	1
DA A	Decimal Adjust A	1	1
SWAP A	Swap nibbles of A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through carry	1	1
INPUT/OUTPUT			
IN A, Pp	Input port to A	1	2
OUTL Pp, A	Output A to port	1	2
ANL Pp, #data	And immediate to port	2	2
ORL Pp, #data	Or immediate to port	2	2
IN A,DBB	Input DDB to A, clear IBF	1	1
OUT DBB, A	Output A to DBB, Set OBF	1	1
MOV STS,A	A_4–A_7 to bits 4–7 of status	1	1
MOVD A,Pp	Input Expander port to A	1	2
MOVD Pp,A	Output A to Expander port	1	2
ANLD Pp,A	And A to Expander port	1	2
ORLD Pp,A	Or A to Expander port	1	2
DATA MOVES			
MOV A, Rr	Move register to A	1	1
MOV A, @Rr	Move data memory to A	1	1
MOV A, #data	Move immediate to A	2	2
MOV Rr, A	Move A to register	1	1
MOV @Rr, A	Move A to data memory	1	1
MOV Rr, #data	Move immediate to register	2	2
MOV @Rr, #data	Move immediate to data memory	2	2
MOV A, PSW	Move PSW to A	1	1
MOV PSW, A	Move A to PSW	1	1
XCH A, Rr	Exchange A and registers	1	1
XCH A, @Rr	Exchange A and data memory	1	1
XCHD A, @Rr	Exchange digit of A and register	1	1

Mnemonic	Description	Bytes	Cycle
DATA MOVES (Continued)			
MOVP A, @A	Move to A from current page	1	2
MOVP3 A, @A	Move to A from page 3	1	2
TIMER/COUNTER			
MOV A,T	Read Timer/Counter	1	1
MOV T,A	Load Timer/Counter	1	1
STRT T	Start Timer	1	1
STRT CNT	Start Counter	1	1
STOP TCNT	Stop Timer/Counter	1	1
EN TCNTI	Enable Timer/Counter Interrupt	1	1
DIS TCNTI	Disable Timer/Counter Interrupt	1	1
CONTROL			
EN DMA	Enable DMA Handshake Lines	1	1
EN I	Enable IBF interrupt	1	1
DIS I	Disable IBF interrupt	1	1
EN FLAGS	Enable Master Interrupts	1	1
SEL RB0	Select register bank 0	1	1
SEL RB1	Select register bank 1	1	1
NOP	No Operation	1	1
REGISTERS			
INC Rr	Increment register	1	1
INC @Rr	Increment data memory	1	1
DEC Rr	Decrement register	1	1
SUBROUTINE			
CALL addr	Jump to subroutine	2	2
RET	Return	1	2
RETR	Return and restore status	1	2
FLAGS			
CLR C	Clear Carry	1	1
CPL C	Complement Carry	1	1
CLR F0	Clear Flag 0	1	1
CPL F0	Complement Flag 0	1	1
CLR F1	Clear F_1 Flag	1	1
CPL F1	Complement F_1 Flag	1	1
BRANCH			
JMP addr	Jump unconditional	2	2
JMPP @A	Jump indirect	1	2
DJNZ Rr, addr	Decrement register and jump on non-zero	2	2
JC addr	Jump on Carry = 1	2	2
JNC addr	Jump on Carry = 0	2	2
JZ addr	Jump on A zero	2	2
JNZ addr	Jump on A not zero	2	2
JT0 addr	Jump on T_0 = 1	2	2
JNT0 addr	Jump on T_0 = 0	2	2
JT1 addr	Jump on T_1 = 1	2	2
JNT1 addr	Jump on T_1 = 0	2	2
JF0 addr	Jump on F_0 Flag = 1	2	2
JF1 addr	Jump on F_1 Flag = 1	2	2
JTF addr	Jump on Timer Flag = 1	2	2
JNIBF addr	Jump on IBF Flag = 0	2	2
JOBF addr	Jump on OBF Flag = 1	2	2
JBb addr	Jump on Accumulator Bit	2	2

ALPHABETIC LISTING

ADD A,Rr Add Register Contents to Accumulator

Opcode:

0 1 1 0	1 r_2 r_1 r_0

The contents of register 'r' are added to the accumulator. Carry is affected.
(A) ← (A) + (Rr) r = 0–7

Example:
```
ADDREG: ADD A,R6        ;ADD REG 6 CONTENTS
                        ;TO ACC
```

ADD A,@Rr Add Data Memory Contents to Accumulator

Opcode:

0 1 1 0	0 0 0 r

The contents of the standard data memory location address by register 'r' bits 0–7 are added to the accumulator. Carry is affected.

(A) ← (A) + ((Rr)) r = 0–1

Example:
```
ADDM: MOV RO,#47        ;MOVE 47 DECIMAL TO REG 0
      ADD A,@RO         ;ADD VALUE OF LOCATION
                        ;47 TO ACC
```

ADD A,#data Add Immediate Data to Accumulator

Opcode:

0 0 0 0	0 0 1 1	•	d_7 d_6 d_5 d_4	d_3 d_2 d_1 d_0

This is a 2-cycle instruction. The specified data is added to the accumulator. Carry is affected.
(A) ← (A) + data

Example:
```
ADDID: ADD A,#ADDER     ;ADD VALUE OF SYMBOL
                        ;ADDER' TO ACC
```

ADDC A,Rr Add Carry and Register Contents to Accumulator

Opcode:

0 1 1 1	1 r_2 r_1 r_0

The content of the carry bit is added to accumulator location 0. The contents of register 'r' are then added to the accumulator. Carry is affected.
(A) ← (A) + (Rr) + (C) r = 0–7

Example:
```
ADDRGC: ADDC A,R4       ;ADD CARRY AND REG 4
                        ;CONTENTS TO ACC
```

ADDC A,@Rr Add Carry and Data Memory Contents to Accumulator

Opcode:

0 1 1 1	0 0 0 r

The content of the carry bit is added to accumulator location 0. Then the contents of the standard data memory location addressed by register 'r' bits 0–7 are added to the accumulator. Carry is affected.

(A) ← (A) + ((Rr)) + (C) r = 0–1

Example:

```
ADDMC: MOV R1,#40      ;MOV '40' DEC TO REG 1
       ADDC A,@R1      ;ADD CARRY AND LOCATION 40
                       ;CONTENTS TO ACC
```

ADDC A,#data Add Carry and Immediate Data to Accumulator

Opcode:

0 0 0 1	0 0 1 1	•	d_7 d_6 d_5 d_4	d_3 d_2 d_1 d_0

This is a 2-cycle instruction. The content of the carry bit is added to accumulator location 0. Then the specified data is added to the accumulator. Carry is affected.

(A) ← (A) + data + (C)

Example:

```
ADDC A,#255            ;ADD CARRY AND '225' DEC
                       ;TO ACC
```

ANL A,Rr Logical AND Accumulator With Register Mask

Opcode:

0 1 0 1	1 r_2 r_1 r_0

Data in the accumulator is logically ANDed with the mask contained in working register 'r'.

(A) ← (A) AND (Rr) r = 0–7

Example:

```
ANDREG: ANL A,R3       ;'AND' ACC CONTENTS WITH MASK
                       ;MASK IN REG 3
```

ANL A,@Rr Logical AND Accumulator With Memory Mask

Opcode:

0 1 0 1	0 0 0 r

Data in the accumulator is logically ANDed with the mask contained in the data memory location referenced by register 'r', bits 0–7.

(A) ← (A) AND ((Rr)) r = 0–1

Example:

```
ANDDM: MOV R0,#0FFH    MOV 'FF' HEX TO REG 0
       ANL A,#0AFH     ;'AND' ACC CONTENTS WITH
                       ;MASK IN LOCATION 63
```

ANL A,#data Logical AND Accumulator With Immediate Mask

Opcode:

0 1 0 1	0 0 1 1	•	d_7 d_6 d_5 d_4	d_3 d_2 d_1 d_0

This is a 2-cycle instruction. Data in the accumulator is logically ANDed with an immediately-specified mask.
(A) ← (A) AND data

Example:

```
ANDID: ANL A,#0AFH          ;'AND' ACC CONTENTS
                            ;WITH MASK 10101111
       ANL A,#3+X/Y         ;'AND' ACC CONTENTS
                            ;WITH VALUE OF EXP
                            '3+X/Y'
```

ANL PP,#data Logical AND PORT 1–2 With Immediate Mask

Opcode:

1 0 0 1	1 0 p_1 p_0	•	d_7 d_6 d_5 d_4	d_3 d_2 d_1 d_0

This is a 2-cycle instruction. Data on the port 'p' is logically ANDed with an immediately-specified mask.
(Pp) ← (Pp) AND data p = 1–2

Note: Bits 0–1 of the opcode are used to represent PORT 1 and PORT 2. If you are coding in binary rather than assembly language, the mapping is as follows:

Bits	p1	p0	Port
	0	0	X
	0	1	1
	1	0	2
	1	1	X

Example:

```
ANDP2: ANL P2,#OF0H         ;'AND' PORT 2 CONTENTS
                            ;WITH MASK'F0' HEX
                            ;(CLEAR P20–23)
```

ANLD Pp,A Logical AND Port 4–7 With Accumulator Mask

Opcode:

1 0 0 1	1 1 p_1 p_0

This is a 2-cycle instruction. Data on port 'p' on the 8243 expander is logically ANDed with the digit mask contained in accumulator bits 0–3.
(Pp) ← (Pp) AND (A0–3) p = 4–7

Note: The mapping of Port 'p' to opcode bits p_1, p_0 is as follows:

P1	P0	Port
0	0	4
0	1	5
1	0	6
1	1	7

Example:

```
ANDP4: ANLD P4,A            ;'AND' PORT 4 CONTENTS
                            ;WITH ACC BITS 0–3
```

CALL address Subroutine Call

Opcode:

a_{10} a_9 a_8 1	0 1 0 0	•	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. The program counter and PSW bits 4–7 are saved in the stack. The stack pointer (PSW bits 0–2) is updated. Program control is then passed to the location specified by 'address'.

Execution continues at the instruction following the CALL upon return from the subroutine.
((SP)) ← (PC), (PSW_{4-7})
(SP) ← (SP) + 1
(PC_{8-9}) ← ($addr_{8-9}$)
(PC_{0-7}) ← ($addr_{0-7}$)

Example: Add three groups of two numbers. Put subtotals in locations 50, 51 and total in location 52.

```
          MOV R0,#50          ;MOVE '50' DEC TO ADDRESS
                              ;REG 0
BEGADD:   MOV A,R1            ;MOVE CONTENTS OF REG 1
                              ;TO ACC
          ADD A,R2            ;ADD REG 2 TO ACC
          CALL SUBTOT         ;CALL SUBROUTINE 'SUBTOT'
          ADD A,R3            ;ADD REG 3 TO ACC
          ADD A,R4            ;ADD REG 4 TO ACC
          CALL SUBTOT         ;CALL SUBROUTINE 'SUBTOT'
          ADD A,R5            ;ADD REG 5 TO ACC
          ADD A,R6            ;ADD REG 6 TO ACC
          CALL SUBTOT         ;CALL SUBROUTINE 'SUBTOT'
                .
                .
                .
SUBTOT:   MOV @R0,A           ;MOVE CONTENTS OF ACC TO
                              ;LOCATION ADDRESSED BY
                              ;REG 0
          INC R0              ;INCREMENT REG 0
          RET                 ;RETURN TO MAIN PROGRAM
```

CLR A Clear Accumulator

Opcode:

0 0 1 0	0 1 1 1

The contents of the accumulator are cleared to zero.
(A) ← 00H

CLR C Clear Carry Bit

Opcode:

1 0 0 1	0 1 1 1

During normal program execution, the carry bit can be set to one by the ADD, ADDC, RLC, CPLC, RRC, and DAA instructions. This instruction resets the carry bit to zero.
(C) ← 0

CLR F1 Clear Flag 1

Opcode:

1 0 1 0	0 1 0 1

The F_1 flag is cleared to zero.
(F_1) ← 0

CLR F0 Clear Flag 0

Opcode:

1 0 0 0	0 1 0 1

F_0 flag is cleared to zero.
$(F_0) \leftarrow 0$

CPL A Complement Accumulator

Opcode:

0 0 1 1	0 1 1 1

The contents of the accumulator are complemented. This is strictly a one's complement. Each one is changed to zero and vice-versa.
(A) ← NOT (A)

Example: Assume accumulator contains 01101010.

```
CPLA: CPL A                    ;ACC CONTENTS ARE COMPLE-
                               ;MENTED TO 10010101
```

CPL C Complement Carry Bit

Opcode:

1 0 1 0	0 1 1 1

The setting of the carry bit is complemented; one is changed to zero, and zero is changed to one.
(C) ← NOT (C)

Example: Set C to one; current setting is unknown.

```
CT01: CLR C                    ;C IS CLEARED TO ZERO
      CPL C                    ;C IS SET TO ONE
```

CPL F0 COMPLEMENT FLAG 0

Opcode:

1 0 0 1	0 1 0 1

The setting of Flag 0 is complemented; one is changed to zero, and zero is changed to one.
$F_0 \leftarrow$ NOT (F0)

CPL F1 Complement Flag 1

Opcode:

1 0 1 1	0 1 0 1

The setting of the F_1 Flag is complemented; one is changed to zero, and zero is changed to one.
$(F_1) \leftarrow$ NOT (F_1)

DA A Decimal Adjust Accumulator

Opcode:

0 1 0 1	0 1 1 1

The 8-bit accumulator value is adjusted to form two 4-bit Binary Coded Decimal (BCD) digits following the binary addition of BCD numbers. The carry bit C is affected. If the contents of bits 0–3 are greater than nine, or if AC is one, the accumulator is incremented by six.

The four high-order bits are then checked. If bits 4–7 exceed nine, or if C is one, these bits are increased by six. If an overflow occurs, C is set to one; otherwise, it is cleared to zero.

Example: Assume accumulator contains 9AH.

```
DA A                    ;ACC ADJUSTED TO 01H with C set
C   AC                  ACC
0   0                   9AH   INITIAL CONTENTS
                        06H   ADD SIX TO LOW DIGIT
0   0                   A1H
                        60H   ADD SIX TO HIGH DIGIT

1   0                   01H   RESULT
```

DEC A Decrement Accumulator

Opcode:

0 0 0 0	0 1 1 1

The contents of the accumulator are decremented by one.
(A) ← (A) − 1

Example: Decrement contents of data memory location 63.

```
MOV R0,#3FH         ;MOVE '3F' HEX TO REG 0
MOV A,@R0           ;MOVE CONTENTS OF LOCATION 63
                    ;TO ACC
DEC A               ;DECREMENT ACC
MOV @R0,A           ;MOVE CONTENTS OF ACC TO
                    ;LOCATION 63
```

DEC Rr Decrement Register

Opcode:

1 1 0 0	1 r_2 r_1 r_0

The contents of working register 'r' are decremented by one.
(Rr) ← (Rr) − 1 r = 0–7

Example:
```
DECR1: DEC R1       ;DECREMENT ADDRESS REG 1
```

DIS I Disable IBF Interrupt

Opcode:

0 0 0 1	0 1 0 1

The input Buffer Full interrupt is disabled. The interrupt sequence is not initiated by $\overline{WR}$ and $\overline{CS}$, however, an IBF interrupt request is latched and remains pending until an EN I (enable IBF interrupt) instruction is executed.

Note: The IBF flag is set and cleared independent of the IBF interrupt request so that handshaking protocol can continue normally.

DIS TCNTI Disable Timer/Counter Interrupt

Opcode:

0 0 1 1	0 1 0 1

The timer/counter interrupt is disabled. Any pending timer interrupt request is cleared. The interrupt sequence is not initiated by an overflow, but the timer flag is set and time accumulation continues.

DJNZ Rr, address Decrement Register and Test

Opcode:

1 1 1 0	1 r_2 r_1 r_0	•	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Register 'r' is decremented and tested for zero. If the register contains all zeros, program control falls through to the next instruction. If the register contents are not zero, control jumps to the specified address within the current page.
(Rr) ← (Rr) − 1
If R ≠ 0, then;
(PC_{0-7}) ← addr

Note: A 10-bit address specification does not cause an error if the DJNZ instruction and the jump target are on the same page. If the DJNZ instruction begins in location 255 of a page, it will jump to a target address on the following page. Otherwise, it is limited to a jump within the current page.

Example: Increment values in data memory locations 50–54.

```
    MOV R0,#50          ;MOVE '50' DEC TO ADDRESS
                        ;REG 0
    MOV R3,#05          ;MOVE '5' DEC TO COUNTER
                        ;REG 3
INCRT: INC @R0          ;INCREMENT CONTENTS OF
                        ;LOCATION ADDRESSED BY
                        ;REG 0
    INC R0              ;INCREMENT ADDRESS IN REG 0
    DJNZ R3,INCRT       ;DECREMENT REG 3——JUMP TO
                        ;'INCRT' IF REG 3 NONZERO
    NEXT——              ;'NEXT' ROUTINE EXECUTED
                        ;IF R3 IS ZERO
```

EN DMA Enable DMA Handshake Lines

Opcode:

1 1 1 0	0 1 0 1

DMA handshaking is enabled using P_{26} as DMA request (DRQ) and P_{27} as DMA acknowledge ($\overline{DACK}$). The $\overline{DACK}$ lines forces $\overline{CS}$ and A_0 low internally and clears DRQ.

EN FLAGS Enable Master Interrupts

Opcode:

1 1 1 1	0 1 0 1

The Output Buffer Full (OBF) and the Input Buffer Full (IBF) flags (IBF is inverted) are routed to P_{24} and P_{25}. For proper operation, a "1" should be written to P_{25} and P_{24} before the EN FLAGS instruction. A "0" written to P_{24} or P_{25} disables the pin.

EN I Enable IBF Interrupt

Opcode:

0 0 0 0	0 1 0 1

The Input Buffer Full interrupt is enabled. A low signal on $\overline{WR}$ and $\overline{CS}$ initiates the interrupt sequence.

EN TCNTI Enable Timer/Counter Interrupt

Opcode:

0 0 1 0	0 1 0 1

The timer/counter interrupt is enabled. An overflow of this register initiates the interrupt sequence.

IN A,DBB Input Data Bus Buffer Contents to Accumulator

Opcode:

0 0 1 0	0 0 1 0

Data in the DBBIN register is transferred to the accumulator and the Input Buffer Full (IBF) flag is set to zero.
(A) ← (DBB)
(IBF) ← 0

Example:

```
INDBB: IN A,DBB          ;INPUT DBBIN CONTENTS TO
                         ;ACCUMULATOR
```

IN A,Pp Input Port 1–2 Data to Accumulator

Opcode:

0 0 0 0	1 0 p_1 p_0

This is a 2-cycle instruction. Data present on port 'p' is transferred (read) to the accumulator.
(A) ← (Pp) p = 1–2 (see ANL instruction)

Example:

```
INP 12: IN A,P1          ;INPUT PORT 1 CONTENTS
                         ;TO ACC
        MOV R6,A         ;MOVE ACC CONTENTS TO
                         ;REG 6
        IN A,P2          ;INPUT PORT 2 CONTENTS
                         ;TO ACC
        MOV R7,A         ;MOVE ACC CONTENTS TO REG 7
```

INC A Increment Accumulator

Opcode:

0 0 0 1	0 1 1 1

The contents of the accumulator are incremented by one.
(A) ← (A) + 1

Example: Increment contents of location 10 in data memory.

```
INCA: MOV R0,#10         ;MOV '10' DEC TO ADDRESS
                         ;REG 0
      MOV A,@R0          ;MOVE CONTENTS OF LOCATION
                         ;10 TO ACC
      INC A              ;INCREMENT ACC
      MOV @R0,A          ;MOVE ACC CONTENTS TO
                         ;LOCATION 10
```

INC Rr Increment Register

Opcode: | 0 0 0 1 | 1 r_2 r_1 r_0 |

The contents of working register 'r' are incremented by one.
(Rr) ← (Rr) + 1 r = 0–7

Example:
```
INCR0: INC R0          ;INCREMENT ADDRESS REG 0
```

INC @Rr Increment Data Memory Location

Opcode: | 0 0 0 1 | 0 0 0 r |

The contents of the resident data memory location addressed by register 'r' bits 0–7 are incremented by one.
((Rr)) ← ((Rr)) + 1 r = 0–1

Example:
```
INCDM: MOV R1,#OFFH    ;MOVE ONES TO REG 1
       INC @R1         ;INCREMENT LOCATION 63
```

JBb address Jump If Accumulator Bit is Set

Opcode: | b_2 b_1 b_0 1 | 0 0 1 0 | • | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 |

This is a 2-cycle instruction. Control passes to the specified address if accumulator bit 'b' is set to one.
(PC_{0-7}) addr if b = 1
(PC) ← (PC) + 2 if b = 0

Example:
```
JB4IS1: JB4 NEXT       ;JUMP TO 'NEXT' ROUTINE
                       ;IF ACC BIT 4 = 1
```

JC address Jump If Carry Is Set

Opcode: | 1 1 1 1 | 0 1 1 0 | • | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 |

This is a 2-cycle instruction. Control passes to the specified address if the carry bit is set to one.
(PC_{0-7}) ← addr if C = 1
(PC) ← (PC) + 2 if C = 0

Example:
```
JC1: JC OVERFLOW       ;JUMP TO 'OVFLOW' ROUTINE
                       ;IF C = 1
```

JF0 address Jump If Flag 0 is Set

Opcode: | 1 0 1 1 | 0 1 1 0 | • | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 |

This is a 2-cycle instruction. Control passes to the specified address if flag 0 is set to one.
(PC_{0-7}) ← addr if F_0 = 1

Example:
```
JF0IS1: JF0 TOTAL      ;JUMP TO 'TOTAL' ROUTINE
                       ;IF F0 = 1
```

JF1 address Jump If C/D Flag (F1) Is Set

Opcode: | 0 1 1 1 | 0 1 1 0 | • | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 |

This is a 2-cycle instruction. Control passes to the specified address if the C/D flag (F_1) is set to one.

$(PC_{0-7}) \leftarrow$ addr if $F_1 = 1$

Example:
```
JF 1IS1: JF1 FILBUF     ;JUMP TO 'FILBUF'
                        ;ROUTINE IF F1 = 1
```

JMP address Direct Jump Within 1K Block

Opcode: | a_{10} a_9 a_8 0 | 0 1 0 0 | • | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 |

This is a 2-cycle instruction. Bits 0–10 of the program counter are replaced with the directly-specified address.

$(PC_{8-10}) \leftarrow$ addr 8–10

$(PC_{0-7}) \leftarrow$ addr 0–7

Example:
```
JMP SUBTOT              ;JUMP TO SUBROUTINE 'SUBTOT'
JMP $-6                 ;JUMP TO INSTRUCTION SIX LOCATIONS
                        ;BEFORE CURRENT LOCATION
JMP 2FH                 ;JUMP TO ADDRESS '2F' HEX
```

JMPP @A Indirect Jump Within Page

Opcode: | 1 0 1 1 | 0 0 1 1 |

This is a 2-cycle instruction. The contents of the program memory location pointed to by the accumulator are substituted for the 'page' portion of the program counter (PC 0–7).

$(PC_{0-7}) \leftarrow ((A))$

Example:
```
Assume accumulator contains OFH
JMPPAG: JMPP @A         ;JMP TO ADDRESS STORED IN
                        ;LOCATION 15 IN CURRENT PAGE
```

JNC address Jump If Carry Is Not Set

Opcode: | 1 1 1 0 | 0 1 1 0 | • | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 |

This is a 2-cycle instruction. Control passes to the specified address if the carry bit is not set, that is, equals zero.

$(PC_{0-7}) \leftarrow$ addr if $C = 0$

Example:
```
JC0: JNC NOVFLO         ;JUMP TO 'NOVFLO' ROUTINE
                        ;IF C = 0
```

JNIBF address Jump If Input Buffer Full Flag Is Low

Opcode: | 1 1 0 1 | 0 1 1 0 | • | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 |

This is a 2-cycle instruction. Control passes to the specified address if the Input Buffer Full flag is low (IBF = 0).

$(PC_{0-7}) \leftarrow$ addr if IBF = 0

Example:
```
LOC 3:JNIBF LOC 3       ;JUMP TO SELF IF IBF = 0
                        ;OTHERWISE CONTINUE
```

JNTO address Jump if TEST 0 is Low

Opcode: | 0 0 1 0 | 0 1 1 0 | • | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 |

This is a 2-cycle instruction. Control passes to the specified address, if the TEST 0 signal is low. Pin is sampled during SYNC.

$(PC_{0-7}) \leftarrow$ addr if $T_0 = 0$

Example:
```
JT0LOW: JNT0 60      ;JUMP TO LOCATION 60 DEC
                     ;IF T0 = 0
```

JNT1 address Jump If TEST 1 is Low

Opcode: | 0 1 0 0 | 0 1 1 0 | • | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 |

This is a 2-cycle instruction. Control passes to the specified address if the TEST 1 signal is low. Pin is sampled during SYNC.

$(PC_{0-7}) \leftarrow$ addr if $T_1 = 0$

Example:
```
JT1LOW: JNT1 OBBH    ;JUMP TO LOCATION 'BB' HEX
                     ;IF T1 = 0
```

JNZ address Jump If Accumulator Is Not Zero

Opcode: | 1 0 0 1 | 0 1 1 0 | • | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 |

This is a 2-cycle instruction. Control passes to the specified address if the accumulator contents are nonzero at the time this instruction is executed.

$(PC_{0-7}) \leftarrow$ addr if $A \neq 0$

Example:
```
JACCNO: JNZ OABH     ;JUMP TO LOCATION 'AB' HEX
                     ;IF ACC VALUE IS NONZERO
```

JOBF Address Jump If Output Buffer Full Flag Is Set

Opcode: | 1 0 0 0 | 0 1 1 0 | • | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 |

This is a 2-cycle instruction. Control passes to the specified address if the Output Buffer Full (OBF) flag is set (= 1) at the time this instruction is executed.

$(PC_{0-7}) \leftarrow$ addr if OBF = 1

Example:
```
JOBFHI: JOBF OAAH    ;JUMP TO LOCATION 'AA' HEX
                     ;IF OBF = 1
```

JTF address Jump If Timer Flag is Set

Opcode: | 0 0 0 1 | 0 1 1 0 | • | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 |

This is a 2-cycle instruction. Control passes to the specified address if the timer flag is set to one, that is, the timer/counter register overflows to zero. The timer flag is cleared upon execution of this instruction. (This overflow initiates an interrupt service sequence if the timer-overflow interrupt is enabled.)

$(PC_{0-7}) \leftarrow$ addr if TF = 1

Example:
```
JTF1: JTF TIMER      ;JUMP TO 'TIMER' ROUTINE
                     ;IF TF = 1
```

JTO address Jump If TEST 0 Is High

Opcode:

0 0 1 1	0 1 1 0	•	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Control passes to the specified address if the TEST 0 signal is high (= 1). Pin is sampled during SYNC.
(PC_{0-7}) ← addr if $T_0 = 1$

Example:
```
JT0HI: JT0 53          ;JUMP TO LOCATION 53 DEC
                       ;IF T0 = 1
```

JT1 address Jump If TEST 1 Is High

Opcode:

0 1 0 1	0 1 1 0	•	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Control passes to the specified address if the TEST 1 signal is high (= 1). Pin is sampled during SYNC.
(PC_{0-7}) ← addr if $T_1 = 1$

Example:
```
JT1HI: JT1 COUNT       ;JUMP TO 'COUNT' ROUTINE
                       ;IF T1 = 1
```

JZ address Jump If Accumulator Is Zero

Opcode:

1 1 0 0	0 1 1 0	•	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Control passes to the specified address if the accumulator contains all zeros at the time this instruction is executed.
(PC_{0-7}) ← addr if A = 0

Example:
```
JACCO: JZ OA3H         ;JUMP TO LOCATION 'A3' HEX
                       ;IF ACC VALUE IS ZERO
```

MOV A,#data Move Immediate Data to Accumulator

Opcode:

0 0 1 0	0 0 1 1	•	d_7 d_6 d_5 d_4	d_3 d_2 d_1 d_0

This is a 2-cycle instruction. The 8-bit value spedified by 'data' is loaded in the accumulator.
(A) ← data

Example:
```
MOV A,#OA3H            ;MOV 'A3' HEX TO ACC
```

MOV A,PSW Move PSW Contents to Accumulator

Opcode:

1 1 0 0	0 1 1 1

The contents of the program status word are moved to the accumulator.
(A) ← (PSW)

Example: Jump to 'RB1SET' routine if bank switch, PSW bit 4, is set.
```
BSCHK: MOV A,PSW       ;MOV PSW CONTENTS TO ACC
       JB4 RB1 SET     ;JUMP TO 'RB1SET' IF ACC
                       ;BIT 4 = 1
```

MOV A,Rr Move Register Contents to Accumulator

Opcode:

1 1 1 1	1 r_2 r_1 r_0

Eight bits of data are moved from working register 'r' into the accumulator.
(A) ← (Rr) r = 0–7

Example:
```
MAR: MOV A,R3          ;MOVE CONTENTS OF REG 3
                       ;TO ACC
```

MOV A,@Rr Move Data Memory Contents to Accumulator

Opcode:

1 1 1 1	0 0 0 r

The contents of the data memory location addressed by bits 0–7 of register 'r' are moved to the accumulator. Register 'r' contents are unaffected.
(A) ← ((Rr)) r = 0–1

Example: Assume R1 contains 00110110.
```
MADM: MOV A,@R1        ;MOVE CONTENTS OF DATA MEM
                       ;LOCATION 54 TO ACC
```

MOV A,T Move Timer/Counter Contents to Accumulator

Opcode:

0 1 0 0	0 0 1 0

The contents of the timer/event-counter register are moved to the accumulator. The timer/event-counter is not stopped.
(A) ← (T)

Example: Jump to "Exit" routine when timer reaches '64', that is, when bit 6 is set—assuming initialization to zero.
```
TIMCHK: MOV A,T        ;MOVE TIMER CONTENTS TO
                       ;ACC
        JB6 EXIT       ;JUMP TO 'EXIT' IF ACC BIT
                       ;6 = 1
```

MOV PSW,A Move Accumulator Contents to PSW

Opcode:

1 1 0 1	0 1 1 1

The contents of the accumulator are moved into the program status word. All condition bits and the stack pointer are affected by this move.
(PSW) ← (A)

Example: Move up stack pointer by two memory locations, that is, increment the pointer by one.
```
INCPTR: MOV A,PSW      ;MOVE PSW CONTENTS TO ACC
        INC A          ;INCREMENT ACC BY ONE
        MOV PSW,A      ;MOVE ACC CONTENTS TO PSW
```

MOV Rr,A Move Accumulator Contents to Register

Opcode: | 1 0 1 0 | 1 r_2 r_1 r_0 |

The contents of the accumulator are moved to register 'r'
(Rr) ← (A) r = 0–7

Example:
```
MRA MOV R0,A        ;MOVE CONTENTS OF ACC TO
                    ;REG 0
```

MOV Rr,#data Move Immediate Data to Register

Opcode: | 1 0 1 1 | 1 r_2 r_1 r_0 | • | d_7 d_6 d_5 d_4 | d_3 d_2 d_1 d_0 |

This is a 2-cycle instruction. The 8-bit value specified by 'data' is moved to register 'r'.
(Rr) ← data r = 0–7

Example:
```
MIR4: MOV R4,#HEXTEN    ;THE VALUE OF THE SYMBOL
                        ;'HEXTEN' IS MOVED INTO
                        ;REG 4
MIR5: MOV R5,#PI*(R*R)  ;THE VAUE OF THE
                        ;EXPRESSION 'PI*(R*R)'
                        ;IS MOVED INTO REG 5
MIR6: MOV R6,#OADH      ;'AD' HEX IS MOVED INTO
                        REG 6
```

MOV @Rr,A Move Accumulator Contents to Data Memory

Opcode: | 1 0 1 0 | 0 0 0 r |

The contents of the accumulator are moved to the data memory location whose address is specified by bits 0–7 of register 'r'. Register 'r' contents are unaffected.
((Rr)) ← (A) r = 0–1

Example: Assume R0 contains 11000111.
```
MDMA: MOV @R,A      ;MOVE CONTENTS OF ACC TO
                    ;LOCATION 7 (REG)
```

MOV @Rr,#data Move Immediate Data to Data Memory

Opcode: | 1 0 1 1 | 0 0 0 r | • | d_7 d_6 d_5 d_4 | d_3 d_2 d_1 d_0 |

This is a 2-cycle instruction. The 8-bit value specified by 'data' is moved to the standard data memory location addressed by register 'r', bit 0–7.

Example: Move the hexadecimal value AC3F to locations 62–63.
```
MIDM: MOV R0,#62        ;MOVE '62' DEC TO ADDR REG0
      MOV @RO,#OACH     ;MOVE 'AC' HEX TO LOCATION 62
      INC R0            ;INCREMENT REG 0 TO '63'
      MOV @R0,#3FH      ;MOVE '3F' HEX TO LOCATION 63
```

MOV STS,A Move Accumulator Contents to STS Register

Opcode:

1 0 0 1	0 0 0 0

The contents of the accumulator are moved into the status register. Only bits 4–7 are affected.
$(STS_{4-7}) \leftarrow (A_{4-7})$

Example: Set ST_4–ST_7 to "1".

```
MSTS: MOV A,#0F0H        ;SET ACC
      MOV STS,A          ;MOVE TO STS
```

MOV T,A Move Accumulator Contents to Timer/Counter

Opcode:

0 1 1 0	0 0 1 0

The contents of the accumulator are moved to the timer/event-counter register.
$(T) \leftarrow (A)$

Example: Initialize and start event counter.

```
INITEC: CLR A            ;CLEAR ACC TO ZEROS
        MOV T,A          ;MOVE ZEROS TO EVENT COUNTER
        STRT CNT         ;START COUNTER
```

MOVD A,Pp Move Port 4–7 Data to Accumulator

Opcode:

0 0 0 0	1 1 p_1 p_0

This is a 2-cycle instruction. Data on 8243 port 'p' is moved (read) to accumulator bits 0–3. Accumulator bits 4–7 are zeroed.
$(A_{0-3}) \leftarrow Pp$ p = 4–7
$(A_{4-7}) \leftarrow 0$

Note: Bits 0–1 of the opcode are used to represent PORTS 4–7. If you are coding in binary rather than assembly language, the mapping is as follows:

Bits		Port
P_1	P_0	
0	0	4
0	1	5
1	0	6
1	1	7

Example:

```
INPPT5: MOVD A,P5        ;MOVE PORT 5 DATA TO ACC
                         ;BITS 0–3, ZERO ACC BITS 4–7
```

MOVD Pp,A Move Accumulator Data to Port 4, 5, 6 and 7

Opcode:

0 0 1 1	1 1 p_1 p_0

This is a 2-cycle instruction. Data in accumulator bits 0–3 is moved (written) to 8243 port 'p'. Accumulator bits 4–7 are unaffected. (See NOTE above regarding port mapping.)

Example: Move data in accumulator to ports 4 and 5.

```
OUTP45: MOVD P4,A        ;MOVE ACC BITS 0–3 TO PORT 4
        SWAP A           ;EXCHANGE ACC BITS 0–3 AND 4–7
        MOVD P5,A        ;MOVE ACC BITS 0–3 TO PORT 5
```

MOVP A,@A Move Current Page Data to Accumulator

Opcode:

1 0 1 0	0 0 1 1

This is a 2-cycle instruction. The contents of the program memory location addressed by the accumulator are moved to the accumulator. Only bits 0–7 of the program counter are affected, limiting the program memory reference to the current page. The program counter is restored following this operation.
(A) ← ((A))

Note: This a 1-byte, 2-cycle instruction. If it appears in location 255 of a program memory page, @A addresses a location in the following page.

Example:

```
MOV128: MOV A,#128          ;MOVE '128' DEC TO ACC
        MOVP A,@A           ;CONTENTS OF 129TH LOCATION
                            ;IN CURRENT PAGE ARE MOVED TO
                            ;ACC
```

MOVP3 A,@A Move Page 3 Data to Accumulator

Opcode:

1 1 1 0	0 0 1 1

This is a 2-cycle instruction. The contents of the program memory location within page 3, addressed by the accumulator, are moved to the accumulator. The program counter is restored following this operation.
(A) ← ((A)) within page 3

Example: Look up ASCII equivalent of hexadecimal code in table contained at the beginning of page 3. Note that ASCII characters are designated by a 7-bit code; the eighth bit is always reset.

```
TABSCH: MOV A,#OB8H         ;MOVE 'B8' HEX TO ACC (10111000)
        ANL A,#7FH          ;LOGICAL AND ACC TO MASK BIT
                            ;7 (00111000)
MOVP3, A,@A                 ;MOVE CONTENTS OF LOCATION
                            ;'38' HEX IN PAGE 3 TO ACC
                            ;(ASCII '8')
```

Access contents of location in page 3 labelled TAB1. Assume current program location is not in page 3.

```
TABSCH: MOV A,#TAB1         ;ISOLATE BITS 0–7
                            ;OF LABEL
                            ;ADDRESS VALUE
        MOVP3 A,@A          ;MOVE CONTENT OF PAGE 3
                            ;LOCATION LABELED 'TAB1'
                            ;TO ACC
```

NOP The NOP Instruction

Opcode:

0 0 0 0	0 0 0 0

No operation is performed. Execution continues with the following instruction.

ORL A,Rr Logical OR Accumulator With Register Mask

Opcode:

0 1 0 0	1 r_2 r_1 r_0

Data in the accumulator is logically ORed with the mask contained in working register 'r'.
(A) ← (A) OR (Rr) r = 0–7

Example:

```
ORREG: ORL A,R4             ;'OR' ACC CONTENTS WITH
                            ;MASK IN REG 4
```

ORL A,@Rr Logical OR Accumulator With Memory Mask

Opcode:

0 1 0 0	0 0 0 r

Data in the accumulator is logically ORed with the mask contained in the data memory location referenced by register 'r', bits 0–7.
(A) ← (A) OR ((Rr)) r = 0–1

Example:
```
ORDM: MOVE R0,#3FH      ;MOVE '3F' HEX TO REG 0
      ORL A, @R0        ;'OR' ACC CONTENTS WITH MASK
                        ;IN LOCATION 63
```

ORL A,#Data Logical OR Accumulator With Immediate Mask

Opcode:

0 1 0 0	0 0 1 1	•	d_7 d_6 d_5 d_4	d_3 d_2 d_1 d_0

This is a 2-cycle instruction. Data in the accumulator is logically ORed with an immediately-specified mask.
(A) ← (A) OR data

Example:
```
ORID: ORL A,#'X'        ;'OR' ACC CONTENTS WITH MASK
                        ;01011000 (ASCII VALUE OF 'X')
```

ORL Pp,#data Logical OR Port 1–2 With Immediate Mask

Opcode:

1 0 0 0	1 0 p_1 p_0	•	d_7 d_6 d_5 d_4	d_3 d_2 d_1 d_0

This is a 2-cycle instruction. Data on port 'p' is logically ORed with an immediately-specified mask.
(Pp) ← (Pp) OR data p = 1–2 (see OUTL instruction)

Example:
```
ORP1: ORL P1,#OFH       ;'OR' PORT 1 CONTENTS WITH
                        ;MASK 'FF' HEX (SET PORT 1
                        'TO ALL ONES)
```

ORLD Pp,A Logical OR Port 4–7 With Accumulator Mask

Opcode:

1 0 0 0	1 1 p_1 p_0

This is a 2-cycle instruction. Data on 8243 port 'p' is logically ORed with the digit mask contained in accumulator bits 0–3,
(Pp) (Pp) OR (A_{0-3}) p = 4–7 (See MOVD instruction)

Example:
```
ORP7; ORLD P7,A         ;'OR' PORT 7 CONTENTS
                        ;WITH ACC BITS 0–3
```

OUT DBB,A Output Accumulator Contents to Data Bus Buffer

Opcode:

0 0 0 0	0 0 1 0

Contents of the accumulator are transferred to the Data Bus Buffer Output register and the Output Buffer Full (OBF) flag is set to one.
(DBB) ← (A)
OBF ← 1

Example:
```
OUTDBB: OUT DBB,A       ;OUTPUT THE CONTENTS OF
                        ;THE ACC TO DBBOUT
```

OUTL Pp,A Output Accumulator Data to Port 1 and 2

Opcode:

0 0 1 1	1 0 p_1 p_0

This is a 2-cycle instruction. Data residing in the accumulator is transferred (written) to port 'p' and latched.

(Pp) ← (A) P = 1–2

Note: Bits 0–1 of the opcode are used to represent PORT 1 and PORT 2. If you are coding in binary rather than assembly language, the mapping is as follows:

Bits		Port
P_1	P_0	
0	0	X
0	1	1
1	0	2
1	1	X

Example:

```
OUTLP; MOV A,R7          ;MOVE REG 7 CONTENTS TO ACC
       OUTL P2,A         ;OUTPUT ACC CONTENTS TO PORT2
       MOV A,R6          ;MOVE REG 6 CONTENTS TO ACC
       OUTL P1,A         ;OUTPUT ACC CONTENTS TO PORT 1
```

RET Return Without PSW Restore

Opcode:

1 0 0 0	0 0 1 1

This is a 2-cycle instruction. The stack pointer (PSW bits 0–2 is decremented. The program counter is then restored from the stack. PSW bits 4–7 are not restored.

(SP) ← (SP) − 1

(PC) ← ((SP))

RETR Return With PSW Restore

Opcode:

1 0 0 1	0 0 1 1

This is a 2-cycle instruction. The stack pointer is decremented. The program counter and bits 4–7 of the PSW are then restored from the stack. Note that RETR should be used to return from an interrupt, but should not be used within the interrupt service routine as it signals the end of an interrupt routine.

(SP) ← (SP) − 1

(PC) ← ((SP))

(PSW_{4-7}) ← ((SP))

RL A Rotate Left Without Carry

Opcode:

1 1 1 0	0 1 1 1

The contents of the accumulator are rotated left one bit. Bit 7 is rotated into the bit 0 position.

(A_{n+1}) ← (A_n) n = 0–6

(A_0) ← (A_7)

Example: Assume accumulator contains 10110001.

```
RLNC: RL A               ;NEW ACC CONTENTS ARE 01100011
```

RLC A Rotate Left Through Carry

Opcode:

1 1 1 1	0 1 1 1

The contents of the accumulator are rotated left one bit. Bit 7 replaces the carry bit; the carry bit is rotated into the bit 0 position.

$(A_{n+1}) \leftarrow (A_n)$ n = 0–6
$(A_0) \leftarrow (C)$
$(C) \leftarrow (A7)$

Example: Assume accumulator contains a 'signed' number; isolate sign without changing value.

```
RLTC: CLR C          ;CLEAR CARRY TO ZERO
      RLC A          ;ROTATE ACC LEFT, SIGN
                     ;BIT (7) IS PLACED IN CARRY
      RR A           ;ROTATE ACC RIGHT—VALUE
                     ;(BITS 0–6) IS RESTORED,
                     ;CARRY UNCHANGED, BIT 7
                     ;IS ZERO
```

RR A Rotate Right Without Carry

Opcode:

0 1 1 1	0 1 1 1

The contents of the accumulator are rotated right one bit. Bit 0 is rotated into the bit 7 position.

$(A) \leftarrow (A_{n+1})$ n = 0–6
$(A_7) \leftarrow (A_0)$

Example Assume accumulator contains 10110001.

```
RRNC: RRA            ;NEW ACC CONTENTS ARE 11011000
```

RRC A Rotate Right Through Carry

Opcode:

0 1 1 0	0 1 1 1

The contents of the accumulator are rotated one bit. Bit 0 replaces the carry bit; the carry bit is rotated into the bit 7 position.

$(A_n) \leftarrow (A_n + 1)$ n = 0–6
$(A_7) \leftarrow (C)$
$(C) \leftarrow (A_0)$

Example Assume carry is not set and accumulator contains 10110001.

```
RRTC: RRCA           ;CARRY IS SET AND ACC
                     ;CONTAINS 01011000
```

SEL RB0 Select Register Bank 0

Opcode:

1 1 0 0	0 1 0 1

PSW BIT 4 is set to zero. References to working registers 0–7 address data memory locations 0–7. This is the recommended setting for normal program execution.
(BS) ← 0

SEL RB1 Select Register Bank 1

Opcode:

1 1 0 1	0 1 0 1

PSW bit 4 is set to one. References to working registers 0–7 address data memory locations 24–31. This is the recommended setting for interrupt service routines, since locations 0–7 are left intact. The setting of PSW bit 4 in effect at the time of an interrupt is restored by the RETR instruction when the interrupt service routine is completed.

Example: Assume an IBF interrupt has occurred, control has passed to program memory location 3, and PSW bit 4 was zero before the interrupt.

```
LOC3: JMP INIT              ;JUMP TO ROUTINE 'INIT'
        .
        .
        .
INIT: MOV R7,A              ;MOV ACC CONTENTS TO
                            ;LOCATION 7
      SEL RB1               ;SELECT REG BANK 1
      MOV R7,#OFAH          ;MOVE 'FA' HEX TO LOCATION 31
        .
        .
        .
      SEL RB0               ;SELECT REG BANK 0
      MOV A,R7              ;RESTORE ACC FROM LOCATION 7
      RETR                  ;RETURN——RESTORE PC AND PSW
```

STOP TCNT Stop Timer/Event Counter

Opcode:

0 1 1 0	0 1 0 1

This instruction is used to stop both time accumulation and event counting.

Example: Disable interrupt, but jump to interrupt routine after eight overflows and stop timer. Count overflows in register 7.

```
START: DIS TCNTI          ;DISABLE TIMER INTERRUPT
       CLR A              ;CLEAR ACC TO ZERO
       MOV T,A            ;MOV ZERO TO TIMER
       MOV R7,A           ;MOVE ZERO TO REG 7
       STRT T             ;START TIMER
MAIN:  JTF COUNT          ;JUMP TO ROUTINE 'COUNT'
                          ;IF TF = 1 AND CLEAR TIMER FLAG
       JMP MAIN           ;CLOSE LOOP
COUNT: INC R7             ;INCREMENT REG 7
       MOV A,R7           ;MOVE REG 7 CONTENTS TO ACC
       JB3 INT            ;JUMP TO ROUTINE 'INT' IF ACC
                          ;BIT 3 IS SET (REG 7 = 8)
       JMP MAIN           ;OTHERWISE RETURN TO ROUTINE
                          ;MAIN
        .
        .
        .
INT: STOP TCNT            ;STOP TIMER
       JMP 7H             ;JUMP TO LOCATION 7 (TIMER
                          ;INTERRUPT ROUTINE)
```

STRT CNT Start Event Counter

Opcode:

0 1 0 0	0 1 0 1

The TEST 1 (T_1) pin is enabled as the event-counter input and the counter is started. The event-counter register is incremented with each high to low transition on the T_1 pin.

Example: Initialize and start event counter. Assume overflow is desired with first T_1 input.

```
STARTC: EN TCNTI          ;ENABLE COUNTER INTERRUPT
        MOV A,#OFFH       ;MOVE 'FF' HEX (ONES) TO
                          ;ACC
        MOV T,A           ;MOVE ONES TO COUNTER
        STRT CNT          ;INPUT AND START
```

STRT T Start Timer

Opcode:

0 1 0 1	0 1 0 1

Timer accumulation is initiated in the timer register. The register is incremented every 32 instruction cycles. The prescaler which counts the 32 cycles is cleared but the timer register is not.

Example: Initialize and start timer.

```
STARTT: EN TCNTI          ;ENABLE TIMER INTERRUPT
        CLR A             :CLEAR ACC TO ZEROS
        MOV T,A           ;MOVE ZEROS TO TIMER
        STRT T            ;START TIMER
```

SWAP A Swap Nibbles Within Accumulator

Opcode:

0 1 0 0	0 1 1 1

Bits 0–3 of the accumulator are swapped with bits 4-7 of the accumulator.
$(A_{4-7}) \longleftrightarrow (A_{0-3})$

Example: Pack bits 0–3 of locations 50-51 into location 50.

```
PCKDIG: MOV R0,#50          ;MOVE '50' DEC TO REG 0
        MOV R1,#51          ;MOVE '51' DEC TO REG 1
        XCHD A,@R0          ;EXCHANGE BIT 0-3 OF ACC
                            ;AND LOCATION 50
        SWAP A              ;SWAP BITS 0-3 AND 4-7 OF ACC
        XCHD A,@ R1         ;EXCHANGE BITS 0-3 OF ACC AND
                            ;LOCATION 51
        MOV @R0,A           ;MOVE CONTENTS OF ACC TO
                            ;LOCATION 51
```

XCH ARr Exchange Accumulator-Register Contents

Opcode:

0 0 1 0	1 r_2 r_1 r_0

The contents of the accumulator and the contents of working register 'r' are exchanged.
$(A) \longleftrightarrow (Rr)$ r = 0–7

Example: Move PSW contents to Reg 7 without losing accumulator contents.

```
XCHAR7: XCH A,R7             ;EXCHANGE CONTENTS OF REG 7
                             ;AND ACC
        MOV A,PSW            ;MOVE PSW CONTENTS TO ACC
        XCH, A,R7            ;EXCHANGE CONTENTS OF REG 7
                             ;AND ACC AGAIN
```

XCH A,@Rr Exchange Accumulator and Data Memory Contents

Opcode:

0 0 1 0	0 0 0 r

The contents of the accumulator and the contents of the data memory location addressed by bits 0–7 of register 'r' are exchanged. Register 'r' contents are unaffected.
$(A) \longleftrightarrow ((Rr))$ r = 0–1

Example: Decrement contents of location 52.

```
DEC 52: MOV R0,#52           ;MOVE '52' DEC TO ADDRESS
                             ;REG 0
        XCH A,@R0            ;EXCHANGE CONTENTS OF ACC
                             ;AND LOCATION 52
        DEC A                ;DECREMENT ACC CONTENTS
        XCH A,@R0            ;EXCHANGE CONTENTS OF ACC
                             ;AND LOCATION 52 AGAIN
```

XCHD A,@Rr Exchange Accumulator and Data Memory 4-bit Data

Opcode:

0 0 1 1	0 0 0 r

This instruction exchanges bits 0–3 of the accumulator with bits 0–3 of the data memory location addressed by bits 0–7 of register 'r'. Bits 4–7 of the accumulator, bits 4–7 of the data memory location, and the contents of register 'r' are unaffected.

$(A_{0-3}) \longleftrightarrow ((Rr_{0-3}))$ r = 0–1

Example: Assume program counter contents have been stacked in locations 22-23.

```
XCHNIB: MOV R0,#23      ;MOVE '23' DEC TO REG 0
        CLR A           ;CLEAR ACC TO ZEROS
        XCHD A,@R0      ;EXCHANGE BITS 0-3 OF ACC
                        ;AND LOCATION 23 (BITS 8-11
                        ;OF PC ARE ZEROED, ADDRESS
                        ;REFERS TO PAGE 0)
```

XRL A,Rr Logical XOR Accumulator With Register Mask

Opcode:

1 1 0 1	1 r_2 r_1 r_0

Data in the accumulator is EXCLUSIVE ORed with the mask contained in working register 'r'.

(A) $\longleftrightarrow$ (A) XOR (Rr) r = 0–7

Example:

```
XORREG: XRL A,R5        ;'XOR' ACC CONTENTS WITH
                        ;MASK IN REG 5
```

XRL A,@Rr Logical XOR Accumulator With Memory Mask

Opcode:

1 1 0 1	0 0 0 r

Data in the accumulator is EXCLUSIVE ORed with the mask contained in the data memory location address by register 'r', bits 0–7.

(A) $\longleftarrow$ (A) XOR ((Rr)) r = 0–1

Example:

```
XORDM: MOV R1,#20H      ;MOVE '20' HEX TO REG 1
       XRL A,@R1        ;'XOR' ACC CONTENTS WITH MASK
                        ;IN LOCATION 32
```

XRL A,#data, Logical XOR Accumulator With Immediate Mask

Opcode:

1 1 0 1	0 0 1 1	•	d_7 d_6 d_5 d_4	d_3 d_2 d_1 d_0

This is a 2-cycle instruction. Data in the accumulator is EXCLUSIVE ORed with an immediately-specified mask.

(A) $\longleftarrow$ (A) XOR data

Example:

```
XORID: XRL A,#HEXTEN    ;XOR CONTENTS OF ACC WITH
                        ;MASK EQUAL VALUE OF SYMBOL
                        ;'HEXTEN'
```

CHAPTER 4
SINGLE-STEP AND PROGRAMMING POWER-DOWN MODES

SINGLE-STEP

The UPI family has a single-step mode which allows the user to manually step through his program one instruction at a time. While stopped, the address of the next instruction to be fetched is available on PORT 1 and the lower 2 bits of PORT 2. The single-step feature simplifies program debugging by allowing the user to easily follow program execution.

Figure 4-1 illustrates a recommended circuit for single-step operation, while Figure 4-2 shows the timing relationship between the SYNC output and the $\overline{SS}$ input. During single-step operation, PORT 1 and part of PORT 2 are used to output address information. In order to retain the normal I/O functions of PORTS 1 and 2, a separate latch can be used as shown in Figure 4-3.

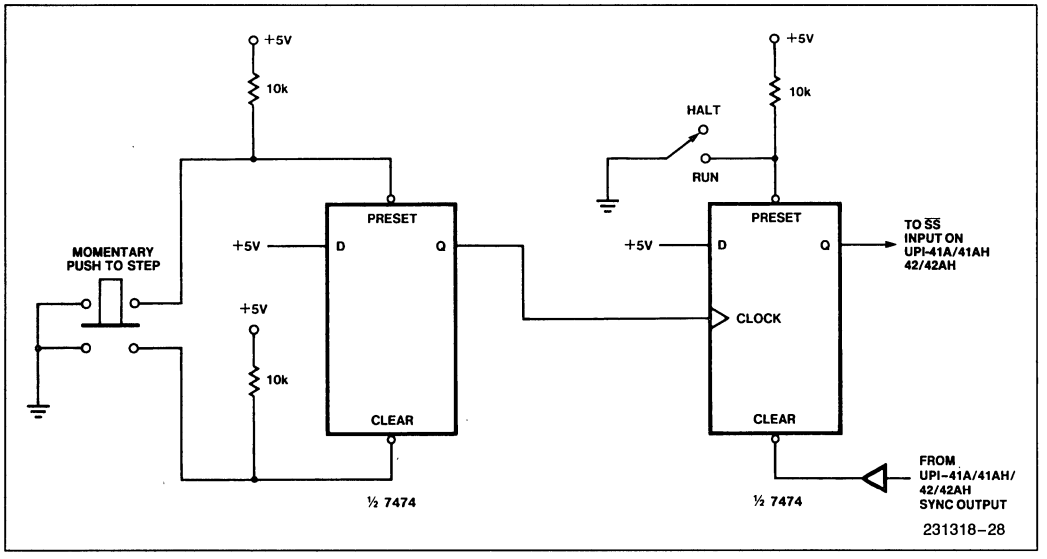


Figure 4-1. Single-Step Circuit

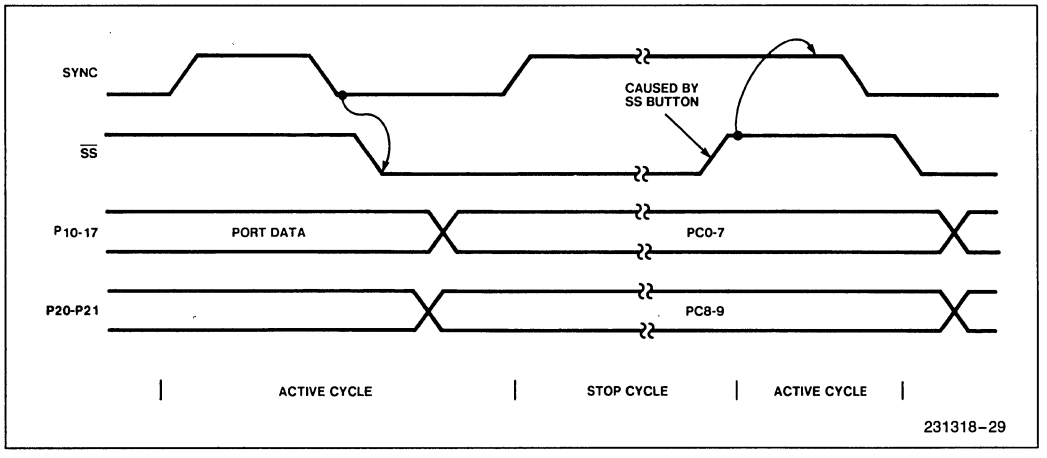


Figure 4-2. Single-Step Timing

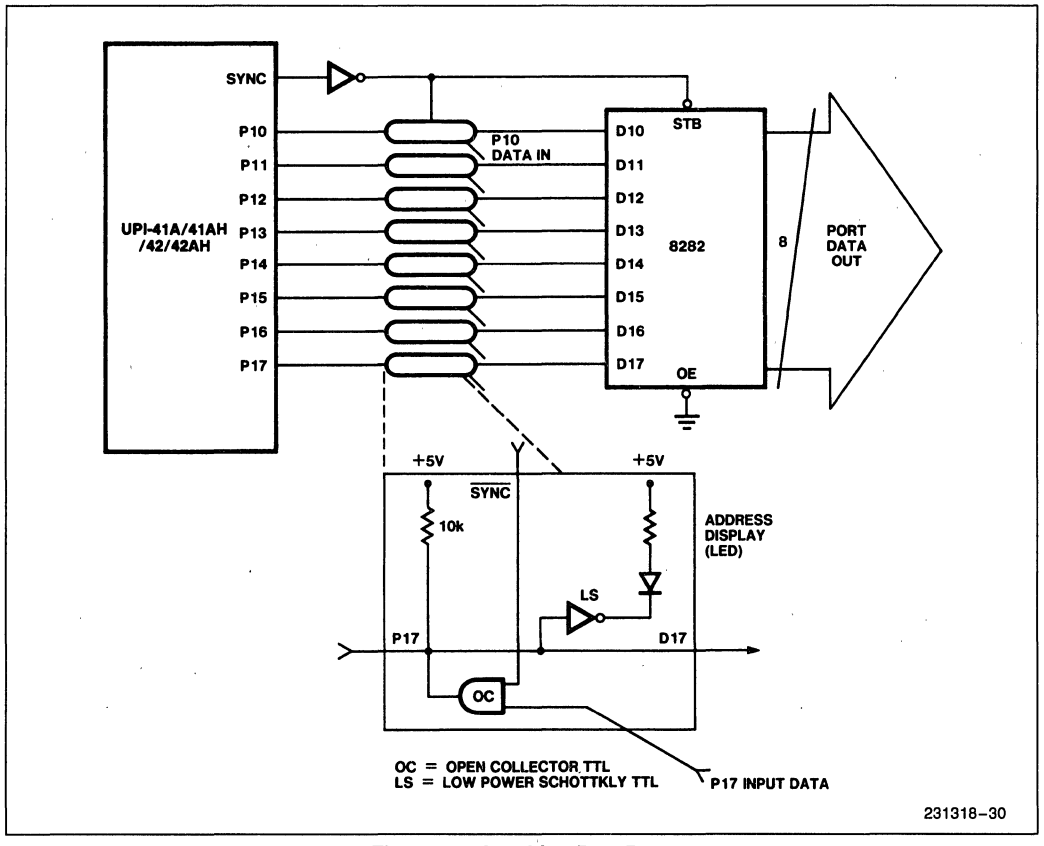


Figure 4-3. Latching Port Data

Timing

The sequence of single-step operation is as follows:

1) The processor is requested to stop by applying a low level on $\overline{SS}$. The $\overline{SS}$ input should not be brought low while SYNC is high. (The UPI samples the $\overline{SS}$ pin in the middle of the SYNC pulse).
2) The processor responds to the request by stopping during the instruction fetch portion of the next instruction. If a double cycle instruction is in progress when the single-step command is received, both cycles will be completed before stopping.
3) The processor acknowledges it has entered the stopped state by raising SYNC high. In this state, which can be maintained indefinitely, the 10-bit address of the next instruction to be fetched is preset on PORT 1 and the lower 2 bits of PORT 2.
4) $\overline{SS}$ is then raised high to bring the processor out of the stopped mode allowing it to fetch the next instruction.The exit from stop is indicated by the processor bringing SYNC low.
5) To stop the processor at the next instruction $\overline{SS}$ must be brought low again before the next SYNC pulse—the circuit in Figure 4-1 uses the trailing edge of the previous pulse. If $\overline{SS}$ is left high, the processor remains in the "RUN" mode.

Figure 4-1 shows a schematic for implementing single-step. A single D-type flip-flop with preset and clear is used to generate $\overline{SS}$. In the RUN mode $\overline{SS}$ is held high by keeping the flip-flop preset (preset has precedence over the clear input). To enter single-step, preset is removed allowing SYNC to bring $\overline{SS}$ low via the clear input. Note that SYNC must be buffered since the SN7474 is equivalent to 3 TTL loads.

The processor is now in the stopped state. The next instruction is initiated by stoppe state. The next instruction is initiated by clocking "1" the flip-flop. This "1" will not appear on $\overline{SS}$ unless SYNC is high (I.e., clear must be removed from the flip-flop). In response to $\overline{SS}$ going high, the processor begins an instruction fetch which brings SYNC low. $\overline{SS}$ is then reset through the clear input and the processor again enters the stopped state.

EXTERNAL ACCESS

The UPI family has an External Access mode (EA) which puts the processor into a test mode. This mode allows the user to disable the internal program memory and execute from external memory. External Access mode is useful in testing because it allows the user to test the processor's functions directly. It is only useful for testing since this mode uses D_0–D_7, PORTS 10–17 and PORTS 20–22.

This mode is invoked by connecting the EA pin to 5V. The 11-bit current program counter contents then come out on PORTS 10–17 and PORTS 20–22 after the SYNC output goes high. (PORT 10 is the least significant bit.) The desired instruction opcode is placed on D_0–D_7 before the start of state S_1. During state S_1, the opcode is sampled from D_0–D_7 and subsequently executed in place of the internal program memory contents.

The program counter contents are multiplexed with the I/O port data on PORTS 10–17 and PORTS 20–22. The I/O port data may be demultiplexed using an external latch on the rising edge of SYNC. The program counter contents may be demultiplexed similarly using the trailing edge of SYNC.

Reading and/or writing the Data Bus Buffer registers is still allowed although only when D_0–D_7 are not being sampled for opcode data. In practice, since this sampling time is not known externally, reads or writes on the system bus are done during SYNC high time. Approximately 600 ns are available for each read or write cycle.

POWER DOWN MODE (UPI-41AH/42AH ONLY)

Extra circuitry is included in the UPI-41AH/42AH version to allow low-power, standby operation. Power is removed from all system elements except the internal data RAM in the low-power mode. Thus the contents of RAM can be maintained and the device draws only 10 to 15% of its normal power.

The V_{CC} pin serves as the 5V power supply pin for all of the UPI-41AH/42AH version's circuitry except the data RAM array. The V_{DD} pin supplies only the RAM array. In normal operation, both V_{CC} and V_{DD} are connected to the same 5V power supply.

To enter the Power-Down mode, the $\overline{\text{RESET}}$ signal to the UPI is asserted. This ensures the memory will not be inadvertently altered by the UPI during power-down. The V_{CC} pin is then grounded while V_{DD} is maintained at 5V. Figure 4-4 illustrates a recommended Power-Down sequence. The sequence typically occurs as follows:

1) Imminent power supply failure is detected by user defined circuitry. The signal must occur early enough to guarantee the UPI-41AH/42AH can save all necessary data before V_{CC} falls outside normal operating tolerance.
2) A "Power Failure" signal is used to interrupt the processor (via a timer overflow interrupt, for instance) and call a Power Failure service routine.
3) The Power Failure routine saves all important data and machine status in the RAM array. The routine may also initiate transfer of a backup supply to the V_{DD} pin and indicate to external circuitry that the Power Failure routine is complete.
4) A $\overline{\text{RESET}}$ signal is applied by external hardware to guarantee data will not be altered as the power supply falls out of limits. $\overline{\text{RESET}}$ must be low until V_{CC} reaches ground potential.

Recovery from the Power-Down mode can occur as any other power-on sequence. An external 1 μfd capacitor on the $\overline{\text{RESET}}$ input will provide the necessary initialization pulse.

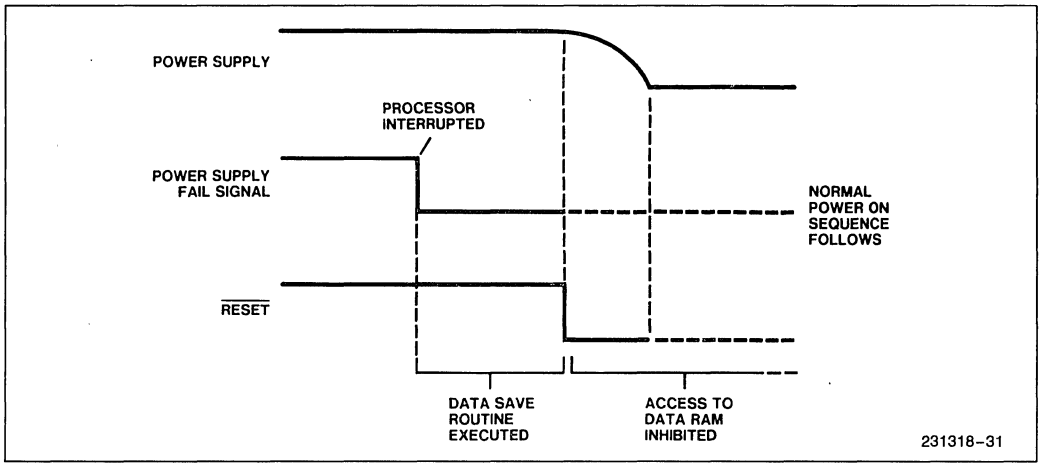


Figure 4-4. Power-Down Sequence

CHAPTER 5
SYSTEM OPERATION

BUS INTERFACE

The UPI-41A/41AH/42/42AH Microcomputer functions as a peripheral to a master processor by using the data bus buffer registers to handle data transfers. The DBB configuration is illustrated in Figure 5-1. The UPI Microcomputer's 8 three-state data lines (D_7–D_0) connect directly to the master processor's data bus. Data transfer to the master is controlled by 4 external inputs to the UPI:

- A_0 Address Input signifying command or data
- $\overline{CS}$ Chip Select
- $\overline{RD}$ Read strobe
- $\overline{WR}$ Write strobe

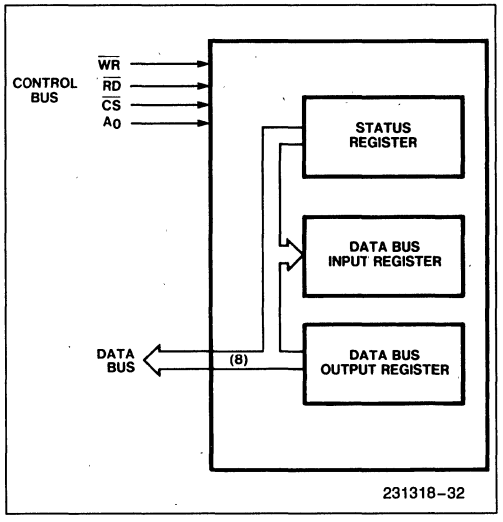


Figure 5-1. Data Bus Register Configuration

The master processor addresses the UPI-41A/41AH/42/42AH Microcomputer as a standard peripheral device. Table 5-1 shows the conditions for data transfer:

Table 5-1. Data Transfer Controls

$\overline{CS}$	A_0	$\overline{RD}$	$\overline{WR}$	Condition
0	0	0	1	Read DBBOUT
0	1	0	1	Read STATUS
0	0	1	0	Write DBBIN data, set $F_1 = 0$
0	1	1	0	Write DBBIN command set $F_1 = 1$
1	x	x	x	Disable DBB

Reading the DBBOUT Register

The sequence for reading the DBBOUT register is shown in Figure 5-2. This operation causes the 8-bit contents of the DBBOUT register to be placed on the system Data Bus. The OBF flag is cleared automatically.

Reading STATUS

The sequence for reading the UPI Microcomputer's 8 STATUS bits is shown in Figure 5-3. This operation causes the 8-bit STATUS register contents to be placed on the system Data Bus as shown.

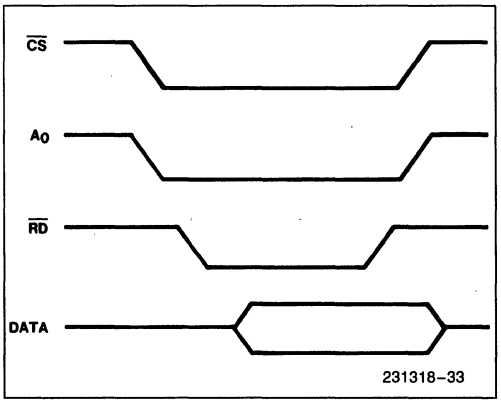


Figure 5-2. DBBOUT Read

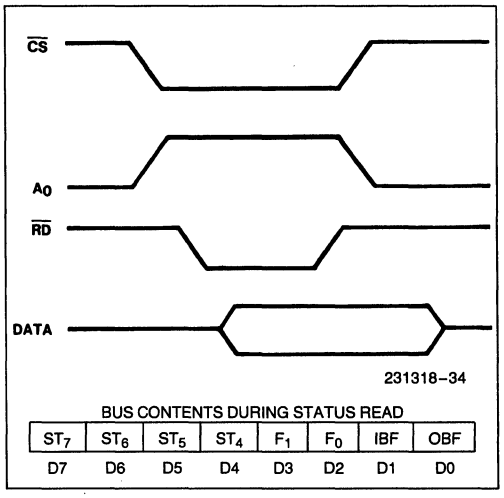


Figure 5-3. Status Read

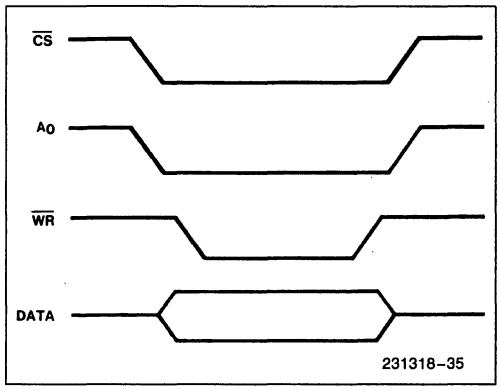


Figure 5-4. Writing Data to DBBIN

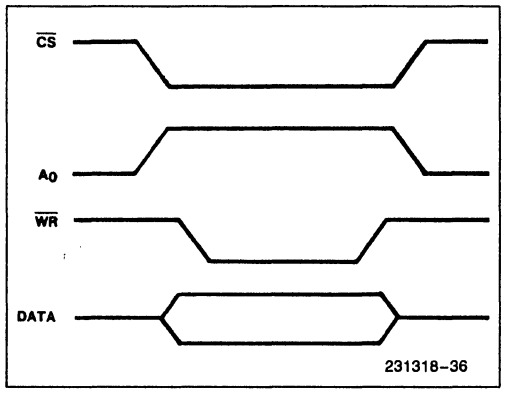


Figure 5-5. Writing Commands to DBBIN

Write Data to DBBIN

The sequence for writing data to the DBBIN register is shown in Figure 5-4. This operation causes the system Data Bus contents to be transferred to the DBBIN register and the IBF flag is set. Also, the F_1 flag is cleared ($F_1 = 0$) and an interrupt request is generated. When the IBF interrupt is enabled, a jump to location 3 will occur. The interrupt request is cleared upon entering the IBF service routine or by a system RESET input.

Writing Commands to DBBIN

The sequence for writing commands to the DBBIN register is shown in Figure 5-5. This sequence is identical to a data write except that the A_0 input is latched in the F_1 flag ($F_1 = 1$). The IBF flag is set and an interrupt request is generated when the master writes a command to DBB.

Operations of Data Bus Registers

The UPI-41A/41AH/42/42AH Microcomputer controls the transfer of DBB data to its accumulator by executing INput and OUTput instructions. An IN A,DBB instruction causes the contents to be transferred to the UPI accumulator and the IBF flag is cleared.

The OUT DBB,A instruction causes the contents of the accumulator to be transferred to the DBBOUT register. The OBF flag is set.

The UPI's data bus buffer interface is applicable to a variety of microprocessors including the 8086, 8088, 8085AH, 8080, and 8048.

A description of the interface to each of these processors follows.

DESIGN EXAMPLES

8085AH Interface

Figure 5-6 illustrates an 8085AH system using a UPI-41A/41AH/42/42AH. The 8085AH system uses a multiplexed address and data bus. During I/O the 8 upper address lines (A_8–A_{15}) contain the same I/O address as the lower 8 address/data lines (A_0–A_7); therefore I/O address decoding is done using only the upper 8 lines to eliminate latching of the address. An 8205 decoder provides address decoding for both the UPI and the 8237. Data is transferred using the two DMA handshaking lines of PORT 2. The 8237 performs the actual bus transfer operation. Using the UPI-41A/41AH/42/42AH's OBF master interrupt, the UPI notifies the 8085AH upon transfer completion using the RST 5.5 interrupt input. The $\overline{IBF}$ master interrupt is not used in this example.

8088 Interface

Figure 5-7 illustrates a UPI-41A/41AH/42/42AH interface to an 8088 minimum mode system. Two 8-bit latches are used to demultiplex the address and data bus. The address bus is 20-lines wide. For I/O only, the lower 16 address lines are used, providing an addressing range of 64K. UPI address selection is accomplished using an 8205 decoder. The A_0 address line of the bus is connected to the corresponding UPI input for register selection. Since the UPI is polled by the 8088, neither DMA nor master interrupt capabilities of the UPI are used in the figure.

8086 Interface

The UPI-41A/41AH/42/42AH can be used on an 8086 maximum mode system as shown in Figure 5-8. The address and data bus is demultiplexed using three

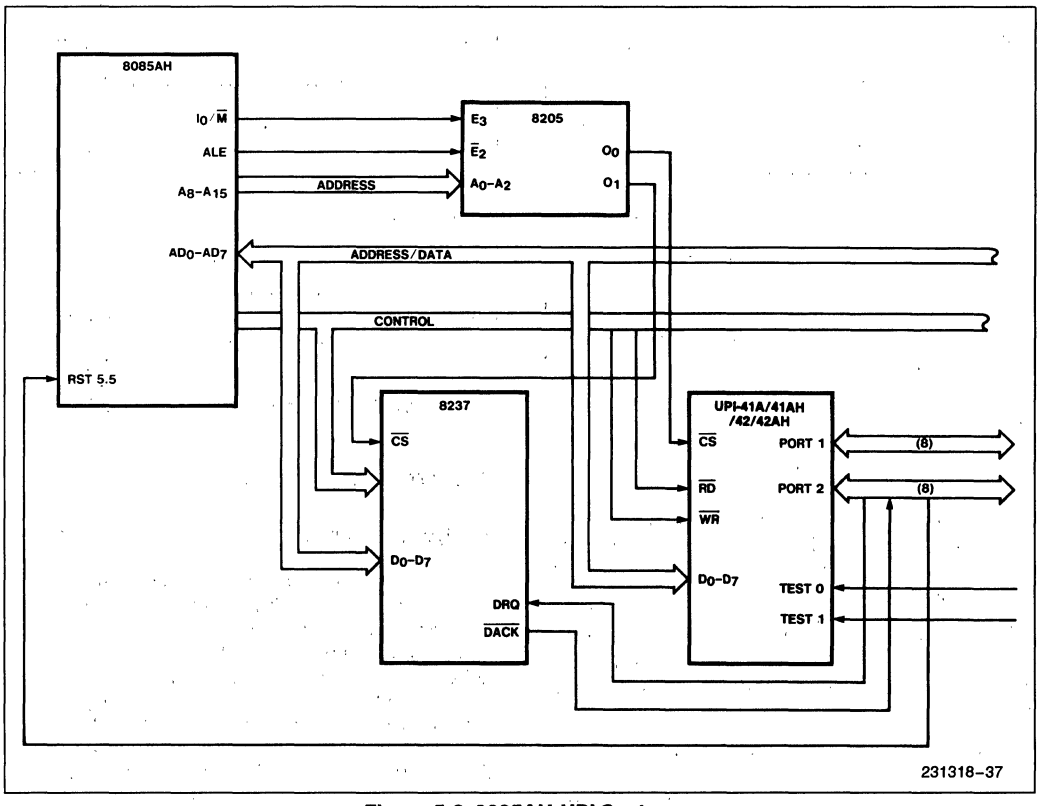


Figure 5-6. 8085AH-UPI System

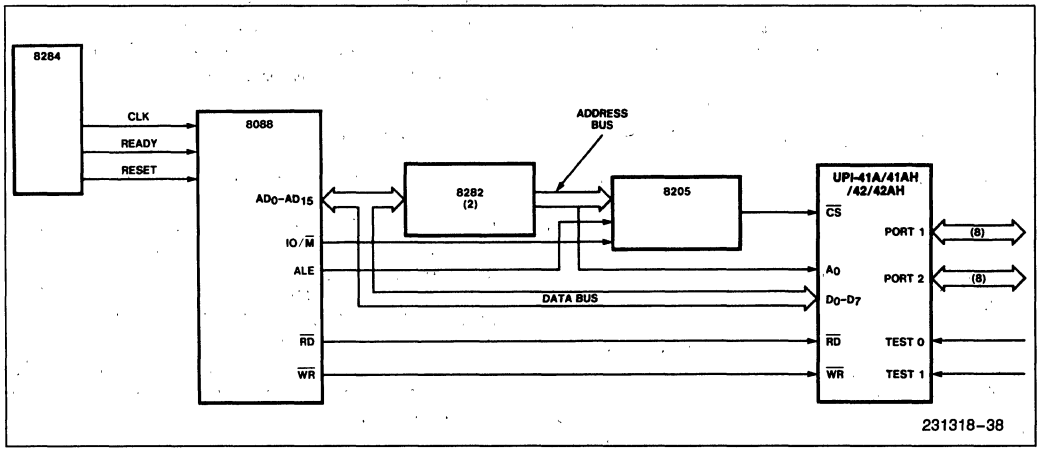


Figure 5-7. 8088-UPI Minimum Mode System

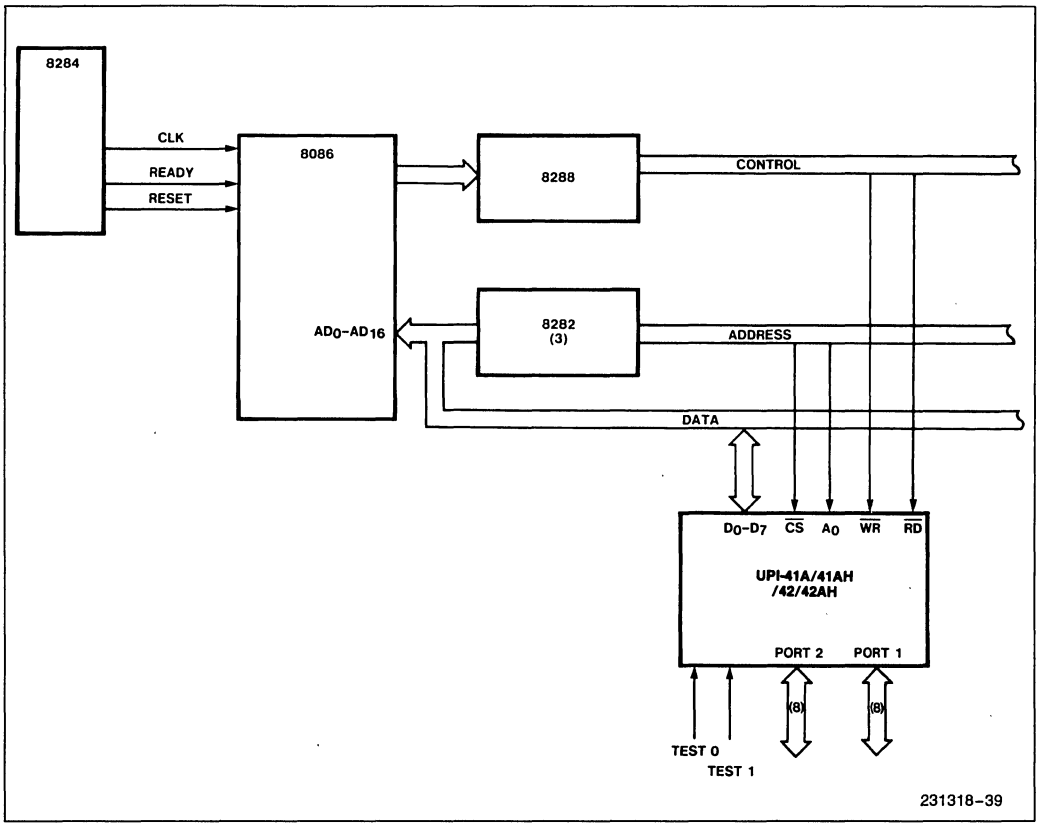


Figure 5-8. 8086-UPI Maximum Mode Systems

8282 latches providing separate address and data buses. The address bus is 20-lines wide and the data bus is 16-lines wide. Multiplexed control lines are decoded by the 8288. The UPI's $\overline{CS}$ input is provided by linear selection. Note that the UPI is both I/O mapped and memory mapped as a result of the linear addressing technique. An address decoder may be used to limit the UPI-41A/41AH/42/42AH to a specific I/O mapped address. Address line A_1 is connected to the UPI's A_0 input. This insures that the registers of the UPI will have even I/O addresses. Data will be transferred on D_0–D_7 lines only. This allows the I/O registers to be accessed using byte manipulation instructions.

8080 Interface

Figure 5-9 illustrates the interface to an 8080A system. In this example, a crystal and capacitor are used for UPI-41A/41AH/42/42AH timing reference and power-on RESET. If the 2-MHz 8080A 2-phase clock were used instead of the crystal, the UPI-41A/41AH/42/42AH would run at only 16% full speed.

The A_0 and $\overline{CS}$ inputs are direct connections to the 8080 address bus. In larger systems, however, either of these inputs may be decoded from the 16 address lines.

The $\overline{RD}$ and $\overline{WR}$ inputs to the UPI can be either the $\overline{IOR}$ and $\overline{IOW}$ or the $\overline{MEMR}$ and $\overline{MEMR}$ signals depending on the I/O mapping technique to be used.

The UPI can be addressed as an I/O device using INput and OUTput instructions in 8080 software.

8048 Interface

Figure 5-10 shows the UPI interface to an 8048 master processor.

The 8048 $\overline{RD}$ and $\overline{WR}$ outputs are directly compatible with the UPI. Figure 5-11 shows a distributed processing system with up to seven UPI's connected to a single 8048 master processor.

In this configuration the 8048 uses PORT 0 as a data bus. I/O PORT 2 is used to select one of the seven

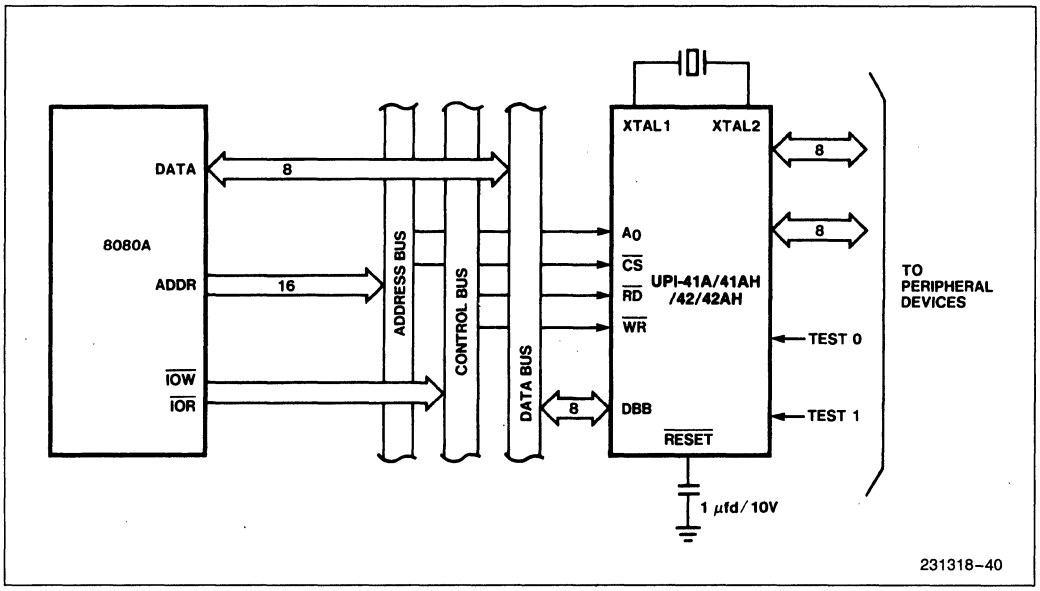


Figure 5-9. 8080A-UPI Interface

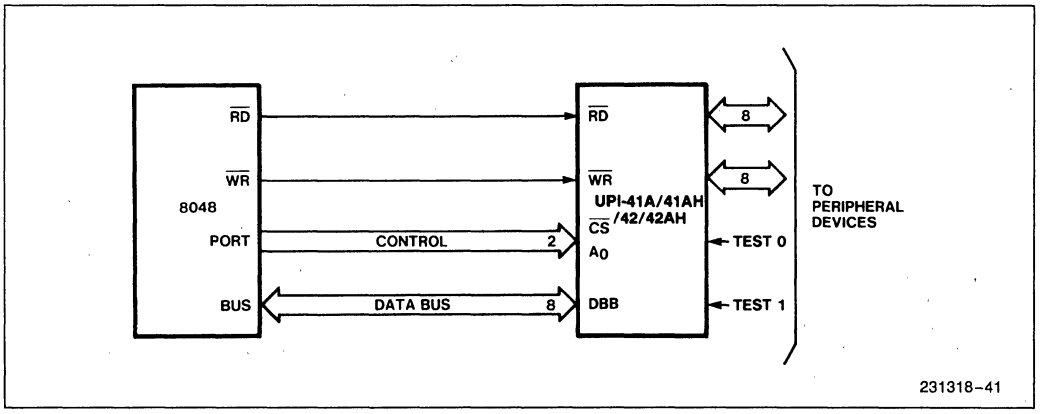


Figure 5-10. 8048-UPI Interface

UPI's when data transfer occurs. The UPI's are programmed to handle isolated tasks and, since they operate in parallel, system throughput is increased.

GENERAL HANDSHAKING PROTOCOL

1) Master reads STATUS register ($\overline{RD}$, $\overline{CS}$, A_0 = (0, 0, 1)) in polling or in response to either an $\overline{IBF}$ or an OBF interrupt.
2) If the UPI DBBIN register is empty (IBF flag = 0), Master writes a word to the DBBIN register ($\overline{WR}$, $\overline{CS}$, A_0 = (0, 0, 1) or (0, 0, 0)). If A_0 = 1, write command word, set F_1. If A_0 = 0, write data word, F_1 = 0.
3) If the UPI DBBOUT register is full (OBF flag = 1), Master reads a word from the DBBOUT register ($\overline{RD}$, $\overline{CS}$, A_0 = (0,0, 0)).
4) UPI recognizes IBF (via IBF interrupt or JNIBF). Input data or command word is processed, depending on F1; IBF is reset. Repeat step 1 above.
5) UPI recognizes OBF flag = 0 (via JOBF). Next word is output to DBBOUT register, OBF is set. Repeat step 1 above.

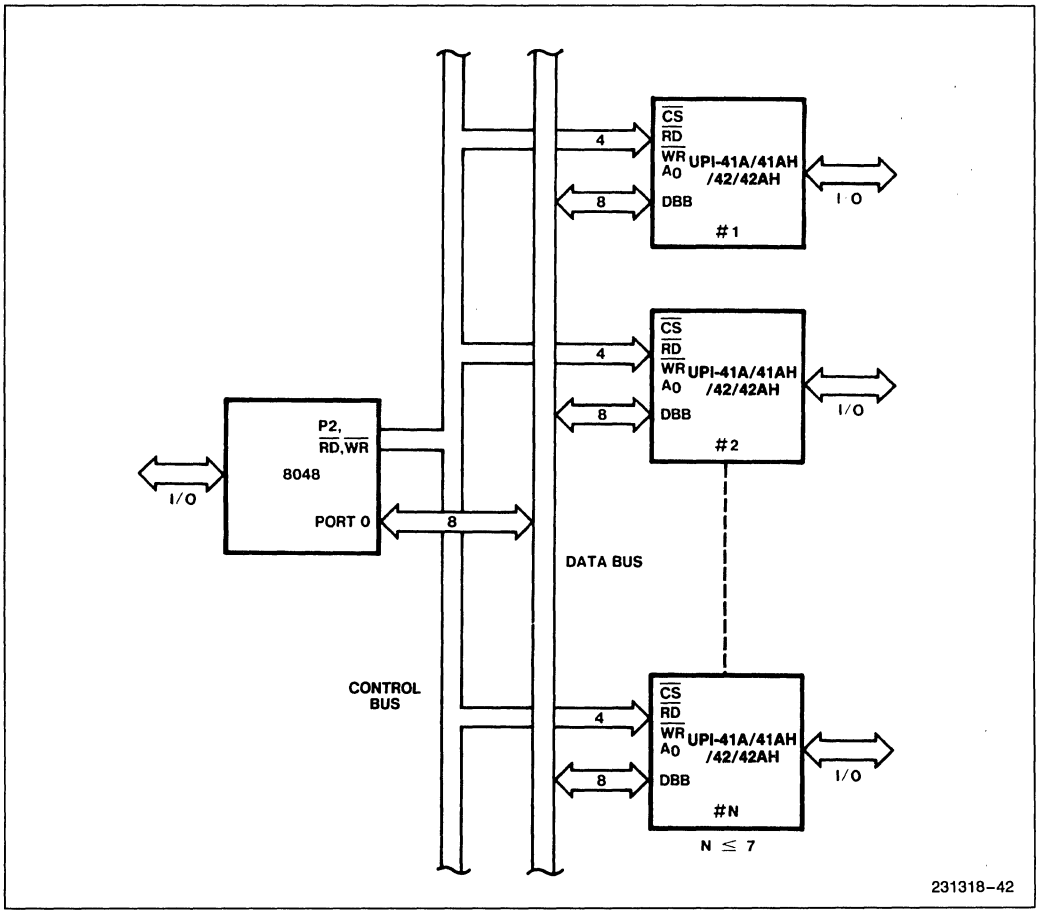


Figure 5-11. Distributed Processor System

CHAPTER 6
APPLICATIONS

ABSTRACTS

The UPI-41A/41AH/42/42AH is designed to fill a wide variety of low to medium speed peripheral interface applications where flexibility and easy implementation are important considerations. The following examples illustrate some typical applications.

Keyboard Encoder

Figure 6-1 illustrates a keyboard encoder configuration using the UPI and the 8243 I/O expander to scan a 128-key matrix. The encoder has switch matrix scanning logic, N-key rollover logic, ROM look-up table, FIFO character buffer, and additional outputs for display functions, control keys or other special functions.

PORT 1 and PORTs 4–7 provide the interface to the keyboard. PORT 1 lines are set one at a time to select the various key matrix rows.

When a row is energized all 16 columns (i.e., PORTs 4–7 inputs) are sampled to determine if any switch in the row is closed. The scanning software is code efficient because the UPI instruction set includes individual bit set/clear operations and expander PORTs 4–7 can be directly addressed with single, 2-byte instructions. Also, accumulator bits can be tested in a single operation. Scan time for 128 keys is about 10 ms. Each matrix point has a unique binary code which is used to address ROM when a key closure is detected. Page 3 of ROM contains a look-up table with useable codes (i.e., ASCII, EBCDIC, etc.) which correspond to each key. When a valid key closure is detected the ROM code corresponding to that key is stored in a FIFO buffer in data memory for transfer to the master processor. To avoid stray noise and switch bounce, a key closure must be detected on two consecutive scans before it is considered valid and loaded into the FIFO buffer. The FIFO buffer allows multiple keys to be processed as they are depressed without regard to when they are released, a condition known as N-key rollover.

The basic features of this encoder are fairly standard and require only about 500 bytes of memory. Since the UPI is programmable and has additional memory capacity it can handle a number of other functions. For example, special keys can be programmed to give an entry on closing as well as opening. Also, I/O lines are

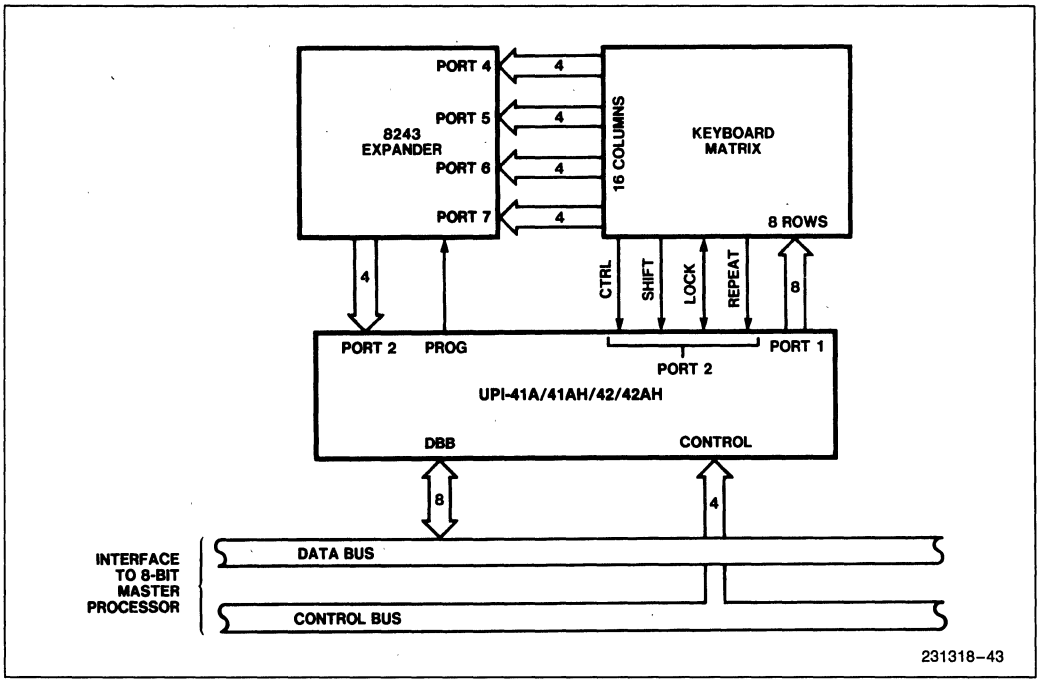


Figure 6-1. Keyboard Encoder Configuration

available to control a 16-digit, 7-segment display. The UPI can also be programmed to recognize special combinations of characters such as commands, then transfer only the decoded information to the master processor.

Matrix Printer Interface

The matrix printer interface illustrated in Figure 6-2 is a typical application for the UPI. The actual printer mechanism could be any of the numerous dot-matrix types and similar configurations can be shown for drum, spherical head, daisy wheel or chain type printers.

The bus structure shown represents a generalized, 8-bit system bus configuration. The UPI's three-state interface port and asynchronous data buffer registers allow it to connect directly to this type of system for efficient, two-way data transfer.

The UPI's two on-board I/O ports provide up to 16 input and output signals to control the printer mechanism. The timer/event counter is used for generating a timing sequence to control print head position, line feed, carriage return, and other sequences. The on-board program memory provides character generation for 5 x 7, 7 x 9, or other dot matrix formats. As an added feature a portion of the data memory can be used as a FIFO buffer so that the master processor can send a block of data at a high rate. The UPI can then output characters from the buffer at a rate the printer can accept while the master processor returns to other tasks.

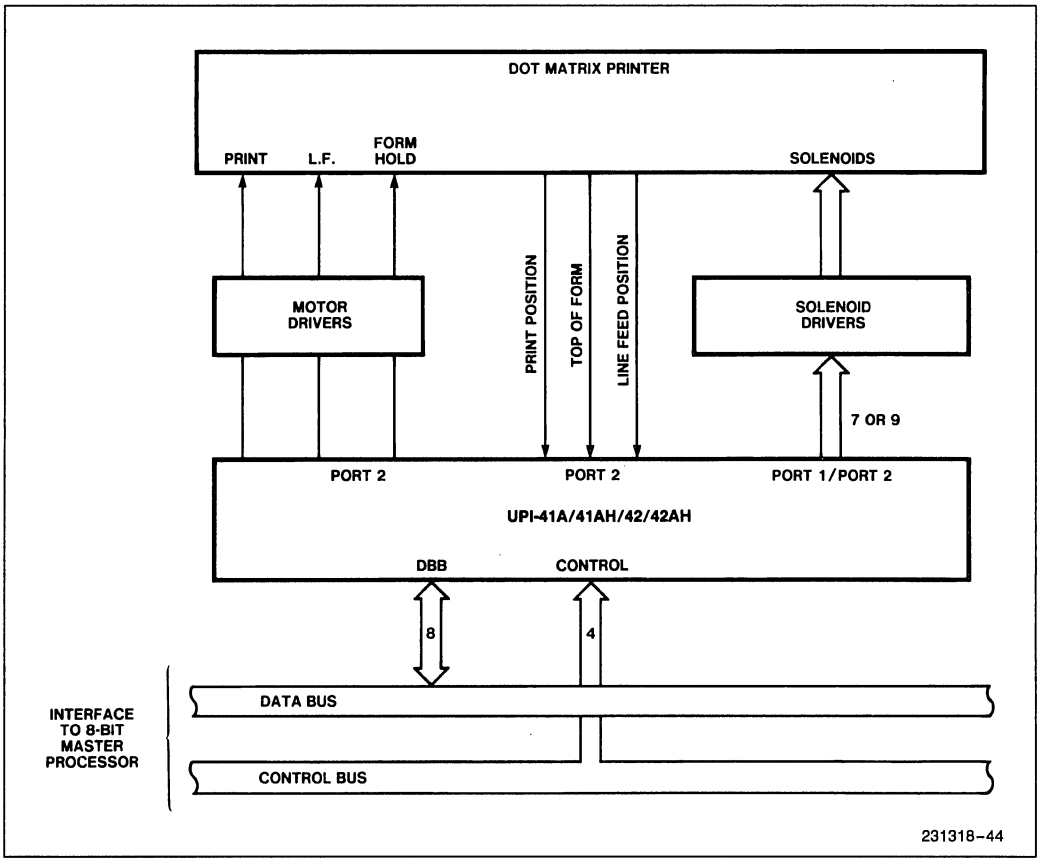


Figure 6-2. Matrix Printer Controller

The 8295 Printer Controller is an example of an UPI preprogrammed as a dot matrix printer interface.

Tape Cassette Controller

Figure 6-3 illustrates a digital cassette interface which can be implemented with the UPI. Two sections of the tape transport are controlled by the UPI: digital data/command logic, and motor servo control.

The motor servo requires a speed reference in the form of a monostable pulse whose width is proportional to the desired speed. The UPI monitors a prerecorded clock from the tape and uses its on-board interval timer to generate the required speed reference pulses at each clock transition.

Recorded data from the tape is supplied serially by the data/command logic and is converted to 8-bit words by the UPI, then transferred to the master processor. At 10 ips tape speed the UPI can easily handle the 8000 bps data rate. To record data, the UPI uses the two input lines to the data/command logic which control the flux direction in the recording head. The UPI also monitors 4 status lines from the tape transport including: end of tape, cassette inserted, busy, and write permit. All control signals can be handled by the UPI's two I/O ports.

Universal I/O Interface

Figure 6-4 shows an I/O interface design based on the UPI. This configuration includes 12 parallel I/O lines and a serial (RS232C) interface for full duplex data transfer up to 1200 baud. This type of design can be used to interface a master processor to a broad spectrum of peripheral devices as well as to a serial communication channel.

PORT 1 is used strictly for I/O in this example while PORT 2 lines provide five functions:

- P_{23}–P_{20} I/O lines (bidirectional)
- P_{24} Request to send (RTS)
- P_{25} Clear to send (CTS)
- P_{26} Interrupt to master
- P_{27} Serial data out

The parallel I/O lines make use of the bidirectional port structure of the UPI. Any line can function as an input or output. All port lines are automatically initialized to 1 by a system $\overline{\text{RESET}}$ pulse and remain latched. An external TTL signal connected to a port line will override the UPI's 50 KΩ internal pull-up so that an INPUT instruction will correctly sample the TTL signal.

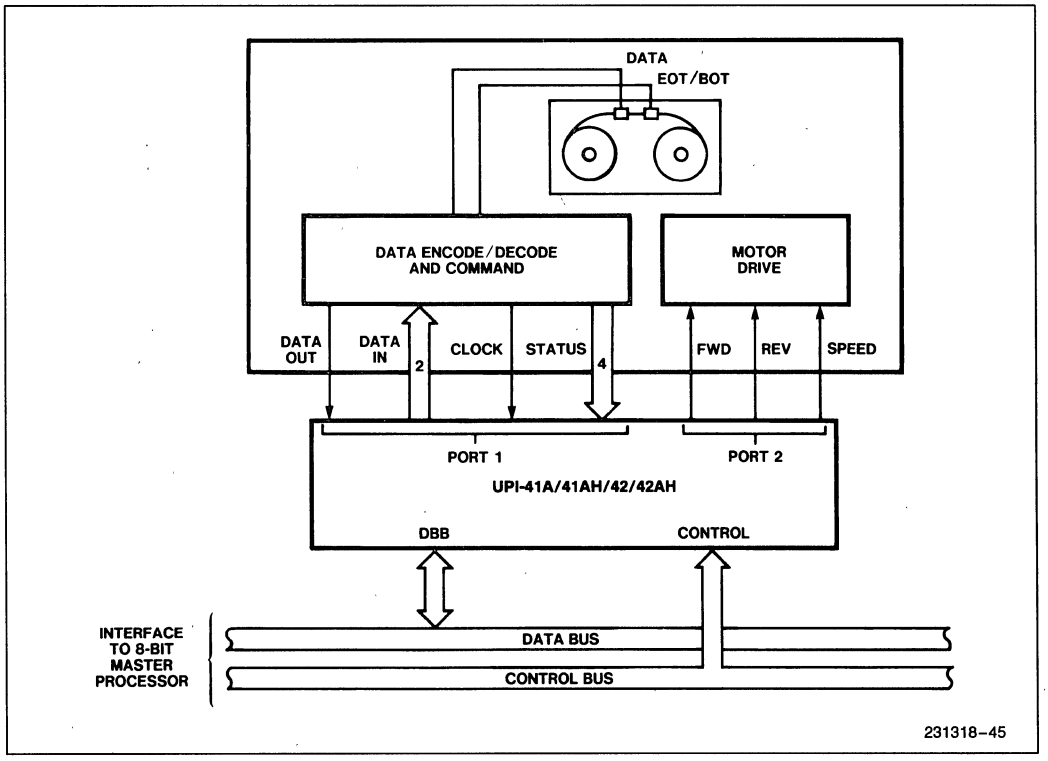


Figure 6-3. Tape Transport Controller

Four PORT 2 lines function as general I/O similar to PORT 1. Also, the RTS signal is generated on PORT 2 under software control when the UPI has serial data to send. The CTS signal is monitored via PORT 2 as an enable to the UPI to send serial data. A PORT 2 line is also used as a software generated interrupt to the master processor. The interrupt functions as a service request when the UPI has a byte of data to transfer or when it is ready to receive. Alternatively, the EN FLAGS instruction could be used to create the OBF and $\overline{IBF}$ interrupts on P_{24} and P_{25}.

The RS232C interface is implemented using the TEST 0 pin as a receive input and a PORT 2 pin as a transmit output. External packages (A_0, A_1) are used to provide RS232C drive requirements. The serial receive software is interrupt driven and uses the on-chip timer to perform time critical serial control. After a start bit is detected the interval timer can be preset to generate an interrupt at the proper time for sampling the serial bit stream. This eliminates the need for software timing loops and allows the processor to proceed to other tasks (i.e., parallel I/O operations) between serial bit samples. Software flags are used so the main program can determine when the interrupt driven receive program has a character assembled for it.

This type of configuration allows system designers flexibility in designing custom I/O interfaces for specific serial and parallel I/O applications. For instance, a second or third serial channel could be substituted in place of the parallel I/O if required. The UPI's data memory can buffer data and commands for up to 4 low-speed channels (110 baud teletypewriter, etc.)

Application Notes

The following application notes illustrate the various applications of the UPI family. Other related publications including the *Microcontroller Handbook* are available through the Intel Literature Department.

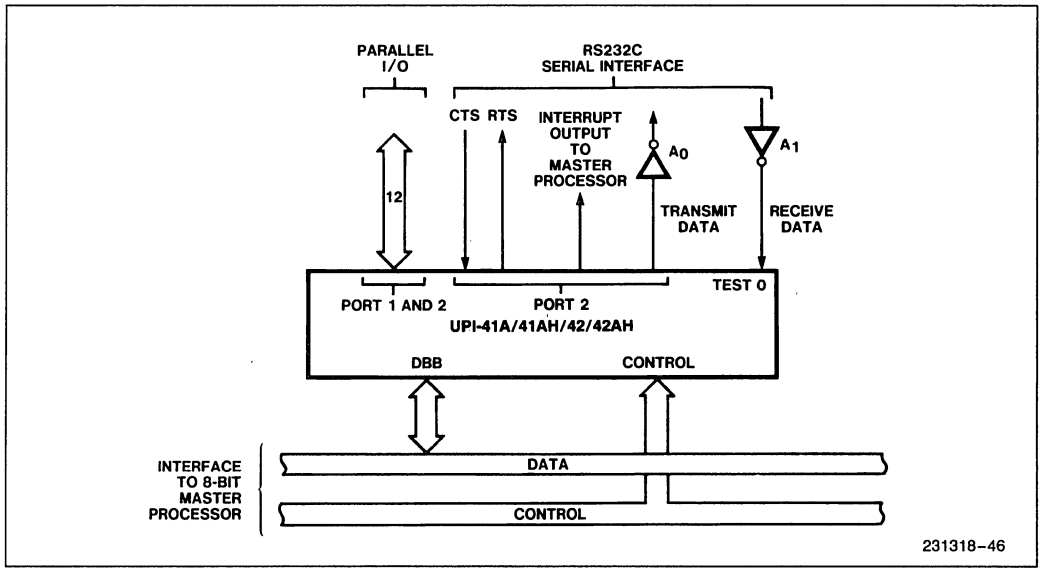


Figure 6-4. Universal I/O Interface

UPI-41AH/42AH
UNIVERSAL PERIPHERAL INTERFACE
8-BIT SLAVE MICROCONTROLLER

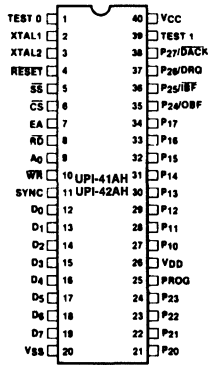
- **UPI-41: 6 MHz; UPI-42: 12.5 MHz**
- **Pin, Software and Architecturally Compatible with all UPI-41 and UPI-42 Products**
- **8-Bit CPU plus ROM/OTP EPROM, RAM, I/O, Timer/Counter and Clock in a Single Package**
- **2048 x 8 ROM/OTP, 256 x 8 RAM on UPI-42, 1024 x 8 ROM/OTP, 128 x 8 RAM on UPI-41, 8-Bit Timer/Counter, 18 Programmable I/O Pins**
- **One 8-Bit Status and Two Data Registers for Asynchronous Slave-to-Master Interface**
- **DMA, Interrupt, or Polled Operation Supported**
- **Fully Compatible with all Intel and Most Other Microprocessor Families**
- **Interchangeable ROM and OTP EPROM Versions**
- **Expandable I/O**
- **Sync Mode Available**
- **Over 90 Instructions: 70% Single Byte**
- **Available in EXPRESS — Standard Temperature Range**
- **int$_e$ligent Programming Algorithm — Fast OTP Programming**
- **Available in 40-Lead Plastic and 44-Lead Plastic Leaded Chip Carrier Packages**

(See Packaging Spec., Order #240800-001)
Package Type P and N

The Intel UPI-41AH and UPI-42AH are general-purpose Universal Peripheral Interfaces that allow the designer to develop customized solutions for peripheral device control.

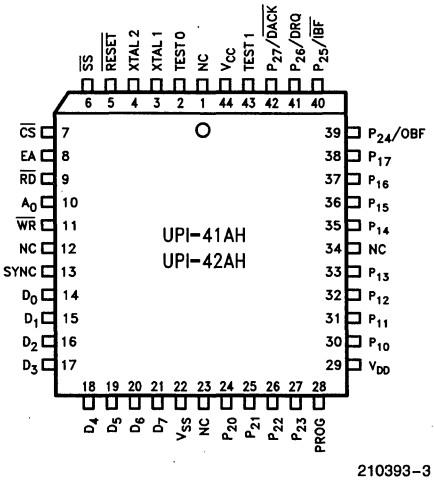
They are essentially "slave" microcontrollers, or microcontrollers with a slave interface included on the chip. Interface registers are included to enable the UPI device to function as a slave peripheral controller in the MCS Modules and iAPX family, as well as other 8-, 16-, and 32-bit systems.

To allow full user flexibility, the program memory is available in ROM and One-Time Programmable EPROM (OTP). All UPI-41AH and UPI-42AH devices are fully pin compatible for easy transition from prototype to production level designs.



210393-2

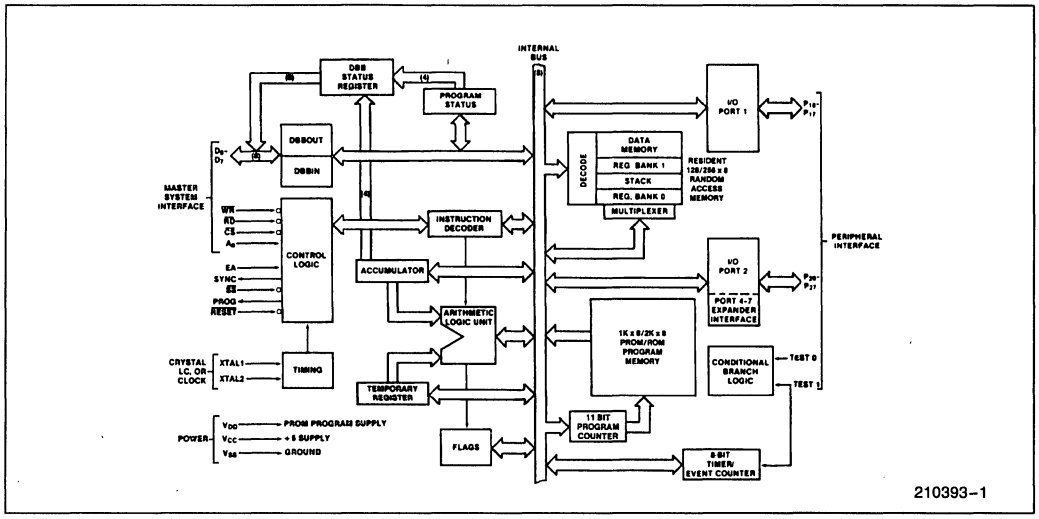
Figure 1. DIP Pin Configuration



210393-3

Figure 2. PLCC Pin Configuration

November 1994
Order Number: 210393-008



210393-1

Figure 3. Block Diagram

UPI PRODUCT MATRIX

UPI Device	ROM	OTP EPROM	RAM	Programming Voltage
8042AH	2K	—	256	—
8242AH	2K	—	256	—
8742AH	—	2K	256	12.5V
8041AH	1K	—	128	—
8741AH	—	1K	128	12.5V

4

THE INTEL 8242

As shown in the UPI-42 product matrix, the UPI-42 will be offered as a pre-programmed 8042 with several software vendors' keyboard controller firmware. The current list of available 8242 versions include keyboard controller firmware from both Phoenix Technologies Ltd., IBM, and Award Software Inc. The 8242 is programmed with Phoenix Technologies Ltd. keyboard controller firmware for AT-compatible systems. This keyboard controller is fully compatible with all AT-compatible operating systems and applications. The 8242PC also contains Phoenix Technologies Ltd. firmware. This keyboard controller provides support for AT, PS/2 and most EISA platforms as well as PS/2-style mouse support for either AT or PS/2 platforms.

The Intel 8242BB is programmed with IBM's keyboard controller firmware. The 8242BB provides an off the shelf keyboard and auxiliary device controller for AT, PS/2, EISA, and PCI architectures.

The 8242WA contains Award Software Inc. firmware. This device provides at AT-compatible keyboard controller for use in IBM PC AT compatible computers. The 8242WB contains a version of Award Software Inc. firmware that provides PS/2 style mouse support in addition to the standard features of the 8242WA.

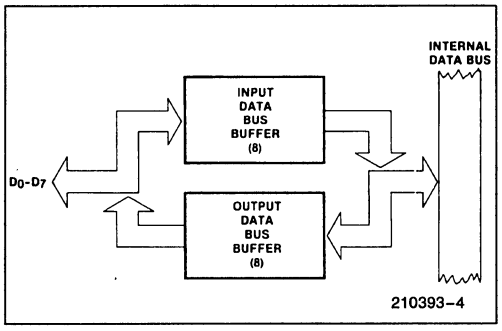
*Contact factory for current code revision available in all versions of the 8242 product lines.

Table 1. Pin Description

Symbol	DIP Pin No.	PLCC Pin No.	Type	Name and Function
TEST 0, TEST 1	1 39	2 43	I	**TEST INPUTS:** Input pins which can be directly tested using conditional branch instructions. **FREQUENCY REFERENCE:** TEST 1 (T_1) also functions as the event timer input (under software control). TEST 0 (T_0) is used during PROM programming and ROM/EPROM verification. It is also used during Sync Mode to reset the instruction state to S1 and synchronize the internal clock to PH1. See the Sync Mode Section.
XTAL 1, XTAL 2	2 3	3 4	I	**INPUTS:** Inputs for a crystal, LC or an external timing signal to determine the internal oscillator frequency.
$\overline{\text{RESET}}$	4	5	I	**RESET:** Input used to reset status flip-flops and to set the program counter to zero. $\overline{\text{RESET}}$ is also used during EPROM programming and verification.
$\overline{\text{SS}}$	5	6	I	**SINGLE STEP:** Single step input used in conjunction with the SYNC output to step the program through each instruction (EPROM). This should be tied to +5V when not used. This pin is also used to put the device in Sync Mode by applying 12.5V to it.
$\overline{\text{CS}}$	6	7	I	**CHIP SELECT:** Chip select input used to select one UPI microcomputer out of several connected to a common data bus.
EA	7	8	I	**EXTERNAL ACCESS:** External access input which allows emulation, testing and ROM/EPROM verification. This pin should be tied low if unused.
$\overline{\text{RD}}$	8	9	I	**READ:** I/O read input which enables the master CPU to read data and status words from the OUTPUT DATA BUS BUFFER or status register.
A_0	9	10	I	**COMMAND/DATA SELECT:** Address Input used by the master processor to indicate whether byte transfer is data ($A_0 = 0$, F1 is reset) or command ($A_0 = 1$, F1 is set). $A_0 = 0$ during program and verify operations.
$\overline{\text{WR}}$	10	11	I	**WRITE:** I/O write input which enables the master CPU to write data and command words to the UPI INPUT DATA BUS BUFFER.
SYNC	11	13	O	**OUTPUT CLOCK:** Output signal which occurs once per UPI instruction cycle. SYNC can be used as a strobe for external circuitry; it is also used to synchronize single step operation.
D_0–D_7 (BUS)	12–19	14–21	I/O	**DATA BUS:** Three-state, bidirectional DATA BUS BUFFER lines used to interface the UPI microcomputer to an 8-bit master system data bus.
P_{10}–P_{17}	27–34	30–33 35–38	I/O	**PORT 1:** 8-bit, PORT 1 quasi-bidirectional I/O lines. P_{10}–P_{17} access the signature row and security bit.
P_{20}–P_{27}	21–24 35–38	24–27 39–42	I/O	**PORT 2:** 8-bit, PORT 2 quasi-bidirectional I/O lines. The lower 4 bits (P_{20}–P_{23}) interface directly to the 8243 I/O expander device and contain address and data information during PORT 4–7 access. The upper 4 bits (P_{24}–P_{27}) can be programmed to provide interrupt Request and DMA Handshake capability. Software control can configure P_{24} as Output Buffer Full (OBF) interrupt, P_{25} as Input Buffer Full ($\overline{\text{IBF}}$) interrupt, P_{26} as DMA Request (DRQ), and P_{27} as DMA ACKnowledge ($\overline{\text{DACK}}$).
PROG	25	28	I/O	**PROGRAM:** Multifunction pin used as the program pulse input during PROM programming. During I/O expander access the PROG pin acts as an address/data strobe to the 8243. This pin should be tied high if unused.
V_{CC}	40	44		**POWER:** +5V main power supply pin.
V_{DD}	26	29		**POWER:** +5V during normal operation. +12.5V during programming operation. Low power standby supply pin.
V_{SS}	20	22		**GROUND:** Circuit ground potential.

UPI-41AH and UPI-42AH FEATURES

1. Two Data Bus Buffers, one for input and one for output. This allows a much cleaner Master/Slave protocol.



2. 8 Bits of Status

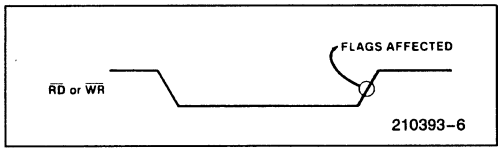
ST_7	ST_6	ST_5	ST_4	F_1	F_0	IBF	OBF
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

ST_4–ST_7 are user definable status bits. These bits are defined by the "MOV STS, A" single byte, single cycle instruction. Bits 4–7 of the acccumulator are moved to bits 4–7 of the status register. Bits 0–3 of the status register are not affected.

MOV STS, A Op Code: 90H

1	0	0	1	0	0	0	0
D_7							D_0

3. $\overline{RD}$ and $\overline{WR}$ are edge triggered. IBF, OBF, F_1 and INT change internally after the trailing edge of $\overline{RD}$ or $\overline{WR}$.

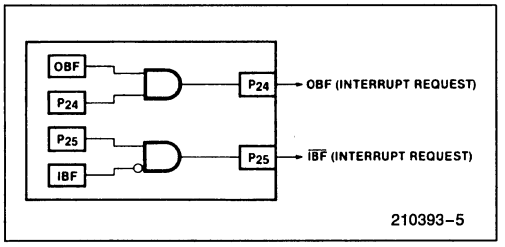


During the time that the host CPU is reading the status register, the UPI is prevented from updating this register or is 'locked out.'

4. P_{24} and P_{25} are port pins or Buffer Flag pins which can be used to interrupt a master processor. These pins default to port pins on Reset.

   If the "EN FLAGS" instruction has been executed, P_{24} becomes the OBF (Output Buffer Full) pin. A "1" written to P_{24} enables the OBF pin (the pin outputs the OBF Status Bit). A "0" written to P_{24} disables the OBF pin (the pin remains low). This pin can be used to indicate that valid data is available from the UPI (in Output Data Bus Buffer).

   If "EN FLAGS" has been executed, P_{25} becomes the $\overline{IBF}$ (Input Buffer Full) pin. A "1" written to P_{25} enables the $\overline{IBF}$ pin (the pin outputs the inverse of the IBF Status Bit. A "0" written to P_{25} disables the $\overline{IBF}$ pin (the pin remains low). This pin can be used to indicate that the UPI is ready for data.



Data Bus Buffer Interrupt Capability

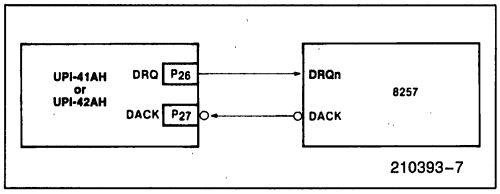
EN FLAGS Op Code: 0F5H

1	1	1	1	0	1	0	1
D_7							D_0

5. P_{26} and P_{27} are port pins or DMA handshake pins for use with a DMA controller. These pins default to port pins on Reset.

   If the "EN DMA" instruction has been executed, P_{26} becomes the DRQ (DMA Request) pin. A "1" written to P_{26} causes a DMA request (DRQ is activated). DRQ is deactivated by DACK•RD, DACK•WR, or execution of the "EN DMA" instruction.

   If "EN DMA" has been executed, P_{27} becomes the $\overline{DACK}$ (DMA ACKnowledge) pin. This pin acts as a chip select input for the Data Bus Buffer registers during DMA transfers.



DMA Handshake Capability

EN DMA Op Code: 0E5H

1	1	1	0	0	1	0	1

D_7 D_0

6. When EA is enabled on the UPI, the program counter is placed on Port 1 and the lower three bits of Port 2 (MSB = P_{22}, LSB = P_{10}). On the UPI this information is multiplexed with PORT DATA (see port timing diagrams at end of this data sheet).
7. The 8741AH and 8742AH support the intelligent Programming Algorithm. (See the Programming Section.)

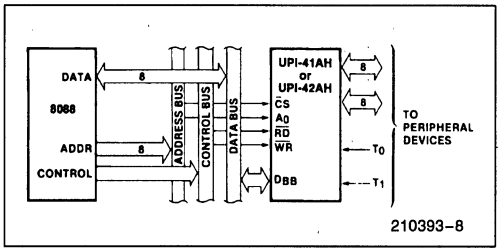


Figure 5. 8088-UPI-41AH/42AH Interface

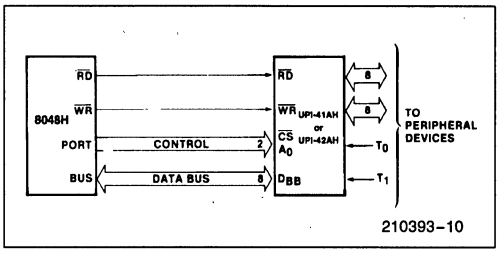


Figure 6. 8048H-UPI-41/42 Interface

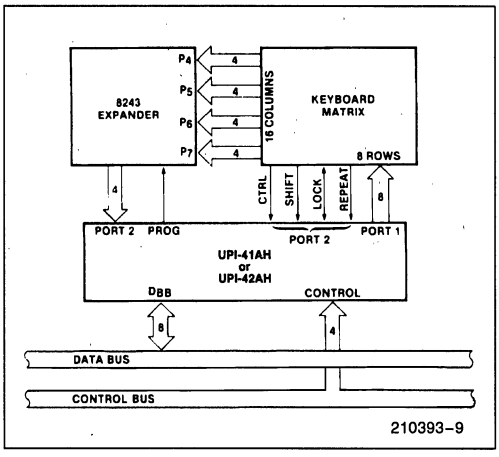


Figure 7. UPI-41/42-8243 Keyboard Scanner

APPLICATIONS

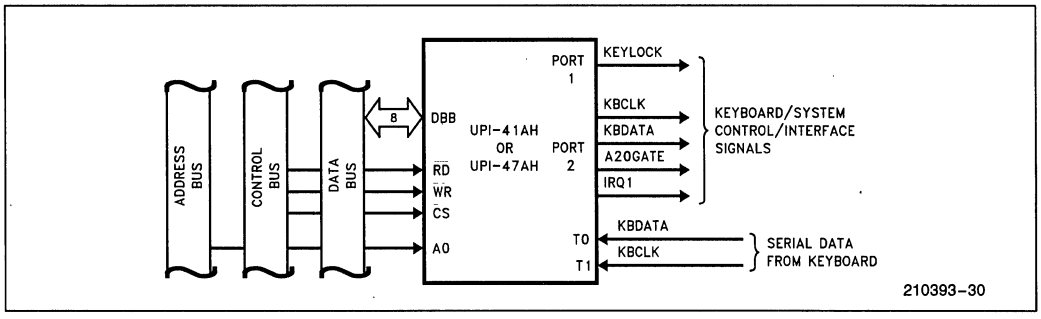


Figure 4. UPI-41AH/42AH Keyboard Controller

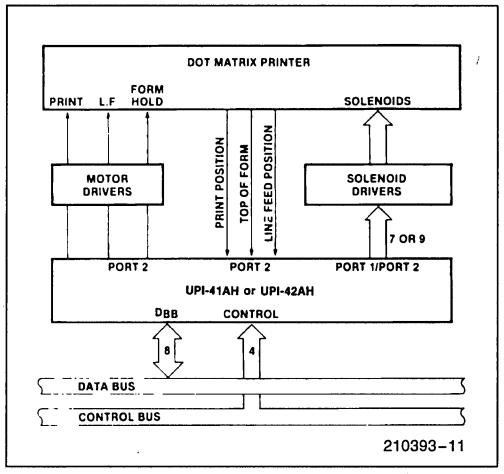


Figure 8. UPI-41AH/42AH 80-Column Matrix Printer Interface

PROGRAMMING AND VERIFYING THE 8741AH AND 8742AH OTP EPROM

Programming Verification

In brief, the programming process consists of: activating the program mode, applying an address, latching the address, applying data, and applying a programming pulse. Each word is programmed completely before moving on to the next and is followed by a verification step. The following is a list of the pins used for programming and a description of their functions:

Pin	Function
XTAL 1	2 Clock Inputs
$\overline{\text{Reset}}$	Initialization and Address Latching
Test 0	Selection of Program or Verify Mode
EA	Activation of Program/Verify Signature Row/Security Bit Modes
BUS	Address and Data Input Data Output During Verify
P_{20-22}	Address Input
V_{DD}	Programming Power Supply
PROG	Program Pulse Input

WARNING
An attempt to program a missocketed 8741AH or 8742AH will result in severe damage to the part. An indication of a properly socketed part is the appearance of the SYNC clock output. The lack of this clock may be used to disable the programmer.

The Program/Verify sequence is:

1. CS = 5V, V_{CC} = 5V, V_{DD} = 5V, RESET = 0V, A_0 = 0V, TEST 0 = 5V, clock applied or internal oscillator operating, BUS floating, PROG = 5V.
2. Insert 8741AH or 8742AH in programming socket
3. TEST 0 = 0V (select program mode)
4. EA = 12.5V (active program mode)
5. V_{CC} = 6V (programming supply)
6. V_{DD} = 12.5V (programming power)
7. Address applied to BUS and P_{20-22}
8. $\overline{\text{RESET}}$ = 5V (latch address)
9. Data applied to BUS
10. PROG = 5V followed by one 1 ms pulse to 0V
11. TEST 0 = 5V (verify mode)
12. Read and verify data on BUS
13. TEST 0 = 0V
14. Apply overprogram pulse
15. $\overline{\text{RESET}}$ = 0V and repeat from step 6
16. Programmer should be at conditions of step 1 when 8741AH or 8742AH is removed from socket

Please follow the int$_e$ligent Programming flow chart for proper programming procedure.

4

int$_e$ligent Programming Algorithm

The int$_e$ligent Programming Algorithm rapidly programs Intel 8741AH/8742AH EPROMs using an efficient and reliable method particularly suited to the production programming environment. Typical programming time for individual devices is on the order of 10 seconds. Programming reliability is also ensured as the incremental program margin of each byte is continually monitored to determine when it has been successfully programmed. A flowchart of the 8741AH/8742AH int$_e$ligent Programming Algorithm is shown in Figure 9.

The int$_e$ligent Programming Algorithm utilizes two different pulse types: initial and overprogram. The duration of the initial PROG pulse(s) is one millisecond, which will then be followed by a longer overprogram pulse of length 3X msec. X is an iteration counter and is equal to the number of the initial one millisecond pulses applied to a particular 8741AH/8742AH location, before a correct verify occurs. Up to 25 one-millisecond pulses per byte are provided for before the overprogram pulse is applied.

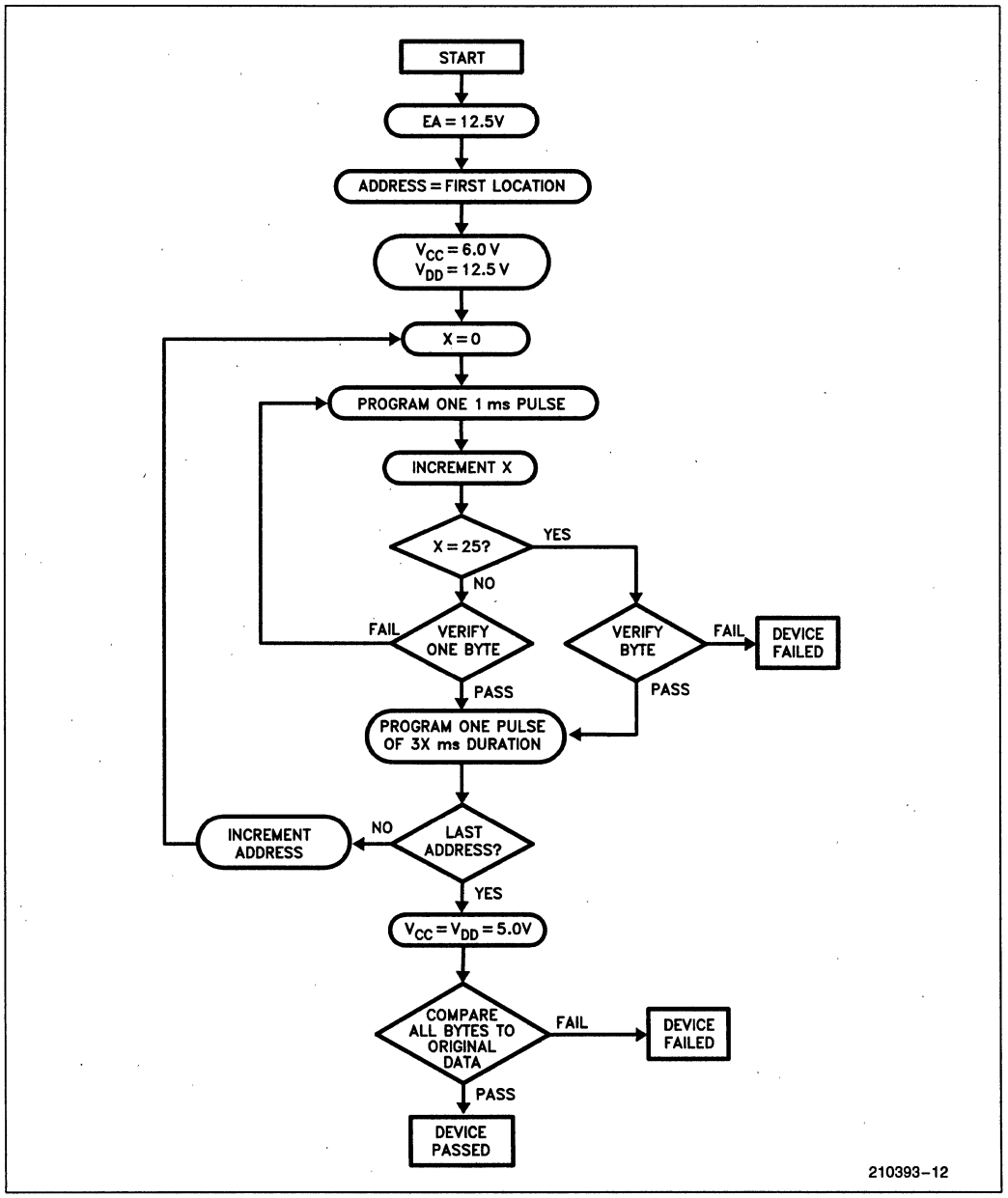


Figure 9. Programming Algorithm

The entire sequence of program pulses and byte verifications is performed at V_{CC} = 6.0V and V_{DD} = 12.5V. When the int$_e$ligent Programming cycle has been completed, all bytes should be compared to the original data with V_{CC} = 5.0, V_{DD} = 5V.

Verify

A verify should be performed on the programmed bits to determine that they have been correctly programmed. The verify is performed with T0 = 5V, V_{DD} = 5V, EA = 12.5V, $\overline{SS}$ = 5V, PROG = 5V, A0 = 0V, and $\overline{CS}$ = 5V.

SECURITY BIT

The security bit is a single EPROM cell outside the EPROM array. The user can program this bit with the appropriate access code and the normal programming procedure, to inhibit any external access to the EPROM contents. Thus the user's resident program is protected. There is no direct external access to this bit. However, the security byte in the signature row has the same address and can be used to check indirectly whether the security bit has been programmed or not. The security bit has no effect on the signature mode, so the security byte can always be examined.

SECURITY BIT PROGRAMMING/ VERIFICATION

Programming

a. Read the security byte of the signature mode. Make sure it is 00H.

b. Apply access code to appropriate inputs to put the device into security mode.

c. Apply high voltage to EA and V_{DD} pins.

d. Follow the programming procedure as per the int$_e$ligent Programming Algorithm with known data on the databus. Not only the security bit, but also the security byte of the signature row is programmed.

e. Verify that the security byte of the signature mode contains the same data as appeared on the data bus. (If DB0–DB7 = high, the security byte will contain FFH.)

f. Read two consecutive known bytes from the EPROM array and verify that the wrong data are retrieved in at least one verification. If the EPROM can still be read, the security bit may have not been fully programmed though the security byte in the signature mode has.

Verification

Since the security bit address overlaps the address of the security byte of the signature mode, it can be used to check indirectly whether the security bit has been programmed or not. Therefore, the security bit verification is a mere read operation of the security byte of the signature row (0FFH = security bit programmed; 00H = security bit unprogrammed). Note that during the security bit programming, the reading of the security byte does not necessarily indicate that the security bit has been successfully programmed. Thus, it is recommended that two consecutive known bytes in the EPROM array be read and the wrong data should be read at least once, because it is highly improbable that random data coincides with the correct ones twice.

SIGNATURE MODE

The UPI-41AH/42AH has an additional 32 bytes of EPROM available for Intel and user signatures and miscellaneous purposes. The 32 bytes are partitioned as follows:

A. **Test code/checksum**—This can accommodate up to 25 bytes of code for testing the internal nodes that are not testable by executing from the external memory. The test code/checksum is present on ROMs, and OTPs.

B. **Intel signature**—This allows the programmer to read from the UPI-41AH/42AH the manufacturer of the device and the exact product name. It facilitates automatic device identification and will be present in the ROM and OTP versions. Location 10H contains the manufacturer code. For Intel, it is 89H. Location 11H contains the device code. The code is 43H and 42H for the 8042AH and OTP 8742AH, and 41H and 40H for the 8041AH and OTP 8741AH, respectively. The code is 44H for any device with the security bit set by Intel.

C. **User signature**—The user signature memory is implemented in the EPROM and consists of 2 bytes for the customer to program his own signature code (for identification purposes and quick sorting of previously programmed materials).

D. **Test signature**—This memory is used to store testing information such as: test data, bin number, etc. (for use in quality and manufacturing control).

E. **Security byte**—This byte is used to check whether the security bit has been programmed (see the security bit section).

The signature mode can be accessed by setting P10 = 0, P11–P17 = 1, and then following the programming and/or verification procedures. The location of the various address partitions are as follows:

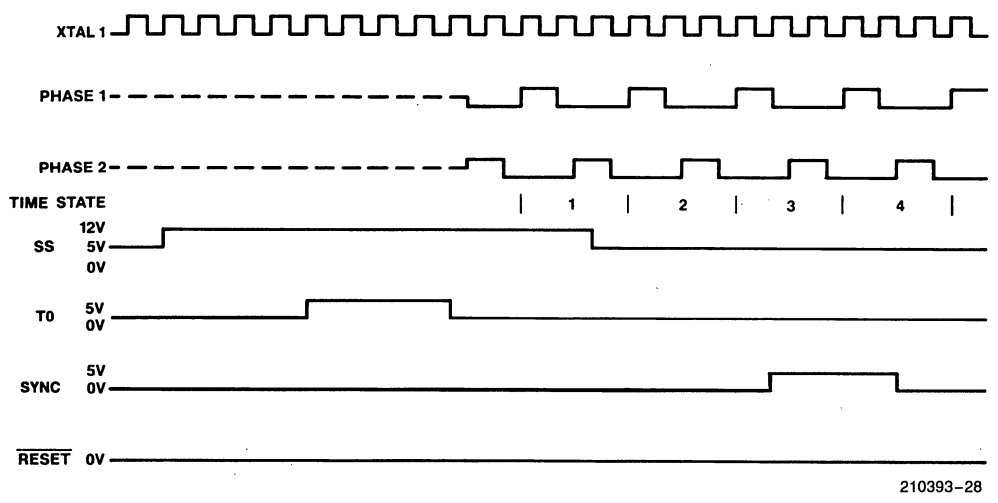
	Address		Device Type	No. of Bytes
Test Code/Checksum	0 16H	0FH 1EH	ROM/OTP	25
Intel Signature	10H	11H	ROM/OTP	2
User Signature	12H	13H	OTP	2
Test Signature	14H	15H	ROM/OTP	2
Security Byte	1FH		OTP	1

SYNC MODE

The Sync Mode is provided to ease the design of multiple controller circuits by allowing the designer to force the device into known phase and state time. The Sync Mode may also be utilized by automatic test equipment (ATE) for quick, easy, and efficient synchronizing between the tester and the DUT (device under test).

Sync Mode is enabled when $\overline{SS}$ pin is raised to high voltage level of +12 volts. To begin synchronization, T0 is raised to 5 volts at least four clock cycles after $\overline{SS}$. T0 must be high for at least four X1 clock cycles to fully reset the prescaler and time state generators. T0 may then be brought down during low state of X1. Two clock cycles later, with the rising edge of X1, the device enters into Time State 1, Phase 1. $\overline{SS}$ is then brought down to 5 volts 4 clocks later after T0. RESET is allowed to go high 5 tCY (75 clocks) later for normal execution of code.

SYNC MODE TIMING DIAGRAMS



210393-28

Minimum Specifications
SYNC Operation Time, t_{SYNC} = 3.5 XTAL 1 Clock cycles. Reset Time, t_{RS} = 4 t_{CY}.

NOTE:
The rising and falling edges of T0 should occur during low state of XTAL1 clock.

4

ACCESS CODE

The following table summarizes the access codes required to invoke the Sync Mode, Signature Mode, and the Security Bit, respectively. Also, the programming and verification modes are included for comparison.

Modes		Control Signals							Data Bus								Access Code								
																	Port 2			Port 1						
		T0	RST	SS	EA	PROG	V_{DD}	V_{CC}	0	1	2	3	4	5	6	7	0	1	2	0	1	2	3	4	5	6	7
Programming Mode		0	0	1	HV	1	V_{DDH}	V_{CC}	Address								Addr			a_0	a_1	X	X	X	X	X	X
		0	1	1	HV	STB	V_{DDH}	V_{CC}	Data In								Addr										
Verification Mode		0	0	1	HV	1	V_{CC}	V_{CC}	Address								Addr			a_0	a_1	X	X	X	X	X	X
		1	1	1	HV	1	V_{CC}	V_{CC}	Data Out								Addr										
Sync Mode		STB High	0	HV	0	X	V_{CC}	V_{CC}	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Signature Mode	Prog	0	0	1	HV	1	V_{DDH}	V_{CC}	Addr. (see Sig Mode Table)								0	0	0	0	1	1	1	1	X	X	1
		0	1	1	HV	STB	V_{DDH}	V_{CC}	Data In								0	0	0								
	Verify	0	0	1	HV	1	V_{CC}	V_{CC}	Addr. (see Sig Mode Table)								0	0	0								
		1	1	1	HV	1	V_{CC}	V_{CC}	Data Out								0	0	0								
Security Bit/Byte	Prog	0	0	1	HV	1	V_{DDH}	V_{CC}	Address								0	0	0								
		0	1	1	HV	STB	V_{DDH}	V_{CC}	Data In								0	0	0								
	Verify	0	0	1	HV	1	V_{CC}	V_{CC}	Address								0	0	0								
		1	1	1	HV	1	V_{CC}	V_{CC}	Data Out								0	0	0								

NOTES:
1. a_0 = 0 or 1; a_1 = 0 or 1. a_0 must = a_1.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ....0°C to +70°C
Storage Temperature .....−65°C to +150°C
Voltage on Any Pin with Respect to Ground.....−0.5V to +7V
Power Dissipation .....1.5 W

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**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.*

D.C. CHARACTERISTICS T_A = 0°C to +70°C, V_{CC} = V_{DD} = +5V ±10%

Symbol	Parameter	UPI-41AH/42AH		Units	Notes
		Min	Max		
V_{IL}	Input Low Voltage (Except XTAL1, XTAL2, RESET)	−0.5	0.8	V	
V_{IL1}	Input Low Voltage (XTAL1, XTAL2, RESET)	−0.5	0.6	V	
V_{IH}	Input High Voltage (Except XTAL1, XTAL2, RESET)	2.0	V_{CC}	V	
V_{IH1}	Input High Voltage (XTAL1, RESET)	3.5	V_{CC}	V	
V_{IH2}	Input High Voltage (XTAL2)	2.2	V_{CC}	V	
V_{OL}	Output Low Voltage (D_0–D_7)		0.45	V	I_{OL} = 2.0 mA

D.C. CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = V_{DD} = +5V \pm 10\%$ (Continued)

Symbol	Parameter	UPI-41AH/42AH		Units	Notes
		Min	Max		
V_{OL1}	Output Low Voltage ($P_{10}P_{17}$, $P_{20}P_{27}$, Sync)		0.45	V	$I_{OL} = 1.6$ mA
V_{OL2}	Output Low Voltage (PROG)		0.45	V	$I_{OL} = 1.0$ mA
V_{OH}	Output High Voltage (D_0–D_7)	2.4		V	$I_{OH} = -400\ \mu A$
V_{OH1}	Output High Voltage (All Other Outputs)	2.4			$I_{OH} = -50\ \mu A$
I_{IL}	Input Leakage Current (T_0, T_1, RD, WR, CS, A_0, EA)		±10	μA	$V_{SS} \le V_{IN} \le V_{CC}$
I_{OFL}	Output Leakage Current (D_0–D_7, High Z State)		±10	μA	$V_{SS} + 0.45 \le V_{OUT} \le V_{CC}$
I_{LI}	Low Input Load Current ($P_{10}P_{17}$, $P_{20}P_{27}$)		0.3	mA	$V_{IL} = 0.8V$
I_{LI1}	Low Input Load Current (RESET, SS)		0.2	mA	$V_{IL} = 0.8V$
I_{DD}	V_{DD} Supply Current		20	mA	Typical = 8 mA
$I_{CC} + I_{DD}$	Total Supply Current		135	mA	Typical = 80 mA
I_{DD} Standby	Power Down Supply Current		20	mA	Typical = 8 mA
I_{IH}	Input Leakage Current (P_{10}–P_{17}, P_{20}–P_{27})		100	μA	$V_{IN} = V_{CC}$
C_{IN}	Input Capacitance		10	pF	$T_A = 25°C$ (1)
C_{IO}	I/O Capacitance		20	pF	$T_A = 25°C$ (1)

NOTE:
1. Sampled, not 100% tested.

4

D.C. CHARACTERISTICS—PROGRAMMING

$T_A = 25°C \pm 5°C$, $V_{CC} = 6V \pm 0.25V$, $V_{DD} = 12.5V \pm 0.5V$

Symbol	Parameter	Min	Max	Units
V_{DDH}	V_{DD} Program Voltage High Level	12	13	V(1)
V_{DDL}	V_{DD} Voltage Low Level	4.75	5.25	V
V_{PH}	PROG Program Voltage High Level	2.0	5.5	V
V_{PL}	PROG Voltage Low Level	−0.5	0.8	V
V_{EAH}	Input High Voltage for EA	12.0	13.0	V(2)
V_{EAL}	EA Voltage Low Level	−0.5	5.25	V
I_{DD}	V_{DD} High Voltage Supply Current		50.0	mA
I_{EA}	EA High Voltage Supply Current		1.0	mA

NOTES:
1. Voltages over 13V applied to pin V_{DD} will permanently damage the device.
2. V_{EAH} must be applied to EA before V_{DDH} and removed after V_{DDL}.
3. V_{CC} must be applied simultaneously or before V_{DD} and must be removed simultaneously or after V_{DD}.

A.C. CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{SS} = 0V$, $V_{CC} = V_{DD} = +5V \pm 10\%$

DBB READ

Symbol	Parameter	Min	Max	Units
t_{AR}	CS, A_0 Setup to RD ↓	0		ns
t_{RA}	CS, A_0 Hold After RD ↑	0		ns
t_{RR}	RD Pulse Width	160		ns
t_{AD}	CS, A_0 to Data Out Delay		130	ns
t_{RD}	RD ↓ to Data Out Delay	0	130	ns
t_{DF}	RD ↑ to Data Float Delay		85	ns

DBB WRITE

Symbol	Parameter	Min	Max	Units
t_{AW}	CS, A_0 Setup to WR ↓	0		ns
t_{WA}	CS, A_0 Hold After WR ↑	0		ns
t_{WW}	WR Pulse Width	160		ns
t_{DW}	Data Setup to WR ↑	130		ns
t_{WD}	Data Hold After WR ↑	0		ns

CLOCK

Symbol	Parameter	Min	Max	Units
t_{CY} (UPI-41AH/42AH)	Cycle Time	1.2	9.20	μs(1)
t_{CYC} (UPI-41AH/42AH)	Clock Period	80	613	ns
t_{PWH}	Clock High Time	30		ns
t_{PWL}	Clock Low Time	30		ns
t_R	Clock Rise Time		10	ns
t_F	Clock Fall Time		10	ns

NOTE:
1. $t_{CY} = 15/f(XTAL)$

A.C. CHARACTERISTICS DMA

Symbol	Parameter	Min	Max	Units
t_{ACC}	DACK to WR or RD	0		ns
t_{CAC}	RD or WR to DACK	0		ns
t_{ACD}	DACK to Data Valid	0	130	ns
t_{CRQ}	RD or WR to DRQ Cleared		110	ns(1)

NOTE:
1. $C_L = 150$ pF.

A.C. CHARACTERISTICS—PROGRAMMING

T_A = 25°C ±5°C, V_{CC} = 6V ±0.25V, V_{DDL} = +5V ±0.25V, V_{DDH} = 12.5V ±0.5V
(8741AH/8742AH ONLY)

Symbol	Parameter	Min	Max	Units
t_{AW}	Address Setup Time to RESET ↑	$4t_{CY}$		
t_{WA}	Address Hold Time After RESET ↑	$4t_{CY}$		
t_{DW}	Data in Setup Time to PROG ↓	$4t_{CY}$		
t_{WD}	Data in Hold Time After PROG ↑	$4t_{CY}$		
t_{PW}	Initial Program Pulse Width	0.95	1.05	ms(1)
t_{TW}	Test 0 Setup Time for Program Mode	$4t_{CY}$		
t_{WT}	Test 0 Hold Time After Program Mode	$4t_{CY}$		
t_{DO}	Test 0 to Data Out Delay		$4t_{CY}$	
t_{WW}	RESET Pulse Width to Latch Address	$4t_{CY}$		
t_r, t_f	PROG Rise and Fall Times	0.5	100	μs
t_{CY}	CPU Operation Cycle Time	2.5	3.75	μs
t_{RE}	RESET Setup Time Before EA ↑	$4t_{CY}$		
t_{OPW}	Overprogram Pulse Width	2.85	78.75	ms(2)
t_{DE}	EA High to V_{DD} High	$1t_{CY}$		

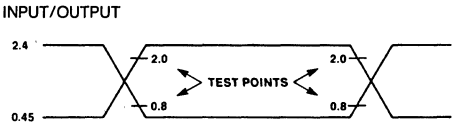
NOTES:
1. Typical Initial Program Pulse width tolerance = 1 ms ±5%.
2. This variation is a function of the iteration counter value, X.
3. If TEST 0 is high, t_{DO} can be triggered by RESET ↑.

A.C. CHARACTERISTICS PORT 2 T_A = 0°C to +70°C, V_{CC} = +5V ±10%

Symbol	Parameter	$f(t_{CY})$(3)	Min	Max	Units
t_{CP}	Port Control Setup Before Falling Edge of PROG	$1/15\ t_{CY}-28$	55		ns(1)
t_{PC}	Port Control Hold After Falling Edge of PROG	$1/10\ t_{CY}$	125		ns(2)
t_{PR}	PROG to Time P2 Input Must Be Valid	$8/15\ t_{CY}-16$		650	ns(1)
t_{PF}	Input Data Hold Time		0	150	ns(2)
t_{DP}	Output Data Setup Time	$2/10\ t_{CY}$	250		ns(1)
t_{PD}	Output Data Hold Time	$1/10\ t_{CY}-80$	45		ns(2)
t_{PP}	PROG Pulse Width	$6/10\ t_{CY}$	750		ns

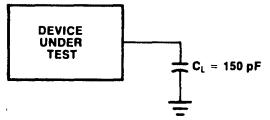
NOTES:
1. C_L = 80 pF.
2. C_L = 20 pF.
3. t_{CY} = 1.25 μs.

A.C. TESTING INPUT/OUTPUT WAVEFORM



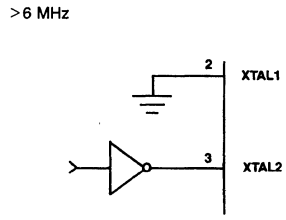
210393-14

A.C. TESTING LOAD CIRCUIT

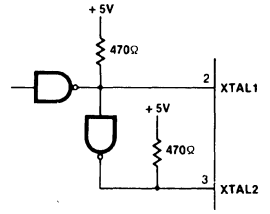


210393-15

DRIVING FROM EXTERNAL SOURCE-TWO OPTIONS



210393-16

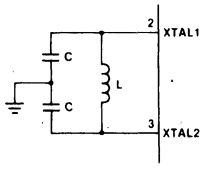


210393-17

Rise and Fall Times Should Not Exceed 10 ns. Resistors to V_{CC} are Needed to Ensure V_{IH} = 3.5V if TTL Circuitry is Used.

LC OSCILLATOR MODE

L	C	NOMINAL
45 H	20 pF	5.2 MHz
120 H	20 pF	3.2 MHz



$$f = \frac{1}{2\pi\sqrt{LC'}}$$

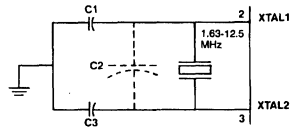
$$C' = \frac{C + 3Cpp}{2}$$

Cpp ≅ 5-10 pF
Pin-to-Pin Capacitance

210393-18

Each C Should be Approximately 20 pF, including Stray Capacitance.

CRYSTAL OSCILLATOR MODE



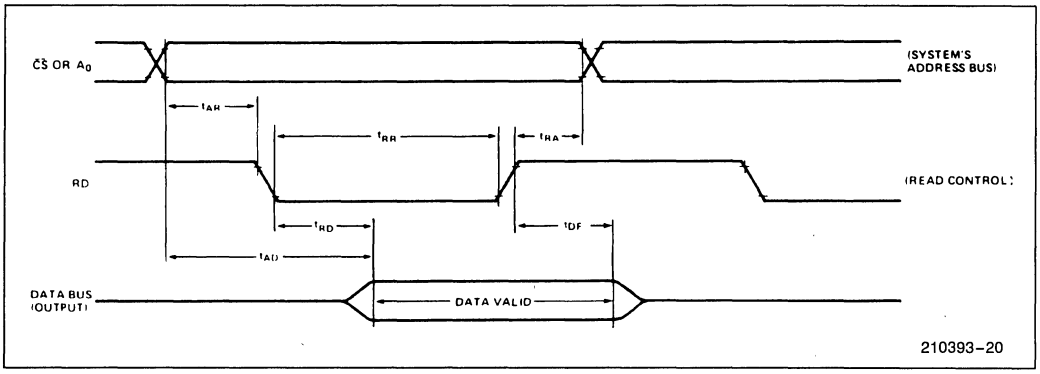
210393-19

C1 5 pF (STRAY 5 pF)
C2 (CRYSTAL + STRAY) 8 pF
C3 20-30 pF INCLUDING STRAY

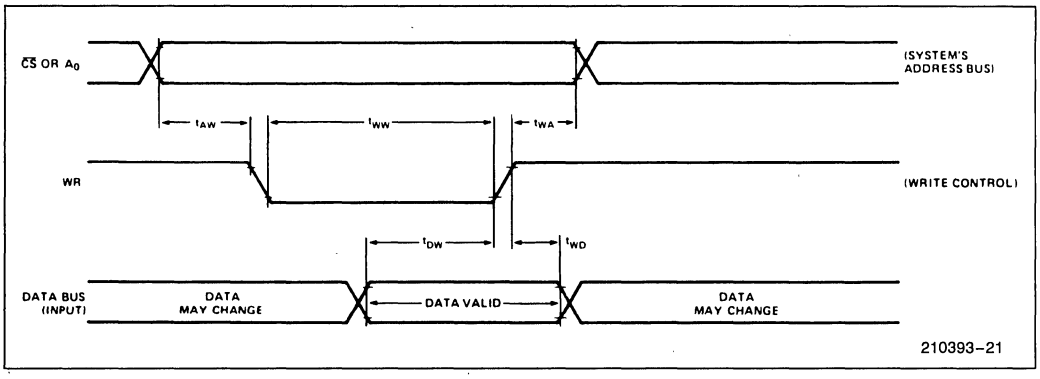
Crystal Series Resistance Should be Less Than 30Ω at 12.5 MHz.

WAVEFORMS

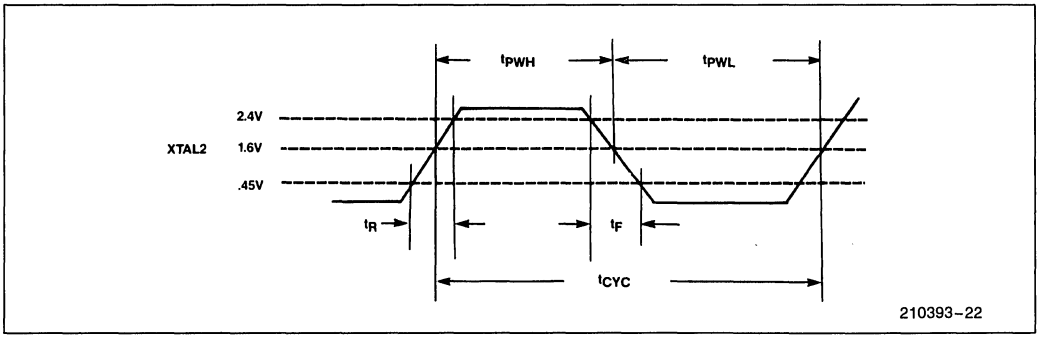
READ OPERATION—DATA BUS BUFFER REGISTER



WRITE OPERATION—DATA BUS BUFFER REGISTER

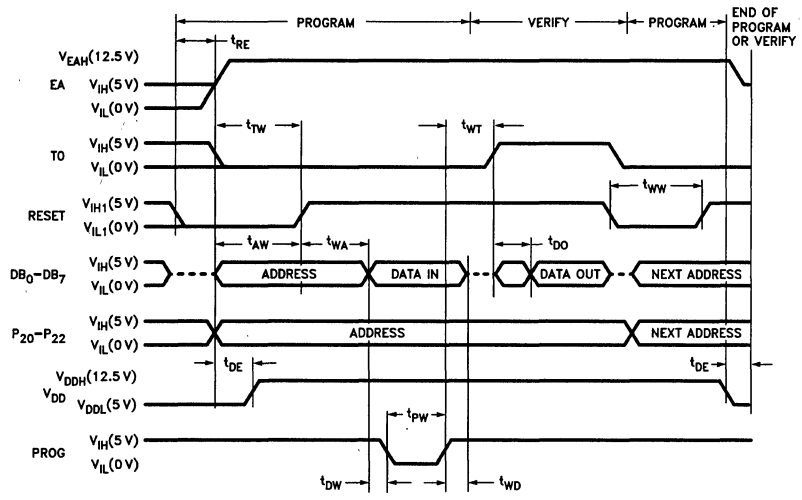


CLOCK TIMING



WAVEFORMS (Continued)

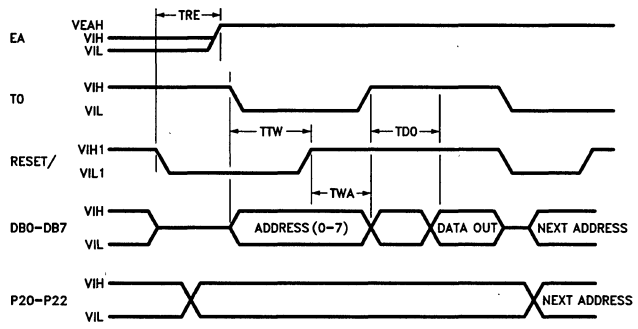
COMBINATION PROGRAM/VERIFY MODE



210393-23

NOTES:
1. A_0 must be held low (0V) during program/verify modes.
2. For V_{IH}, V_{IH1}, V_{IL}, V_{IL1}, V_{DDH}, and V_{DDL}, please consult the D.C. Characteristics Table.
3. When programming the 8741AH/8742AH, a 0.1 μF capacitor is required across V_{DD} and ground to suppress spurious voltage transients which can damage the device.

VERIFY MODE

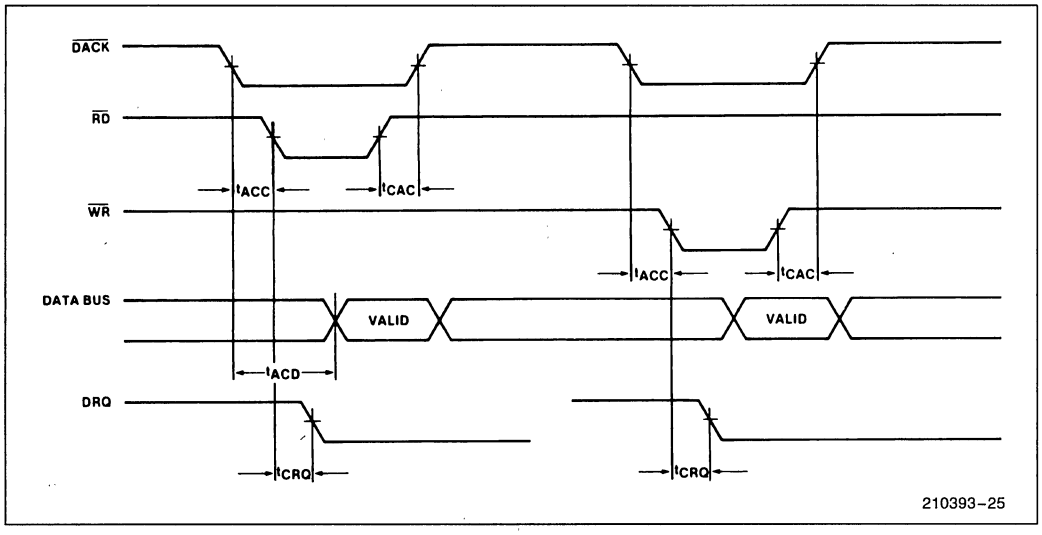


210393-29

NOTES:
1. PROG must float if EA is low.
2. PROG must float or = 5V when EA is high.
3. P_{10}–P_{17} = 5V or must float.
4. P_{24}–P_{27} = 5V or must float.
5. A_0 must be held low during programming/verify modes.

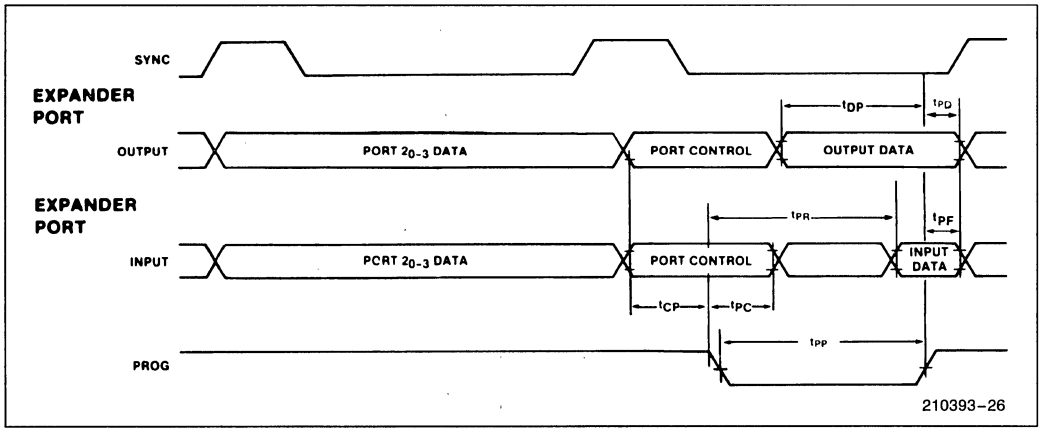
WAVEFORMS (Continued)

DMA



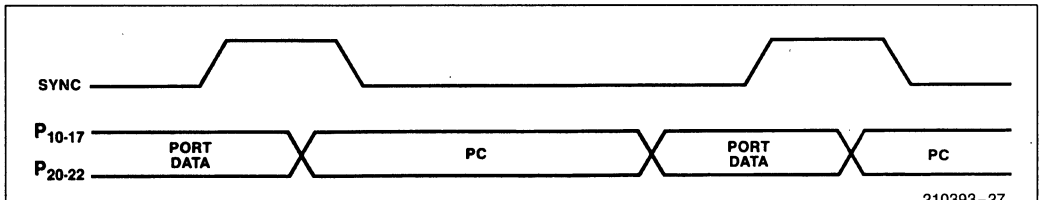
210393-25

PORT 2



210393-26

PORT TIMING DURING EXTERNAL ACCESS (EA)



210393-27

On the Rising Edge of SYNC and EA is Enabled, Port Data is Valid and can be Strobed. On the Trailing Edge of Sync the Program Counter Contents are Available.

Table 2. UPI Instruction Set

Mnemonic	Description	Bytes	Cycles
ACCUMULATOR			
ADD A, Rr	Add register to A	1	1
ADD A, @Rr	Add data memory to A	1	1
ADD A, #data	Add immediate to A	2	2
ADDC A, Rr	Add register to A with carry	1	1
ADDC A, @Rr	Add data memory to A with carry	1	1
ADDC A, #data	Add immediate to A with carry	2	2
ANL A, Rr	AND register to A	1	1
ANL, A @Rr	AND data memory to A	1	1
ANL A, #data	AND immediate to A	2	2
ORL A, Rr	OR register to A	1	1
ORL, A, @Rr	OR data memory to A	1	1
ORL A, #data	OR immediate to A	2	2
XRL A, Rr	Exclusive OR register to A	1	1
XRL A, @Rr	Exclusive OR data memory to A	1	1
XRL A, #data	Exclusive OR immediate to A	2	2
INC A	Increment A	1	1
DEC A	Decrement A	1	1
CLR A	Clear A	1	1
CPL A	Complement A	1	1
DA A	Decimal Adjust A	1	1
SWAP A	Swap nibbles of A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through carry	1	1
INPUT/OUTPUT			
IN A, Pp	Input port to A	1	2
OUTL Pp, A	Output A to port	1	2
ANL Pp, #data	AND immediate to port	2	2
ORL Pp, #data	OR immediate to port	2	2
IN A, DBB	Input DBB to A, clear IBF	1	1
OUT DBB, A	Output A to DBB, set OBF	1	1
MOV STS, A	A_4–A_7 to Bits 4–7 of Status	1	1
MOVD A, Pp	Input Expander port to A	1	2
MOVD Pp, A	Output A to Expander port	1	2
ANLD Pp, A	AND A to Expander port	1	2
ORLD Pp, A	OR A to Expander port	1	2

Mnemonic	Description	Bytes	Cycles
DATA MOVES			
MOV A, Rr	Move register to A	1	1
MOV A, @Rr	Move data memory to A	1	1
MOV A, #data	Move immediate to A	2	2
MOV Rr, A	Move A to register	1	1
MOV @Rr, A	Move A to data memory	1	1
MOV Rr, #data	Move immediate to register	2	2
MOV @Rr, #data	Move immediate to data memory	2	2
MOV A, PSW	Move PSW to A	1	1
MOV PSW, A	Move A to PSW	1	1
XCH A, Rr	Exchange A and register	1	1
XCH A, @Rr	Exchange A and data memory	1	1
XCHD A, @Rr	Exchange digit of A and register	1	1
MOVP A, @A	Move to A from current page	1	2
MOVP3, A, @A	Move to A from page 3	1	2
TIMER/COUNTER			
MOV A, T	Read Timer/Counter	1	1
MOV T, A	Load Timer/Counter	1	1
STRT T	Start Timer	1	1
STRT CNT	Start Counter	1	1
STOP TCNT	Stop Timer/Counter	1	1
EN TCNTI	Enable Timer/Counter Interrupt	1	1
DIS TCNTI	Disable Timer/Counter Interrupt	1	1
CONTROL			
EN DMA	Enable DMA Handshake Lines	1	1
EN I	Enable IBF Interrupt	1	1
DIS I	Diable IBF Interrupt	1	1
EN FLAGS	Enable Master Interrupts	1	1
SEL RB0	Select register bank 0	1	1
SEL RB1	Select register bank 1	1	1
NOP	No Operation	1	1
REGISTERS			
INC Rr	Increment register	1	1
INC @Rr	Increment data memory	1	1
DEC Rr	Decrement register	1	1

Table 2. UPI Instruction Set (Continued)

Mnemonic	Description	Bytes	Cycles
SUBROUTINE			
CALL addr	Jump to subroutine	2	2
RET	Return	1	2
RETR	Return and restore status	1	2
FLAGS			
CLR C	Clear Carry	1	1
CPL C	Complement Carry	1	1
CLR F0	Clear Flag 0	1	1
CPL F0	Complement Flag 0	1	1
CLR F1	Clear F1 Flag	1	1
CPL F1	Complement F1 Flag	1	1
BRANCH			
JMP addr	Jump unconditional	2	2
JMPP @A	Jump indirect	1	2
DJNZ Rr, addr	Decrement register and jump	2	2
JC addr	Jump on Carry = 1	2	2
JNC addr	Jump on Carry = 0	2	2
JZ addr	Jump on A Zero	2	2
JNZ addr	Jump on A not Zero	2	2
JT0 addr	Jump on T0 = 1	2	2
JNT0 addr	Jump on T0 = 0	2	2
JT1 addr	Jump on T1 = 1	2	2
JNT1 addr	Jump on T1 = 0	2	2
JF0 addr	Jump on F0 Flag = 1	2	2
JF1 addr	Jump on F1 Flag = 1	2	2
JTF addr	Jump on Timer Flag = 1, Clear Flag	2	2
JNIBF addr	Jump on IBF Flag = 0	2	2
JOBF addr	Jump on OBF Flag = 1	2	2
JBb addr	Jump on Accumula- for Bit	2	2

8741A
UNIVERSAL PERIPHERAL INTERFACE
8-BIT MICROCOMPUTER

- **8-Bit CPU plus ROM, RAM, I/O, Timer and Clock in a Single Package**
- **One 8-Bit Status and Two Data Registers for Asynchronous Slave-to-Master Interface**
- **DMA, Interrupt, or Polled Operation Supported**
- **1024 x 8 EPROM, 64 x 8 RAM, 8-Bit Timer/Counter, 18 Programmable I/O Pins**
- **Fully Compatible with All Microprocessor Families**
- **3.6 MHz 8741A-8 Available**
- **Expandable I/O**
- **RAM Power-Down Capability**
- **Over 90 Instructions: 70% Single Byte**
- **Available in EXPRESS**
  — Standard Temperature Range
  — Extended Temperature Range

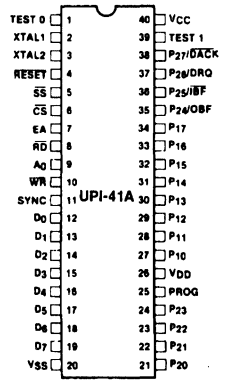
The Intel 8741A is a general purpose, programmable interface device designed for use with a variety of 8-bit microprocessor systems. It contains a low cost microcomputer with program memory, data memory, 8-bit CPU, I/O ports, timer/counter, and clock in a single 40-pin package. Interface registers are included to enable the UPI device to function as a peripheral controller in MCS®-48, MCS-80, MCS-85, MCS-86, and other 8-bit systems.

The UPI-41A has 1K words of program memory and 64 words of data memory on-chip.

The device has two 8-bit, TTL compatible I/O ports and two test inputs. Individual port lines can function as either inputs or outputs under software control. I/O can be expanded with the 8243 device which is directly compatible and has 16 I/O lines. An 8-bit programmable timer/counter is included in the UPI device for generating timing sequences or counting external inputs. Additional UPI features include: single 5V supply, single-step mode for debug and dual working register banks.

Because it's a complete microcomputer, the UPI provides more flexibility for the designer than conventional LSI interface devices. It is designed to be an efficient controller as well as an arithmetic processor. Applications include keyboard scanning, printer control, display multiplexing and similar functions which involve interfacing peripheral devices to microprocessor systems.

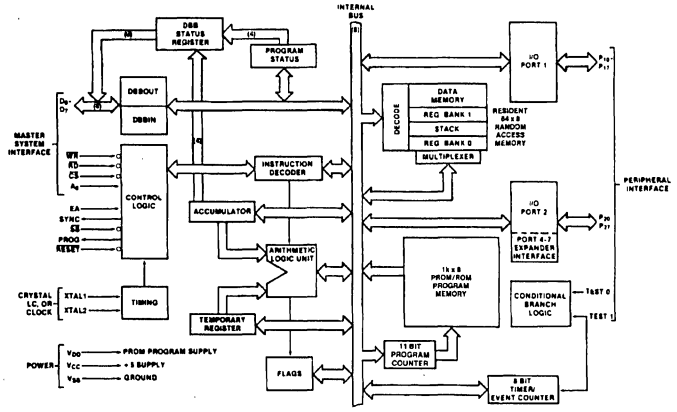
Pin Configuration



290241-2

October 1989
Order Number: 290241-001

Block Diagram



290241-1

Table 1. Pin Description

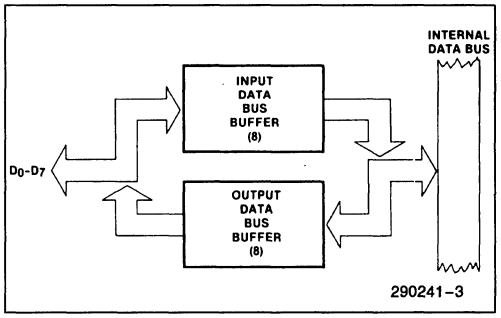
Signal	Description
D_0–D_7 (BUS)	Three-state, bidirectional DATA BUS BUFFER lines used to interface the UPI-41A to an 8-bit master system data bus.
P_{10}–P_{17}	8-bit, PORT 1 quasi-bidirectional I/O lines.
P_{20}–P_{27}	8-bit, PORT 2 quasi-bidirectional I/O lines. The lower 4 bits (P_{20}–P_{23}) interface directly to the 8243 I/O expander device and contain address and data information during PORT 4–7 access. The upper 4 bits (P_{24}–P_{27}) can be programmed to provide interrupt Request and DMA Handshake capability. Software control can configure P_{24} as OBF (Output Buffer Full), P_{25} as $\overline{IBF}$ (Input Buffer Full), P_{26} as DRQ (DMA Request), and P_{27} as $\overline{DACK}$ (DMA ACKnowledge).
$\overline{WR}$	I/O write input which enables the master CPU to write data and command words to the UPI-41A INPUT DATA BUS BUFFER.
$\overline{RD}$	I/O read input which enables the master CPU to read data and status words from the OUTPUT DATA BUS BUFFER or status register.
$\overline{CS}$	Chip select input used to select one UPI-41A out of several connected to a common data bus.
A_0	Address input used by the master processor to indicate whether byte transfer is data or command. During a write operation flag F_1 is set to the status of the A_0 input.
TEST 0, TEST 1	Input pins which can be directly tested using conditional branch instructions. (T_1) also functions as the event timer input (under software control). T_0 is used during PROM programming and verification in the 8741A.

Signal	Description
XTAL 1, XTAL 2	Inputs for a crystal, LC or an external timing signal to determine the internal oscillator frequency.
SYNC	Output signal which occurs once per UPI-41A instruction cycle. SYNC can be used as a strobe for external circuitry; it is also used to synchronize single step operation.
EA	External access input which allows emulation, testing and PROM verification.
PROG	Multifunction pin used as the program pulse input during PROM programming. During I/O expander access the PROG pin acts as an address/data strobe to the 8243.
$\overline{RESET}$	Input used to reset status flip-flops and to set the program counter to zero. $\overline{RESET}$ is also used during PROM programming and verification. $\overline{RESET}$ should be held low for a minimum of 8 instruction cycles after power-up.
$\overline{SS}$	Single step input used in the 8741A in conjunction with the SYNC output to step the program through each instruction.
V_{CC}	+5V main power supply pin.
V_{DD}	+5V during normal operation. +25V during programming operation. Low power standby supply pin in ROM version.
V_{SS}	Circuit ground potential.

4

UPI-41A FEATURES AND ENHANCEMENTS

1. Two Data Bus Buffers, one for input and one for output. This allows a much cleaner Master/Slave protocol.



2. 8 Bits of Status

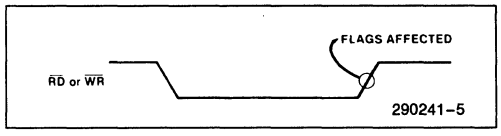
ST_7	ST_6	ST_5	ST_4	F_1	F_0	IBF	OBF
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

ST_4–ST_7 are user definable status bits. These bits are defined by the "MOV STS, A" single byte, single cycle instruction. Bits 4–7 of the accumulator are moved to bits 4–7 of the status register. Bits 0–3 of the status register are not affected.

MOV STS, A Op Code: 90H

1	0	0	1	0	0	0	0
D_7							D_0

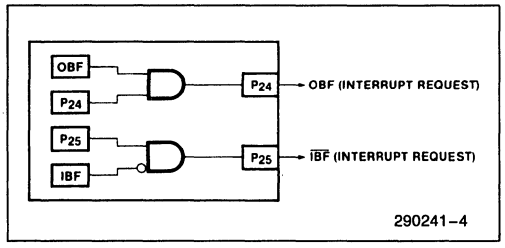
3. $\overline{RD}$ and $\overline{WR}$ are edge triggered. IBF, OBF, F_1 and INT change internally after the trailing edge of $\overline{RD}$ or $\overline{WR}$.



4. P_{24} and P_{25} are port pins or Buffer Flag pins which can be used to interrupt a master processor. These pins default to port pins on Reset.

   If the "EN FLAGS" instruction has been executed, P_{24} becomes the OBF (Output Buffer Full) pin. A "1" written to P_{24} enables the OBF pin (the pin outputs the OBF Status Bit). A "0" written to P_{24} disables the OBF pin (the pin remains low). This pin can be used to indicate that valid data is available from the UPI41A (in Output Data Bus Buffer).

   If "EN FLAGS" has been executed, P_{25} becomes the $\overline{IBF}$ (Input Buffer Full) pin. A "1" written to P_{25} enables the $\overline{IBF}$ pin (the pin outputs the inverse of the IBF Status Bit). A "0" written to P_{25} disables the $\overline{IBF}$ pin (the pin remains low). This pin can be used to indicate that the UPI is ready for data.



Data Bus Buffer Interrupt Capability

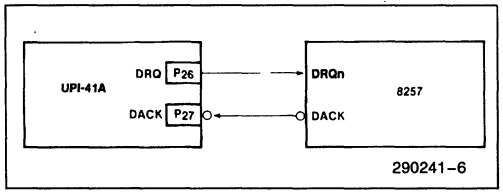
EN FLAGS Op Code: 0F5H

1	1	1	1	0	1	0	1
D_7							D_0

5. P_{26} and P_{27} are port pins or DMA handshake pins for use with a DMA controller. These pins default to port pins on Reset.

   If the "EN DMA" instruction has been executed, P_{26} becomes the DRQ (DMA Request) pin. A "1" written to P_{26} causes a DMA request (DRQ is activated). DRQ is deactivated by DACK•RD, DACK•WR, or execution of the "EN DMA" instruction.

   If "EN DMA" has been executed, P_{27} becomes the $\overline{DACK}$ (DMA Acknowledge) pin. This pin acts as a chip select input for the Data Bus Buffer registers during DMA transfers.



DMA Handshake Capability

EN DMA Op Code: 0E5H

1	1	1	0	0	1	0	1
D_7							D_0

APPLICATIONS

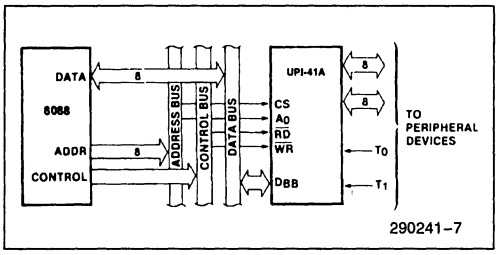


Figure 1. 8085A-8741A Interface

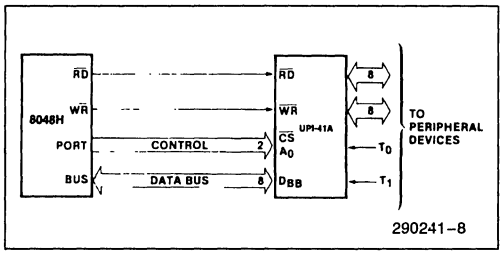


Figure 2. 8048-8741A Interface

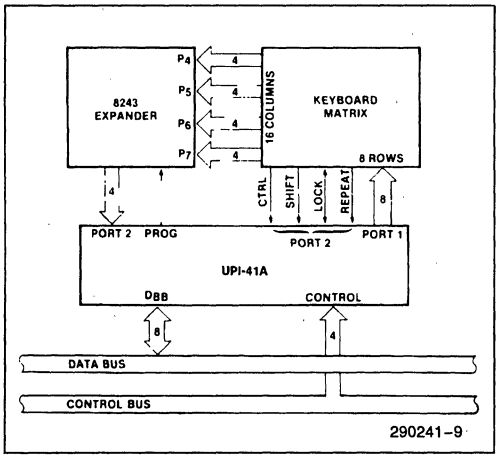


Figure 3. 8741A-8243 Keyboard Scanner

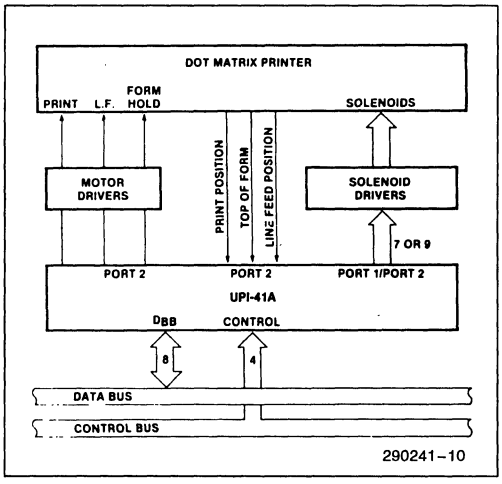


Figure 4. 8741A Matrix Printer Interface

4

PROGRAMMING, VERIFYING, AND ERASING THE 8741A EPROM

Programming Verification

In brief, the programming process consists of: activating the program mode, applying an address, latching the address, applying data, and applying a programming pulse. Each word is programmed completely before moving on to the next and is followed by a verification step. The following is a list of the pins used for programming and a description of their functions:

Pin	Function
XTAL 1	Clock Input (1 to 6 MHz)
$\overline{\text{Reset}}$	Initialization and Address Latching
Test 0	Selection of Program or Verify Mode
EA	Activation of Program/Verify Modes
BUS	Address and Data Input Data Output during Verify
P20–1	Address Input
V_{DD}	Programming Power Supply
PROG	Program Pulse Input

WARNING:
An attempt to program a missocketed 8741A will result in severe damage to the part. An indication of a properly socketed part is the appearance of the SYNC clock output. The lack of this clock may be used to disable the programmer.

The Program/Verify sequence is:

1. A_0 = 0V, CS = 5V, EA = 5V, $\overline{\text{RESET}}$ = 0V, TEST0 = 5V, V_{DD} = 5V, clock applied or internal oscillator operating, BUS and PROG floating
2. Insert 8741A in programming socket
3. TEST 0 = 0V (select program mode)
4. EA = 23V (active program mode)
5. Address applied to BUS and P20–1
6. $\overline{\text{RESET}}$ = 5V (latch address)
7. Data applied to BUS
8. V_{DD} = 25V (programming power)
9. PROG = 0V followed by one 50 ms pulse to 23V
10. V_{DD} = 5V
11. TEST 0 = 5V (verify mode)
12. Read and verify data on BUS
13. TEST 0 = 0V
14. $\overline{\text{RESET}}$ = 0V and repeat from step 6
15. Programmer should be at conditions of step 1 when 8741A is removed from socket

8741A Erasure Characteristics

The erasure characteristics of the 8741A are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000-4000Å range. Data show that constant exposure to room level fluorescent lighting could erase the typical 8741A in approximately 3 years while it would take approximately one week to cause erasure when exposed to direct sunlight. If the 8741A is to be exposed to these types of lighting conditions for extended periods of time, opaque labels are available from Intel which should be placed over the 8741A window to prevent unintentional erasure.

The recommended erasure procedure for the 8741A is exposure to shortwave ultraviolet light which has a wavelength of 2537Å. The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of 15 w-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12,000 μW/cm^2 power rating. The 8741A should be placed within one inch of the lamp tubes during erasure. Some lamps have a filter on their tubes which should be removed before erasure.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature under Bias ....0°C to +70°C
Storage Temperature .....−65°C to +150°C
Voltage on Any Pin with
Respect to Ground.....0.5V to +7V
Power Dissipation.....1.5W

'NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**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.*

D.C. CHARACTERISTICS

T_A = 0°C to +70°C, V_{SS} 0V, V_{CC} = V_{DD} = +5V ±10%

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IL}	Input Low Voltage (except XTAL1, XTAL2, $\overline{RESET}$)	−0.5	0.8	V	
V_{IL1}	Input Low Voltage (XTAL1, XTAL2, $\overline{RESET}$)	−0.5	0.6	V	
V_{IH}	Input High Voltage (except XTAL1, XTAL2, $\overline{RESET}$)	2.2	V_{CC}		
V_{IH1}	Input High Voltage (XTAL1, XTAL2, $\overline{RESET}$)	3.8	V_{CC}	V	
V_{OL}	Output Low Voltage (D_0–D_7)		0.45	V	I_{OL} = 2.0 mA
V_{OL1}	Output Low Voltage ($P_{10}P_{17}$, $P_{20}P_{27}$, Sync)		0.45	V	I_{OL} = 1.6 mA
V_{OL2}	Output Low Voltage (PROG)		0.45	V	I_{OL} = 1.0 mA
V_{OH}	Output High Voltage (D_0–D_7)	2.4		V	I_{OH} = −400 μA
V_{OH1}	Output High Voltage (All Other Outputs)	2.4		V	I_{OH} = −50 μA
I_{IL}	Input Leakage Current (T_0, T_1, $\overline{RD}$,$\overline{WR}$, $\overline{CS}$, A_0, EA)		±10	μA	$V_{SS} \le V_{IN} \le V_{CC}$
I_{OZ}	Output Leakage Current (D_0–D_7, High Z State)		±10	μA	V_{SS} +0.45 $\le V_{IN} \le V_{CC}$
I_{LI}	Low Input Load Current ($P_{10}P_{17}$, $P_{20}P_{27}$)		0.5	mA	V_{IL} = 0.8V
I_{LI1}	Low Input Load Current ($\overline{RESET}$, $\overline{SS}$)		0.2	mA	V_{IL} = 0.8V
I_{DD}	V_{DD} Supply Current		15	mA	Typical = 5 mA
I_{CC} + I_{DD}	Total Supply Current		125	mA	Typical = 60 mA

A.C. CHARACTERISTICS

T_A = 0°C to +70°C, V_{SS} = 0V, V_{CC} = V_{DD} = +5V ±10%

DBB READ

Symbol	Parameter	Min	Max	Unit	Test Conditions
t_{AR}	$\overline{CS}$, A_0 Setup to $\overline{RD}$ ↓	0		ns	
t_{RA}	$\overline{CS}$, A_0 Hold after $\overline{RD}$ ↑	0		ns	
t_{RR}	$\overline{RD}$ Pulse Width	250		ns	
t_{AD}	$\overline{CS}$, A_0 to Data Out Delay		225	ns	C_L = 150 pF
t_{RD}	$\overline{RD}$ ↓ to Data Out Delay		225	ns	C_L = 150 pF
t_{DF}	$\overline{RD}$ ↑ to Data Float Delay		100	ns	
t_{CY}	Cycle Time (except 8741A-8)	2.5	15	μs	6.0 MHz XTAL
t_{CY}	Cycle Time (8741A-8)	4.17	15	μs	3.6 MHz XTAL

4

DBB WRITE

Symbol	Parameter	Min	Max	Units	Test Conditions
t_{AW}	$\overline{CS}$, A_0 Setup to $\overline{WR}\downarrow$	0		ns	
t_{WA}	$\overline{CS}$, A_0 Hold after $\overline{WR}\uparrow$	0		ns	
t_{WW}	$\overline{WR}$ Pulse Width	250		ns	
t_{DW}	Data Setup to $\overline{WR}\uparrow$	150		ns	
t_{WD}	Data Hold after $\overline{WR}\uparrow$	0		ns	

A.C. TIMING SPECIFICATION FOR PROGRAMMING

T_A = 0°C to +70°C, V_{CC} = +5V ±10%

Symbol	Parameter	Min	Max	Units	Test Conditions
t_{AW}	Address Setup Time to $\overline{RESET}\uparrow$	$4t_{CY}$			
t_{WA}	Address Hold Time after $\overline{RESET}\uparrow$	$4t_{CY}$			
t_{DW}	Data in Setup Time to PROG ↑	$4t_{CY}$			
t_{WD}	Data in Hold Time after PROG ↓	$4t_{CY}$			
t_{PH}	$\overline{RESET}$ Hold Time to Verify	$4t_{CY}$			
t_{VDDW}	V_{DD} Setup Time to PROG ↑	$4t_{CY}$			
t_{VDDH}	V_{DD} Hold Time after PROG ↓	0			
t_{PW}	Program Pulse Width	50	60	ms	
t_{TW}	Test 0 Setup Time for Program Mode	$4t_{CY}$			
t_{WT}	Test 0 Hold Time after Program Mode	$4t_{CY}$			
t_{DO}	Test 0 to Data Out Delay		$4t_{CY}$		
t_{WW}	$\overline{RESET}$ Pulse Width to Latch Address	$4t_{CY}$			
t_r, t_f	V_{DD} and PROG Rise and Fall Times	0.5	2.0	μs	
t_{CY}	CPU Operation Cycle Time	5.0		μs	
t_{RE}	$\overline{RESET}$ Setup Time before EA ↑	$4t_{CY}$			

NOTE:
1. If TEST 0 is high, t_{DO} can be triggered by $\overline{RESET}\uparrow$.

D.C. SPECIFICATION FOR PROGRAMMING

T_A = 25°C ±5°C, V_{CC} = 5V ±5%, V_{DD} = 25V ±1V

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{DOH}	V_{DD} Program Voltage High Level	24.0	26.0	V	
V_{DDL}	V_{DD} Voltage Low Level	4.75	5.25	V	
V_{PH}	PROG Program Voltage High Level	21.5	24.5	V	
V_{PL}	PROG Voltage Low Level		0.2	V	
V_{EAH}	EA Program or Verify Voltage High Level	21.5	24.5	V	
V_{EAL}	EA Voltage Low Level		5.25	V	
I_{DD}	V_{DD} High Voltage Supply Current		30.0	mA	
I_{PROG}	PROG High Voltage Supply Current		16.0	mA	
I_{EA}	EA High Voltage Supply Current		1.0	mA	

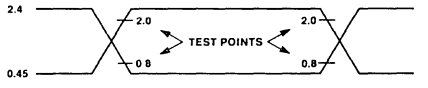
A.C. CHARACTERISTICS—DMA

Symbol	Parameter	Min	Max	Units	Test Conditions
t_{ACC}	$\overline{DACK}$ to $\overline{WR}$ or $\overline{RD}$	0		ns	
t_{CAC}	$\overline{RD}$ or $\overline{WR}$ to $\overline{DACK}$	0		ns	
t_{ACD}	$\overline{DACK}$ to Data Valid		225	ns	C_L = 150 pF
t_{CRQ}	$\overline{RD}$ or $\overline{WR}$ to DRQ Cleared		200	ns	

A.C. CHARACTERISTICS—PORT 2 T_A = 0°C to +70°C, V_{CC} = +5V ±10%

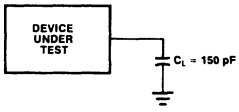
Symbol	Parameter	Min	Max	Units	Test Conditions
t_{CP}	Port Control Setup before Falling Edge of PROG	10		ns	
t_{PC}	Port Control Hold after Falling Edge of PROG	100		ns	
t_{PR}	PROG to Time P2 Input Must Be Valid		810	ns	
t_{PF}	Input Data Hold Time	0	150	ns	
t_{DP}	Output Data Setup Time	250		ns	
t_{PD}	Output Data Hold Time	65		ns	
t_{PP}	PROG Pulse Width	1200		ns	

A.C. TESTING INPUT/OUTPUT WAVEFORM



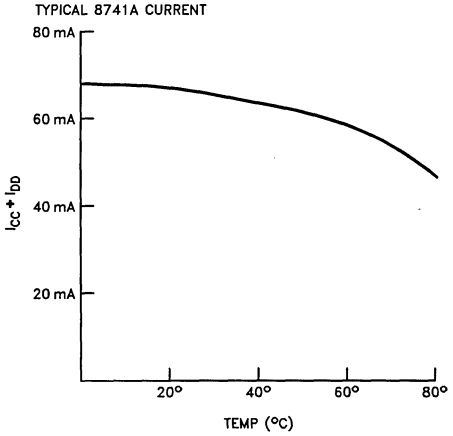
290241-23

A.C. TESTING LOAD CIRCUIT



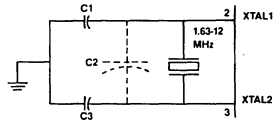
290241-15

TYPICAL 8741A CURRENT



290241-14

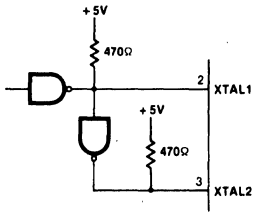
CRYSTAL OSCILLATOR MODE



290241-11

Crystal series resistance should be
< 75Ω at 6 MHz; < 180Ω at 3.6 MHz.

DRIVING FROM EXTERNAL SOURCE

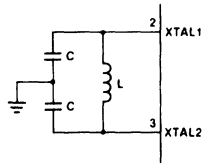


290241-12

Both XTAL1 and XTAL2 should be driven.
Resistors to V_{CC} are needed to ensure V_{IH} = 3.8V if TTL circuitry is used.

LC OSCILLATOR MODE

L	C	NOMINAL f
45 μH	20 pF	5.2 MHz
120 μH	20 pF	3.2 MHz



$$f = \frac{1}{2\pi\sqrt{LC'}}$$

$$C' = \frac{C + 3Cpp}{2}$$

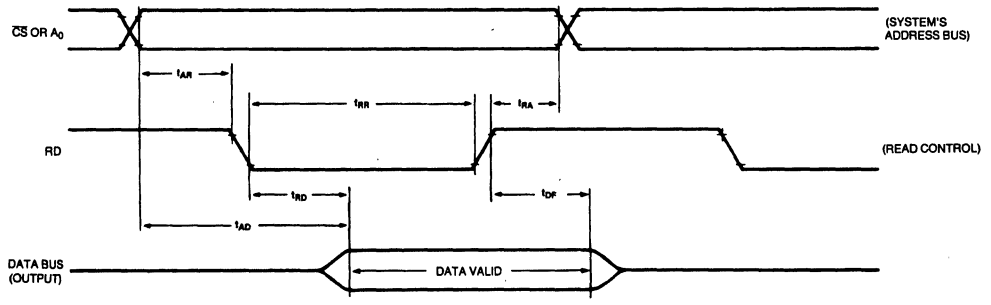
Cpp ≅ 5–10 pF
Pin-to-Pin Capacitance

290241-13

Each C should be approximately 20 pF, including stray capacitance.

WAVEFORMS

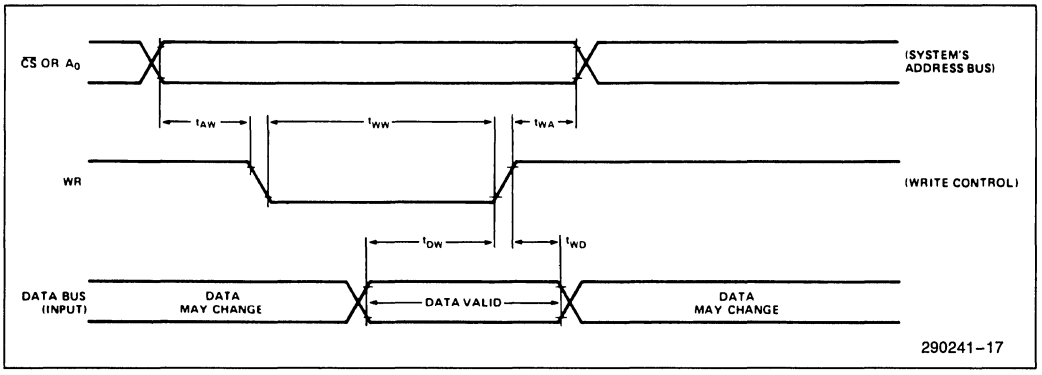
READ OPERATION—DATA BUS BUFFER REGISTER



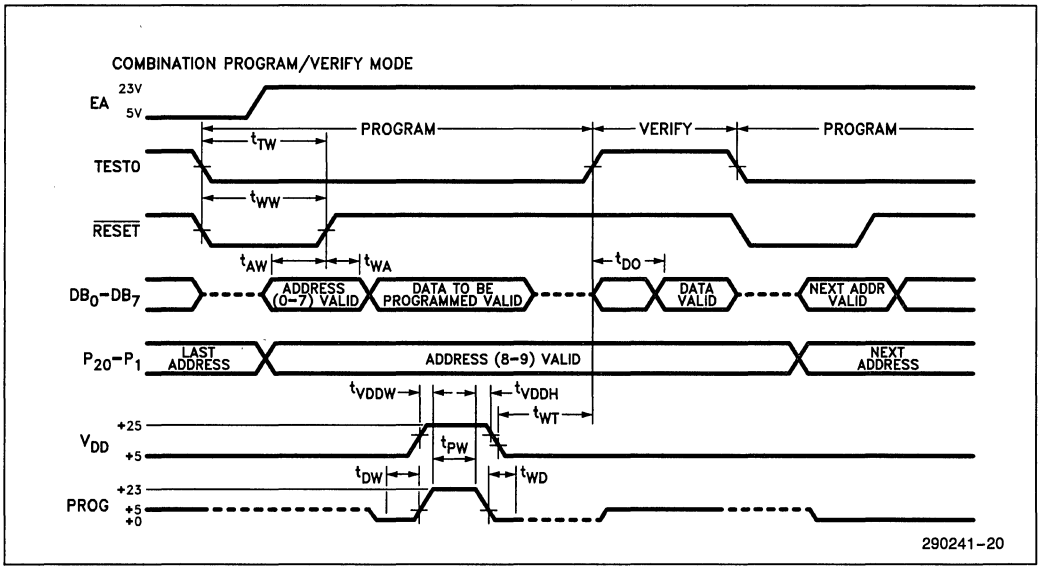
290241-16

WAVEFORMS

WRITE OPERATION—DATA BUS BUFFER REGISTER



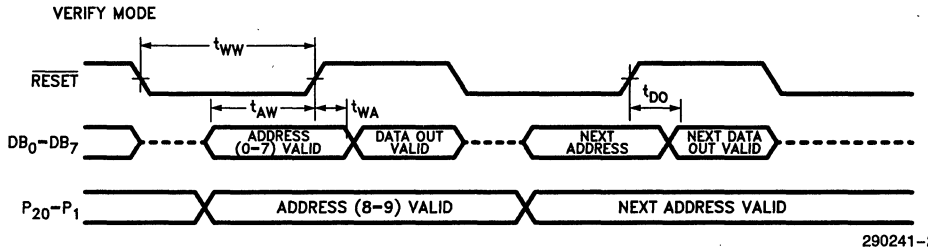
COMBINATION PROGRAM/VERIFY MODE



4

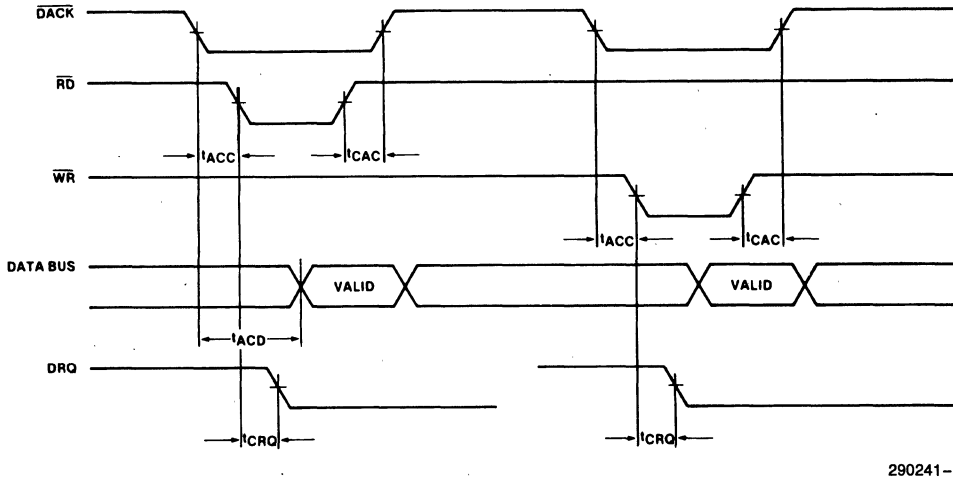
WAVEFORMS

VERIFY MODE

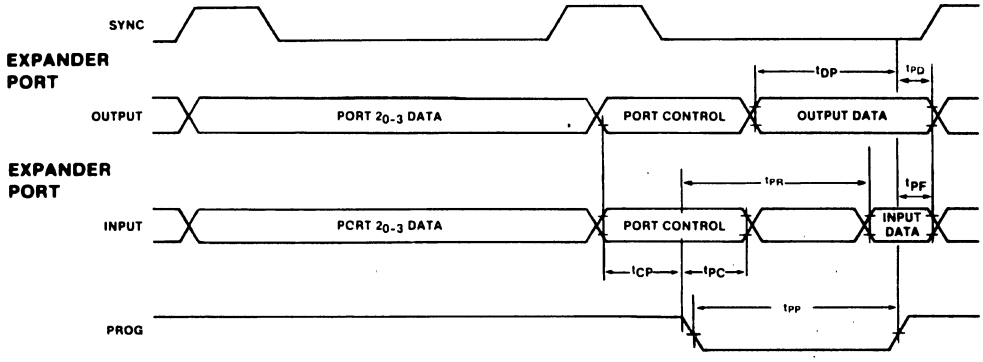


NOTES:
1. PROG must float if EA is low (i.e., ≠23V), or if T0 = 5V for the 8741A.
2. XTAL1 and XTAL2 driven by 3.6 MHz clock will give 7.17 μs t_{CY}. This is acceptable for 8741-8 parts as well as standard parts.
PROG must float or = 5V when EA is high.
3. A_0 must be held low (i.e., = 0V) during program/verify modes.

DMA

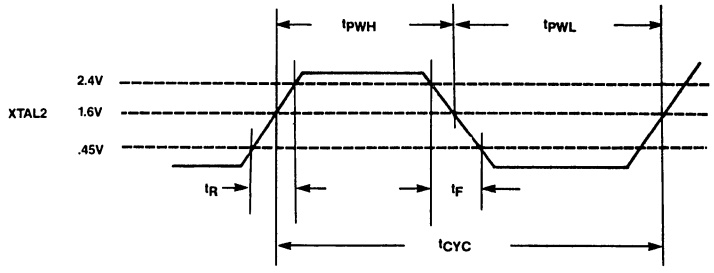


PORT 2 TIMING



290241-19

PORT TIMING DURING EXTERNAL ACCESS (EA)



290241-18

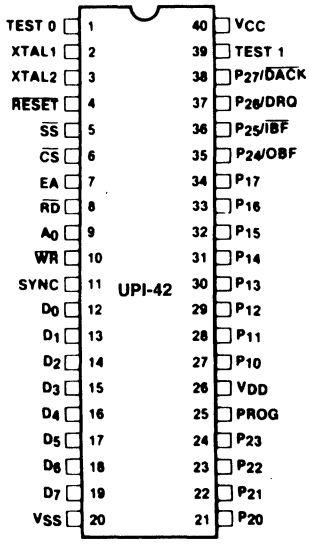
On the rising edge of SYNC and EA is enabled, port data is valid and can be strobed. On the trailing edge of sync the program counter contents are available.

8742
UNIVERSAL PERIPHERAL INTERFACE 8-BIT SLAVE MICROCONTROLLER

- **8742: 12 MHz**
- **Pin, Software and Architecturally Compatible with 8741A**
- **8-Bit CPU plus ROM, RAM, I/O, Timer and Clock in a Single Package**
- **2048 x 8 EPROM, 128 x 8 RAM, 8-Bit Timer/Counter, 18 Programmable I/O Pins**
- **One 8-Bit Status and Two Data Registers for Asynchronous Slave-to-Master Interface**
- **DMA, Interrupt, or Polled Operation Supported**
- **Fully Compatible with all Intel and Most Other Microprocessor Families**
- **Expandable I/O**
- **RAM Power-Down Capability**
- **Over 90 Instructions: 70% Single Byte**
- **Available in EXPRESS — Standard Temperature Range**

The Intel 8742 is a general-purpose Universal Peripheral Interface that allows designers to grow their own customized solution for peripheral device control. It contains a low-cost microcomputer with 2K of program memory, 128 bytes of data memory, 8-bit timer/counter, and clock generator in a single 40-pin package. Interface registers are included to enable the UPI device to function as a peripheral controller in the MCS®-48, MCS-51, MCS-80, MCS-85, 8088, 8086 and other 8-, 16-bit systems.

The 8742 is software, pin, and architecturally compatible with the 8741A. The 8742 doubles the on-chip memory space to allow for additional features and performance to be incorporated in upgraded 8741A designs. For new designs, the additional memory and performance of the 8742 extends the UPI concept to more complex motor control tasks, 80-column printers and process control applications as examples.



290256-2

Figure 1. Pin Configuration

November 1991
Order Number: 290256-001

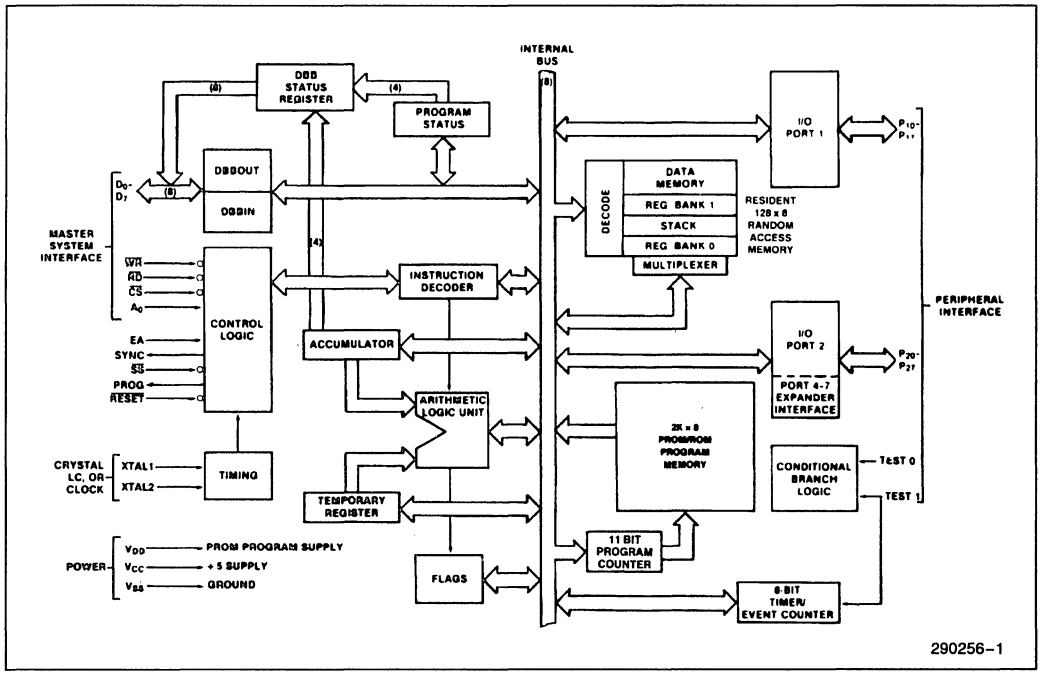


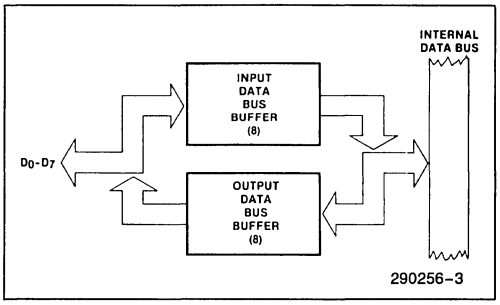
Figure 2. Block Diagram

Table 1. Pin Description

Symbol	DIP Pin No.	Type	Name and Function
TEST 0, TEST 1	1 39	I	**TEST INPUTS:** Input pins which can be directly tested using conditional branch instructions. **FREQUENCY REFERENCE:** TEST 1 (T_1) also functions as the event timer input (under software control). TEST 0 (T_0) is used during PROM programming and EPROM verification.
XTAL 1, XTAL 2	2 3	I	**INPUTS:** Inputs for a crystal, LC or an external timing signal to determine the internal oscillator frequency.
$\overline{\text{RESET}}$	4	I	**RESET:** Input used to reset status flip-flops and to set the program counter to zero. $\overline{\text{RESET}}$ is also used during EPROM programming and verification.
$\overline{\text{SS}}$	5	I	**SINGLE STEP:** Single step input used in conjunction with the SYNC output to step the program through each instruction (EPROM). This should be tied to +5V when not used.
$\overline{\text{CS}}$	6	I	**CHIP SELECT:** Chip select input used to select one UPI microcomputer out of several connected to a common data bus.
EA	7	I	**EXTERNAL ACCESS:** External access input which allows emulation, testing and EPROM verification. This pin should be tied low if unused.
$\overline{\text{RD}}$	8	I	**READ:** I/O read input which enables the master CPU to read data and status words from the OUTPUT DATA BUS BUFFER or status register.
A_0	9	I	**COMMAND/DATA SELECT:** Address Input used by the master processor to indicate whether byte transfer is data ($A_0 = 0$, F1 is reset) or command ($A_0 = 1$, F1 is set). $A_0 = 0$ during program and verify operations.
$\overline{\text{WR}}$	10	I	**WRITE:** I/O write input which enables the master CPU to write data and command words to the UPI INPUT DATA BUS BUFFER.
SYNC	11	O	**OUTPUT CLOCK:** Output signal which occurs once per UPI instruction cycle. SYNC can be used as a strobe for external circuitry; it is also used to synchronize single step operation.
D_0–D_7 (BUS)	12–19	I/O	**DATA BUS:** Three-state, bidirectional DATA BUS BUFFER lines used to interface the UPI microcomputer to an 8-bit master system data bus.
P_{10}–P_{17}	27–34	I/O	**PORT 1:** 8-bit, PORT 1 quasi-bidirectional I/O lines.
P_{20}–P_{27}	21–24 35–38	I/O	**PORT 2:** 8-bit, PORT 2 quasi-bidirectional I/O lines. The lower 4 bits (P_{20}–P_{23}) interface directly to the 8243 I/O expander device and contain address and data information during PORT 4–7 access. The upper 4 bits (P_{24}–P_{27}) can be programmed to provide interrupt Request and DMA Handshake capability. Software control can configure P_{24} as Output Buffer Full (OBF) interrupt, P_{25} as Input Buffer Full ($\overline{\text{IBF}}$) interrupt, P_{26} as DMA Request (DRQ), and P_{27} as DMA ACKnowledge ($\overline{\text{DACK}}$).
PROG	25	I/O	**PROGRAM:** Multifunction pin used as the program pulse input during PROM programming. During I/O expander access the PROG pin acts as an address/data strobe to the 8243. This pin should be tied high if unused.
V_{CC}	40		**POWER:** +5V main power supply pin.
V_{DD}	26		**POWER:** +5V during normal operation. +21V during programming operation. Low power standby supply pin.
V_{SS}	20		**GROUND:** Circuit ground potential.

UPI-42 FEATURES

1. Two Data Bus Buffers, one for input and one for output. This allows a much cleaner Master/Slave protocol.



2. 8 Bits of Status

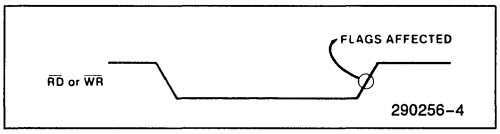
ST_7	ST_6	ST_5	ST_4	F_1	F_0	IBF	OBF
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

ST_4–ST_7 are user definable status bits. These bits are defined by the "MOV STS, A" single byte, single cycle instruction. Bits 4–7 of the acccumulator are moved to bits 4–7 of the status register. Bits 0–3 of the status register are not affected.

MOV STS, A Op Code: 90H

1	0	0	1	0	0	0	0
D_7							D_0

3. $\overline{RD}$ and $\overline{WR}$ are edge triggered. IBF, OBF, F_1 and INT change internally after the trailing edge of $\overline{RD}$ or $\overline{WR}$.

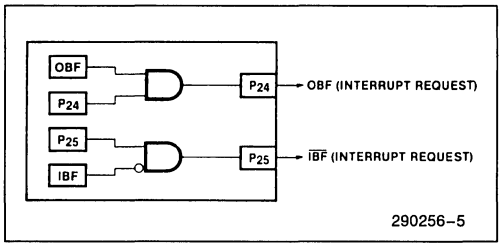


During the time that the host CPU is reading the status register, the 8742 is prevented from updating this register or is "locked out".

4. P_{24} and P_{25} are port pins or Buffer Flag pins which can be used to interrupt a master processor. These pins default to port pins on Reset.

If the "EN FLAGS" instruction has been executed, P_{24} becomes the OBF (Output Buffer Full) pin. A "1" written to P_{24} enables the OBF pin (the pin outputs the OBF Status Bit). A "0" written to P_{24} disables the OBF pin (the pin remains low). This pin can be used to indicate that valid data is available from the UPI-41A (in Output Data Bus Buffer).

If "EN FLAGS" has been executed, P_{25} becomes the $\overline{IBF}$ (Input Buffer Full) pin. A "1" written to P_{25} enables the $\overline{IBF}$ pin (the pin outputs the inverse of the IBF Status Bit. A "0" written to P_{25} disables the $\overline{IBF}$ pin (the pin remains low). This pin can be used to indicate that the UPI is ready for data.



Data Bus Buffer Interrupt Capability

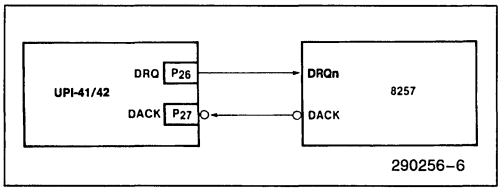
EN FLAGS Op Code: 0F5H

1	1	1	1	0	1	0	1
D_7							D_0

5. P_{26} and P_{27} are port pins or DMA handshake pins for use with a DMA controller. These pins default to port pins on Reset.

If the "EN DMA" instruction has been executed, P_{26} becomes the DRQ (DMA Request) pin. A "1" written to P_{26} causes a DMA request (DRQ is activated). DRQ is deactivated by DACK•RD, DACK•WR, or execution of the "EN DMA" instruction.

If "EN DMA" has been executed, P_{27} becomes the $\overline{DACK}$ (DMA Acknowledge) pin. This pin acts as a chip select input for the Data Bus Buffer registers during DMA transfers.



DMA Handshake Capability

EN DMA Op Code: 0E5H

1	1	1	0	0	1	0	1
D_7							D_0

6. The RESET input on the 8742, includes a 2-stage synchronizer to support reliable reset operation for 12 MHz operation.

7. When EA is enabled on the 8742, the program counter is placed on Port 1 and the lower three bits of Port 2 (MSB = P_{22}, LSB = P_{10}). On the 8742 this information is multiplexed with PORT DATA (see port timing diagrams at end of this data sheet).

4

APPLICATIONS

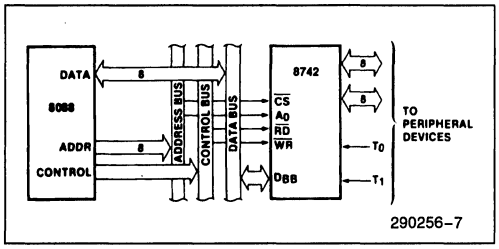


Figure 3. 8088-8742 Interface

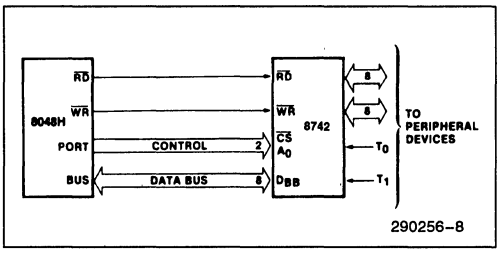


Figure 4. 8048H-8742 Interface

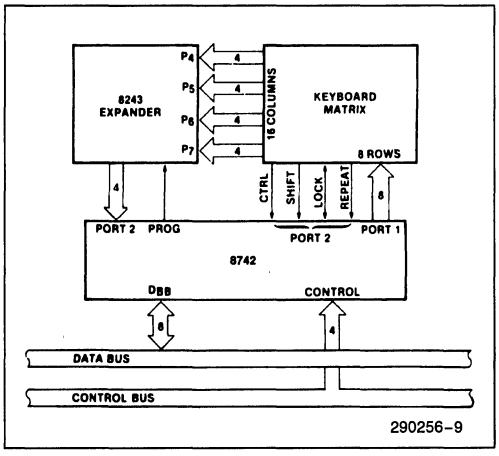


Figure 5. 8742-8243 Keyboard Scanner

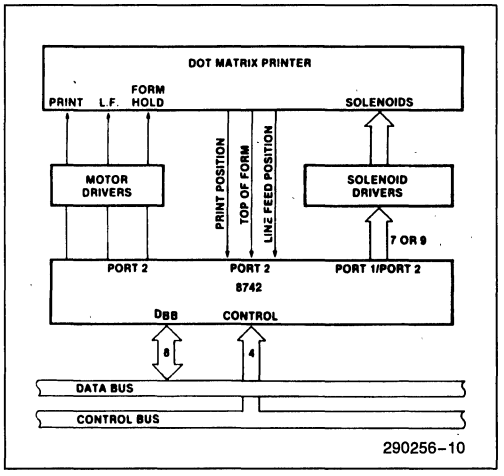


Figure 6. 8742 80-Column Matrix Printer Interface

PROGRAMMING, VERIFYING, AND ERASING THE 8742 EPROM

Programming Verification

In brief, the programming process consists of: activating the program mode, applying an address, latching the address, applying data, and applying a programming pulse. Each word is programmed completely before moving on to the next and is followed by a verification step. The following is a list of the pins used for programming and a description of their functions:

Pin	Function
XTAL 1	Clock-Input
$\overline{\text{Reset}}$	Initialization and Address Latching
Test 0	Selection of Program or Verify Mode
EA	Activation of Program/Verify Modes
BUS	Address and Data Input Data Output During Verify
P_{20-12}	Address Input
V_{DD}	Programming Power Supply
PROG	Program Pulse Input

WARNING
An attempt to program a missocketed 8742 will result in severe damage to the part. An indication of a properly socketed part is the appearance of the SYNC clock output. The lack of this clock may be used to disable the programmer.

The Program/Verify sequence is:

1. A_0 = 0V, CS = 5V, EA = 5V, RESET = 0V, TESTO = 5V, V_{DD} = 5V, clock applied or internal oscillator operating, BUS floating, PROG = 5V.
2. Insert 8742 in programming socket
3. TEST 0 = 0V (select program mode)
4. EA = 18V (active program mode)
5. Address applied to BUS and P_{20-22}
6. $\overline{\text{RESET}}$ = 5V (latch address)
7. Data applied to BUS**
8. V_{DD} = 21V (programming power)
9. PROG = V_{CC} followed by one 50 ms pulse to 18V
10. V_{DD} = 5V
11. TEST 0 = 5V (verify mode)
12. Read and verify data on BUS
13. TEST 0 = 0V
14. $\overline{\text{RESET}}$ = 0V and repeat from step 5
15. Programmer should be at conditions of step 1 when 8742 is removed from socket

8742 Erasure Characteristics

The erasure characteristics of the 8742 are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000-4000Å range. Data shows that constant exposure to room level fluorescent lighting could erase the typical 8742 in approximately 3 years while it would take approximately one week to cause erasure when exposed to direct sunlight. If the 8742 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels are available from Intel which should be placed over the 8742 window to prevent unintentional erasure.

The recommended erasure procedure for the 8742 is exposure to shortwave ultraviolet light which has a wavelength of 2537Å. The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of 15 w-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12,000 μW/cm^2 power rating. The 8742 should be placed within one inch of the lamp tubes during erasure. Some lamps have a filter on their tubes which should be removed before erasure.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias .....0°C to 70°C
Storage Temperature .....−65°C to +150°C
Voltage on Any Pin With Respect to Ground .....−0.5 to +7V
Power Dissipation.....1.5W

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**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.*

D.C. CHARACTERISTICS $T_A = 0°$ to $+70°C$, $V_{CC} = V_{DD} = +5V \pm 10\%$

Symbol	Parameter	8742		Units	Test Conditions
		Min	Max		
V_{IL}	Input Low Voltage (Except XTAL1, XTAL2, RESET)	−0.5	0.8	V	
V_{IL1}	Input Low Voltage (XTAL1, XTAL2, RESET)	−0.5	0.6	V	
V_{IH}	Input High Voltage (Except XTAL1, XTAL2, RESET)	2.0	V_{CC}	V	
V_{IH1}	Input High Voltage (XTLA1, XTAL2, RESET)	3.5	V_{CC}	V	
V_{OL}	Output Low Voltage (D_0–D_7)		0.45	V	$I_{OL} = 2.0$ mA
V_{OL1}	Output Low Voltage (P_{10}–P_{17}, P_{20}–P_{27}, Sync)		0.45	V	$I_{OL} = 1.6$ mA
V_{OL2}	Output Low Voltage (PROG)		0.45	V	$I_{OL} = 1.0$ mA
V_{OH}	Output High Voltage (D_0–D_7)	2.4		V	$I_{OH} = -400\ \mu A$
V_{OH1}	Output High Voltage (All Other Outupts)	2.4			$I_{OH} = -50\ \mu A$
I_{IL}	Input Leakage Current (T_0, T_1, RD, WR, CS, A_0, EA)		±10	μA	$V_{SS} \le V_{IN} \le V_{CC}$
I_{OFL}	Output Leakage Current (D_0–D_7, High Z State)		±10	μA	$V_{SS} + 0.45 \le V_{OUT} \le V_{CC}$
I_{LI}	Low Input Load Current (P_{10}–P_{17}, P_{20}–P_{27})		0.3	mA	$V_{IL} = 0.8V$
I_{LI1}	Low Input Load Current (RESET, SS)		0.2	mA	$V_{IL} = 0.8V$
I_{DD}	V_{DD} Supply Current		10	mA	Typical = 5 mA
$I_{CC} + I_{DD}$	Total Supply Current		125	mA	Typical = 60 mA
I_{IH}	Input Leakage Current (P_{10}–P_{17}, P_{20}–P_{27})		100	μA	$V_{IN} = V_{CC}$
C_{IN}	Input Capacitance		10	pF	
$C_{1\,0}$	I/O Capacitance		20	pF	

D.C. CHARACTERISTICS—PROGRAMMING

$T_A = 25°C \pm 5°C$, $V_{CC} = 5V \pm 5\%$, $V_{DD} = 21V \pm 0.5V$

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{DOH}	V_{DD} Program Voltage High Level	20.5	21.5	V	
V_{DDL}	V_{DD} Voltage Low Level	4.75	5.25	V	
V_{PH}	PROG Program Voltage High Level	17.5	18.5	V	
V_{PL}	PROG Voltage Low Level	$V_{CC}-0.5$	V_{CC}	V	
V_{EAH}	EA Program or Verify Voltage High Level	17.5	18.5	V	
V_{EAL}	EA Voltage Low Level		5.25	V	
I_{DD}	V_{DD} High Voltage Supply Current		30.0	mA	
I_{PROG}	PROG High Voltage Supply Current		1.0	mA	
I_{EA}	EA High Voltage Supply Current		1.0	mA	

A.C. CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{SS} = 0V$, $V_{CC} = V_{DD} = +5V \pm 10\%$

DBB READ

Symbol	Parameter	8742		Units
		Min	Max	
t_{AR}	CS, A_0 Setup to RD ↓	0		ns
t_{RA}	CS, A_0 Hold after RD ↑	0		ns
t_{RR}	RD Pulse Width	160		ns
t_{AD}	CS, A_0 to Data Out Delay		130	ns
t_{RD}	RD ↓ to Data Out Delay		130	ns
t_{DF}	RD ↑ to Data Float Delay		85	ns
t_{CY}	Cycle Time	1.25	15	μs(1)

DBB WRITE

Symbol	Parameter	Min	Max	Units
t_{AW}	CS, A_0 Setup to WR ↓	0		ns
t_{WA}	CS, A_0 Hold after WR ↑	0		ns
t_{WW}	WR Pulse Width	160		ns
t_{DW}	Data Setup to WR ↑	130		ns
t_{WD}	Data Hold after WR ↑	0		ns

NOTE:
1. $T_{CY} = 15/f(XTAL)$

A.C. CHARACTERISTICS $T_A = 25°C \pm 5°C$, $V_{CC} = 5V \pm 5\%$, $V_{DD} = +21V \pm 0.5$

PROGRAMMING

Symbol	Parameter	Min	Max	Units	Test Conditions
t_{AW}	Address Setup Time to RESET ↑	$4t_{CY}$			
t_{WA}	Address Hold Time after RESET ↑	$4t_{CY}$			
t_{DW}	Data in Setup Time to PROG ↑	$4t_{CY}$			
t_{WD}	Data in Hold Time after PROG ↓	$4t_{CY}$			
t_{PH}	RESET Hold Time to Verify	$4t_{CY}$			
t_{VDDW}	V_{DD} Setup Time to PROG ↑	0	1.0	mS	
t_{VDDH}	V_{DD} Hold Time after PROG ↑	0	1.0	mS	
t_{PW}	Program Pulse Width	50	60	mS	
t_{TW}	Test 0 Setup Time for Program Mode	$4t_{CY}$			
t_{WT}	Test 0 Hold Time after Program Mode	$4t_{CY}$			
t_{DO}	Test 0 to Data Out Delay		$4t_{CY}$		
t_{WW}	RESET Pulse Width to Latch Address	$4t_{CY}$			
t_r, t_f	V_{DD} and PROG Rise and Fall Times	0.5	2.0	μs	
t_{CY}	CPU Operation Cycle Time	4.0		μs	
t_{RE}	RESET Setup Time before EA ↑	$4t_{CY}$			

NOTE:
If TEST 0 is high, t_{DO} can be triggered by RESET ↑.

A.C. CHARACTERISTICS DMA

Symbol	Parameter	8642/8742		Units
		Min	Max	
t_{ACC}	DACK to WR or RD	0		ns
t_{CAC}	RD or WR to DACK	0		ns
t_{ACD}	DACK to Data Valid		130	ns
t_{CRQ}	RD or WR to DRQ Cleared		100	ns(1)

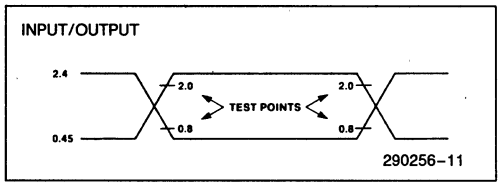
NOTE:
1. C_L = 150 pF.

A.C. CHARACTERISTICS PORT 2 T_A = 0°C to +70°C, V_{CC} = +5V ±10%

Symbol	Parameter	f(t_{CY})	8742/8642(3)		Units
			Min	Max	
t_{CP}	Port Control Setup before Falling Edge of PROG	1/15 t_{CY} − 28	55		ns(1)
t_{PC}	Port Control Hold after Falling Edge of PROG	1/10 t_{CY}	125		ns(2)
t_{PR}	PROG to Time P2 Input Must Be Valid	8/15 t_{CY} − 16		650	ns(1)
t_{PF}	Input Data Hold Time		0	150	ns(2)
t_{DP}	Output Data Setup Time	2/10 t_{CY}	250		ns(1)
t_{PD}	Output Data Hold Time	1/10 t_{CY} − 80	45		ns(2)
t_{PP}	PROG Pulse Width	6/10 t_{CY}	750		ns

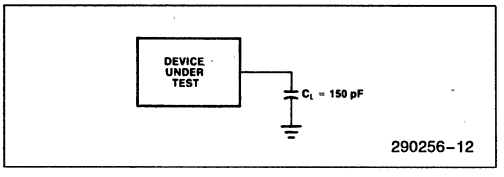
NOTES:
1. C_L = 80 pF.
2. C_L = 20 pF.
3. t_{CY} = 1.25 μs.

A.C. TESTING INPUT/OUTPUT WAVEFORM



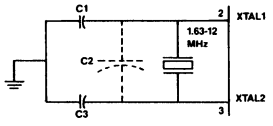
290256-11

A.C. TESTING LOAD CIRCUIT



290256-12

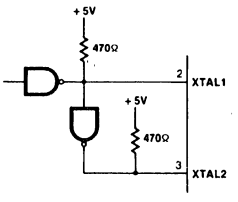
CRYSTAL OSCILLATOR MODE



290256-13

Crystal Series Resistance Should be <750 at 12 MHz; <180Ω at 3.6 MHz.

DRIVING FROM EXTERNAL SOURCE

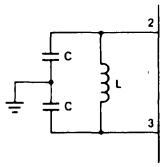


290256-14

Rise and Fall Times Should Not Exceed 20 ns. Resistors to V_{CC} are Needed to Ensure V_{IH} = 3.5V if TTL Circuitry is Used.

LC OSCILLATOR MODE

L	C	NOMINAL
45 H	20 pF	5.2 MHz
120 H	20 pF	3.2 MHz



$$f = \frac{1}{2\pi\sqrt{LC'}}$$

$$C' = \frac{C + 3Cpp}{2}$$

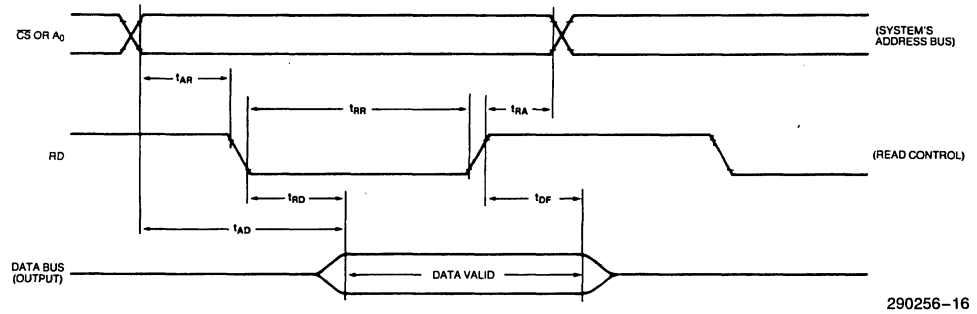
Cpp ≅ 5 pF–10 pF
Pin-to-Pin Capacitance

290256–15

Each C Should be Approximately 20 pF, including Stray Capacitance.

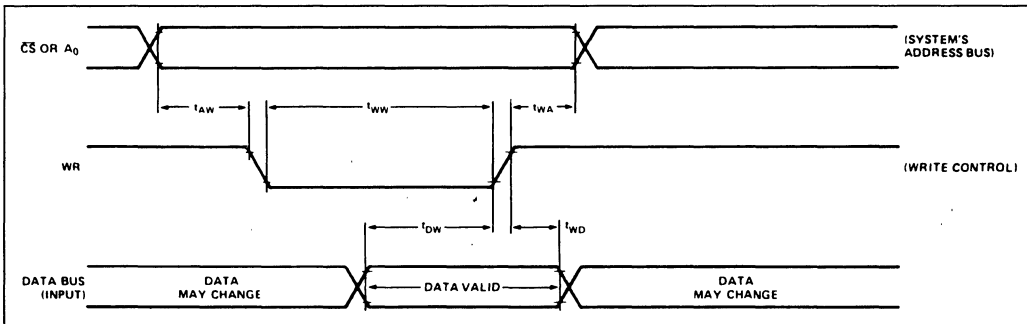
WAVEFORMS

READ OPERATION—DATA BUS BUFFER REGISTER



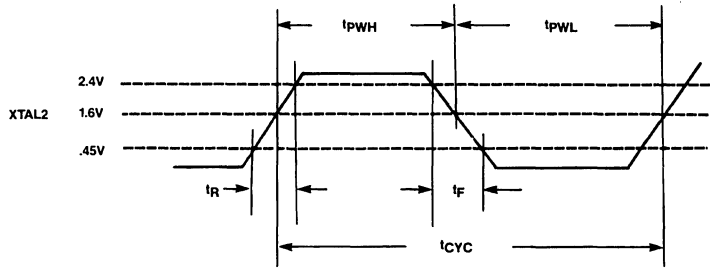
290256–16

WRITE OPERATION—DATA BUS BUFFER REGISTER



290256–17

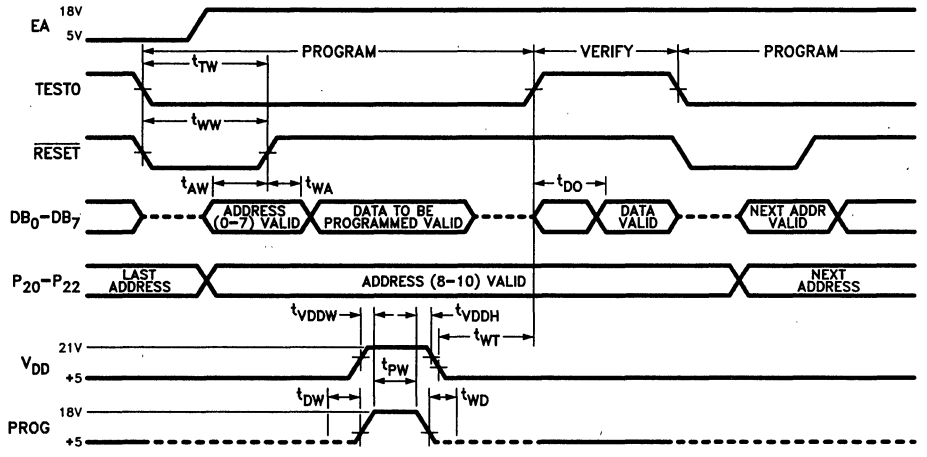
CLOCK TIMING



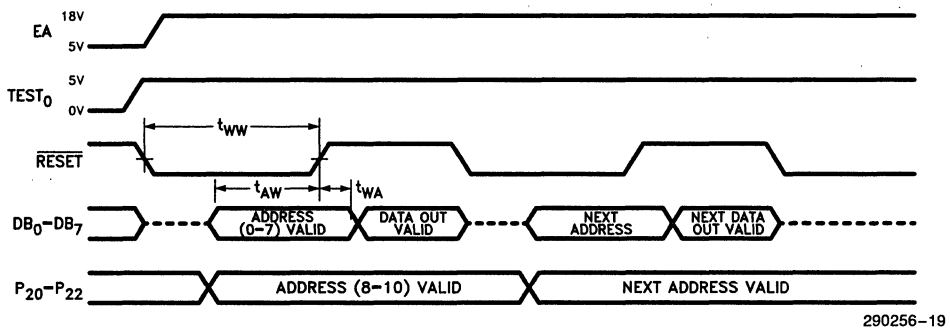
290256–23

WAVEFORMS

COMBINATION PROGRAM/VERIFY MODE



VERIFY MODE



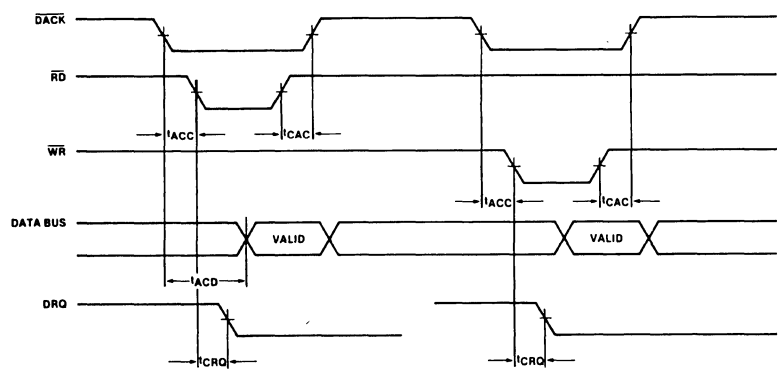
NOTES:
1. PROG must float if EA is low or EA is low or if $TEST_0$ = 5V.
2. A_0 must be held low (i.e., = 0V) during program/verify modes.
3. Test 0 must be held high.

The 8742 EPROM can be programmed by the following Intel products:

1. Universal PROM Programmer (UPP 103) peripheral of the Intellec Development System with a UPP-549 Personality Card.
2. iUP-200/iUP-201 PROM Programmer with the iUP-F87/44 Personality Module.

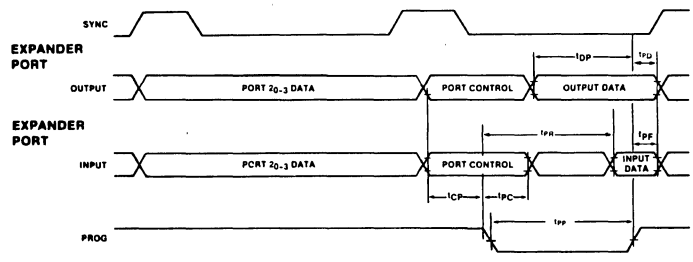
WAVEFORMS (Continued)

DMA



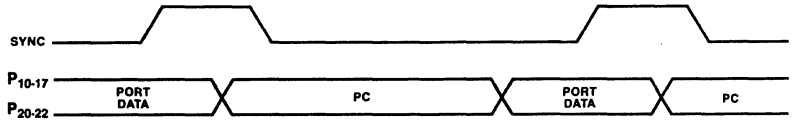
290256-20

PORT 2



290256-21

PORT TIMING DURING EXTERNAL ACCESS (EA)



290256-22

On the Rising Edge of SYNC and EA is Enabled, Port Data is Valid and can be Strobed on the Trailing Edge of Sync the Program Counter Contents are Available.

UPI-C42/UPI-L42
UNIVERSAL PERIPHERAL INTERFACE
CHMOS 8-BIT SLAVE MICROCONTROLLER

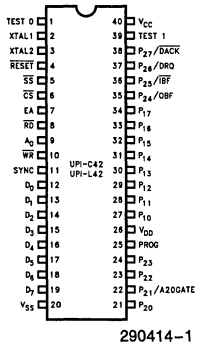
- **Pin, Software and Architecturally Compatible with all UPI-41 and UPI-42 Products**
- **Low Voltage Operation with the UPI-L42**
  — Full 3.3V Support
- **Integrated Auto A20 Gate Support**
- **Suspend Power Down Mode**
- **Security Bit Code Protection Support**
- **8-Bit CPU plus ROM/OTP EPROM, RAM, I/O, Timer/Counter and Clock in a Single Package**
- **4096 x 8 ROM/OTP, 256 x 8 RAM 8-Bit Timer/Counter, 18 Programmable I/O Pins**
- **DMA, Interrupt, or Polled Operation Supported**
- **One 8-Bit Status and Two Data Registers for Asynchronous Slave-to-Master Interface**
- **Fully Compatible with all Intel and Most Other Microprocessor Families**
- **Interchangeable ROM and OTP EPROM Versions**
- **Expandable I/O**
- **Sync Mode Available**
- **Over 90 Instructions: 70% Single Byte**
- **Quick Pulse Programming Algorithm**
  — Fast OTP Programming
- **Available in 40-Lead Plastic, 44-Lead Plastic Leaded Chip Carrier, and 44-Lead Quad Flat Pack Packages**
  (See Packaging Spec., Order #240800, Package Type P, N, and S)

The UPI-C42 is an enhanced CHMOS version of the industry standard Intel UPI-42 family. It is fabricated on Intel's CHMOS III-E process. The UPI-C42 is pin, software, and architecturally compatible with the NMOS UPI family. The UPI-C42 has all of the same features of the NMOS family plus a larger user programmable memory array (4K), integrated auto A20 gate support, and lower power consumption inherent to a CHMOS product.

The UPI-L42 offers the same functionality and socket compatibility as the UPI-C42 as well as providing low voltage 3.3V operation.

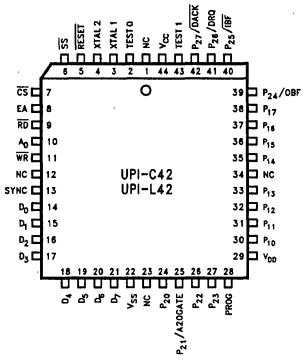
The UPI-C42 is essentially a "slave" microcontroller, or a microcontroller with a slave interface included on the chip. Interface registers are included to enable the UPI device to function as a slave peripheral controller in the MCS Modules and iAPX family, as well as other 8-, 16-, and 32-bit systems.

To allow full user flexibility, the program memory is available in ROM and One-Time Programmable EPROM (OTP).



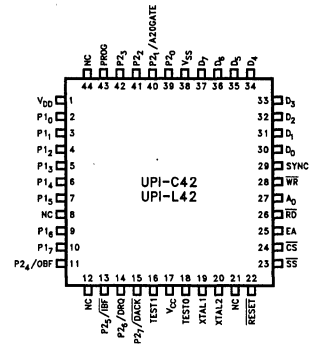
290414-1

Figure 1. DIP Pin Configuration



290414-2

Figure 2. PLCC Pin Configuration



290414-3

Figure 3. QFP Pin Configuration

October 1995
Order Number: 290414-003

intel®

5

Support Peripherals

UPI-C42/UPI-L42
UNIVERSAL PERIPHERAL INTERFACE
CHMOS 8-BIT SLAVE MICROCONTROLLER

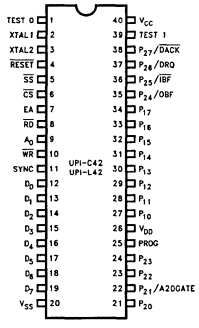
- **Pin, Software and Architecturally Compatible with all UPI-41 and UPI-42 Products**
- **Low Voltage Operation with the UPI-L42**
  — Full 3.3V Support
- **Hardware A20 Gate Support**
- **Suspend Power Down Mode**
- **Security Bit Code Protection Support**
- **8-Bit CPU plus ROM/OTP EPROM, RAM, I/O, Timer/Counter and Clock in a Single Package**
- **4096 x 8 ROM/OTP, 256 x 8 RAM 8-Bit Timer/Counter, 18 Programmable I/O Pins**
- **DMA, Interrupt, or Polled Operation Supported**
- **One 8-Bit Status and Two Data Registers for Asynchronous Slave-to-Master Interface**
- **Fully Compatible with all Intel and Most Other Microprocessor Families**
- **Interchangeable ROM and OTP EPROM Versions**
- **Expandable I/O**
- **Sync Mode Available**
- **Over 90 Instructions: 70% Single Byte**
- **Quick Pulse Programming Algorithm**
  — Fast OTP Programming
- **Available in 40-Lead Plastic, 44-Lead Plastic Leaded Chip Carrier, and 44-Lead Quad Flat Pack Packages**
  (See Packaging Spec., Order #240800, Package Type P, N, and S)

The UPI-C42 is an enhanced CHMOS version of the industry standard Intel UPI-42 family. It is fabricated on Intel's CHMOS III-E process. The UPI-C42 is pin, software, and architecturally compatible with the NMOS UPI family. The UPI-C42 has all of the same features of the NMOS family plus a larger user programmable memory array (4K), hardware A20 gate support, and lower power consumption inherent to a CHMOS product.

The UPI-L42 offers the same functionality and socket compatibility as the UPI-C42 as well as providing low voltage 3.3V operation.

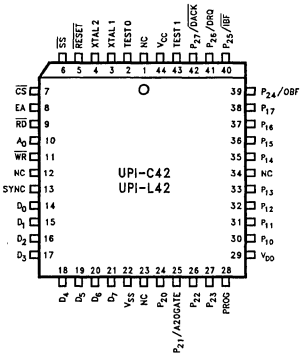
The UPI-C42 is essentially a "slave" microcontroller, or a microcontroller with a slave interface included on the chip. Interface registers are included to enable the UPI device to function as a slave peripheral controller in the MCS Modules and iAPX family, as well as other 8-, 16-, and 32-bit systems.

To allow full user flexibility, the program memory is available in ROM and One-Time Programmable EPROM (OTP).



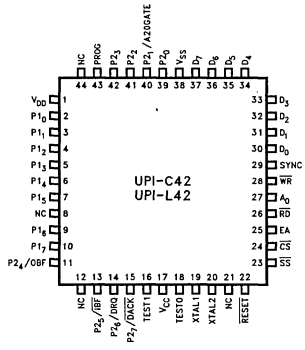
290414-1

Figure 1. DIP Pin Configuration



290414-2

Figure 2. PLCC Pin Configuration



290414-3

Figure 3. QFP Pin Configuration

Table 1. Pin Description

Symbol	DIP Pin No.	PLCC Pin No.	QFP Pin No.	Type	Name and Function
TEST 0, TEST 1	1 39	2 43	18 16	I	**TEST INPUTS:** Input pins which can be directly tested using conditional branch instructions. **FREQUENCY REFERENCE:** TEST 1 (T_1) functions as the event timer input (under software control). TEST 0 (T_0) is a multi-function pin used during PROM programming and ROM/EPROM verification, during Sync Mode to reset the instruction state to S1 and synchronize the internal clock to PH1.
XTAL 1	2	3	19	O	**OUTPUT:** Output from the oscillator amplifier.
XTAL 2	3	4	20	I	**INPUT:** Input to the oscillator amplifier and internal clock generator circuits.
$\overline{\text{RESET}}$	4	5	22	I	**RESET:** Input used to reset status flip-flops, set the program counter to zero, and force the UPI-C42 from the suspend power down mode. $\overline{\text{RESET}}$ is also used during EPROM programming and verification.
$\overline{\text{SS}}$	5	6	23	I	**SINGLE STEP:** Single step input used in conjunction with the SYNC output to step the program through each instruction (EPROM). This should be tied to +5V when not used. This pin is also used to put the device in Sync Mode by applying 12.5V to it.
$\overline{\text{CS}}$	6	7	24	I	**CHIP SELECT:** Chip select input used to select one UPI microcomputer out of several connected to a common data bus.
EA	7	8	25	I	**EXTERNAL ACCESS:** External access input which allows emulation, testing and ROM/EPROM verification. This pin should be tied low if unused.
$\overline{\text{RD}}$	8	9	26	I	**READ:** I/O read input which enables the master CPU to read data and status words from the OUTPUT DATA BUS BUFFER or status register.
A_0	9	10	27	I	**COMMAND/DATA SELECT:** Address Input used by the master processor to indicate whether byte transfer is data (A_0 = 0, F1 is reset) or command (A_0 = 1, F1 is set). A_0 = 0 during program and verify operations.
$\overline{\text{WR}}$	10	11	28	I	**WRITE:** I/O write input which enables the master CPU to write data and command words to the UPI INPUT DATA BUS BUFFER.
SYNC	11	13	29	O	**OUTPUT CLOCK:** Output signal which occurs once per UPI instruction cycle. SYNC can be used as a strobe for external circuitry; it is also used to synchronize single step operation.
D_0–D_7 (BUS)	12–19	14–21	30–37	I/O	**DATA BUS:** Three-state, bidirectional DATA BUS BUFFER lines used to interface the UPI microcomputer to an 8-bit master system data bus.
P_{10}–P_{17}	27–34	30–33 35–38	2–10	I/O	**PORT 1:** 8-bit, PORT 1 quasi-bidirectional I/O lines. P_{10}–P_{17} access the signature row and security bit.

Table 1. Pin Description (Continued)

Symbol	DIP Pin No.	PLCC Pin No.	QFP Pin No.	Type	Name and Function
P_{20}–P_{27}	21–24 35–38	24–27 39–42	39–42 11, 13–15	I/O	**PORT 2:** 8-bit, PORT 2 quasi-bidirectional I/O lines. The lower 4 bits (P_{20}–P_{23}) interface directly to the 8243 I/O expander device and contain address and data information during PORT 4–7 access. P_{21} can be programmed to provide hardware A20 gate support. The upper 4 bits (P_{24}–P_{27}) can be programmed to provide interrupt Request and DMA Handshake capability. Software control can configure P_{24} as Output Buffer Full (OBF) interrupt, P_{25} as Input Buffer Full ($\overline{IBF}$) interrupt, P_{26} as DMA Request (DRQ), and P_{27} as DMA ACKnowledge ($\overline{DACK}$).
PROG	25	28	43	I/O	**PROGRAM:** Multifunction pin used as the program pulse input during PROM programming. During I/O expander access the PROG pin acts as an address/data strobe to the 8243. This pin should be tied high if unused.
V_{CC}	40	44	17		**POWER:** +5V main power supply pin.
V_{DD}	26	29	1		**POWER:** +5V during normal operation. +12.75V during programming operation. Low power standby supply pin.
V_{SS}	20	22	38		**GROUND:** Circuit ground potential.

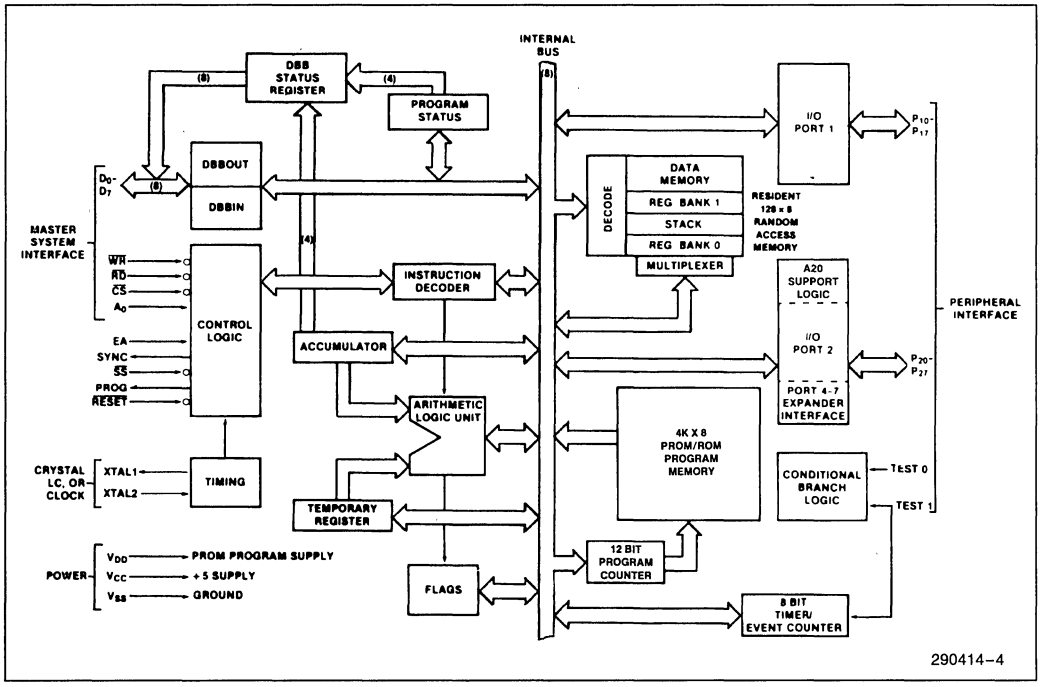


Figure 4. Block Diagram

UPI-C42/L42 PRODUCT SELECTION GUIDE

UPI-C42: Low power CHMOS version of the UPI-42.

Device	Package	ROM	OTP	Comments
80C42	N, P S	4K		ROM Device
82C42PC	N, P, S			Phoenix MultiKey/42 firmware, PS/2 style mouse support
82C42PD	N, P, S			Phoenix MultiKey/42L firmware, KBC and SCC for portable apps.
82C42PE	N, P, S			Phoenix MultiKey/42G firmware, Energy Efficient KBC solution
87C42	N, P, S		4K	One Time Programmable Version

UPI-L42: The low voltage 3.3V version of the UPI-C42.

Device	Package	ROM	OTP	Comments
80L42	N, P S	4K		ROM Device
82L42PC	N, P, S			Phoenix MultiKey/42 firmware, PS/2 style mouse support
82L42PD	N, P, S			Phoenix MultiKey/42L firmware, KBC and SCC for portable apps.
87L42	N, P, S		4K	One Time Programmable Version

N = 44 lead PLCC, P = 40 lead PDIP, S = 44 lead QFP, D = 40 lead CERDIP
KBC = Key Board Control, SCC = Scan Code Control

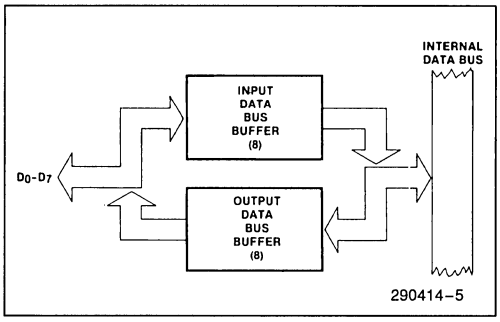
THE INTEL 82C42

As shown in the UPI-C42 product matrix, the UPI-C42 is offered as a pre-programmed 80C42 with various versions of MultiKey/42 keyboard controller firmware developed by Phoenix Technologies Ltd.

The 82C42PC provides a low powered solution for industry standard keyboard and PS/2 style mouse control. The 82C42PD provides a cost effective means for keyboard and scan code control for notebook platforms. The 82C42PE allows a quick time to market, low cost solution for energy efficient desktop designs.

UPI-42 COMPATIBLE FEATURES

1. Two Data Bus Buffers, one for input and one for output. This allows a much cleaner Master/Slave protocol.



2. 8 Bits of Status

ST_7	ST_6	ST_5	ST_4	F_1	F_0	IBF	OBF
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

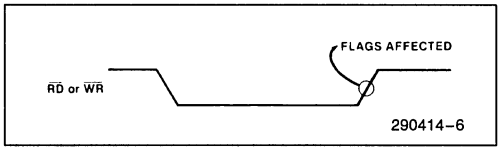
ST_4–ST_7 are user definable status bits. These bits are defined by the "MOV STS, A" single byte, single cycle instruction. Bits 4–7 of the acccumulator are moved to bits 4–7 of the status register. Bits 0–3 of the status register are not affected.

MOV STS, A Op Code: 90H

1	0	0	1	0	0	0	0
D_7							D_0

3. $\overline{RD}$ and $\overline{WR}$ are edge triggered. IBF, OBF, F_1 and INT change internally after the trailing edge of $\overline{RD}$ or $\overline{WR}$.

   During the time that the host CPU is reading the status register, the UPI is prevented from updating this register or is 'locked out.'

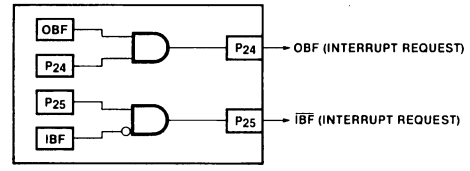


4. P_{24} and P_{25} are port pins or Buffer Flag pins which can be used to interrupt a master processor. These pins default to port pins on Reset.

   If the "EN FLAGS" instruction has been executed, P_{24} becomes the OBF (Output Buffer Full) pin. A "1" written to P_{24} enables the OBF pin (the pin outputs the OBF Status Bit). A "0" written to P_{24} disables the OBF pin (the pin remains low). This pin can be used to indicate that valid data is available from the UPI (in Output Data Bus Buffer).

   If "EN FLAGS" has been executed, P_{25} becomes the $\overline{IBF}$ (Input Buffer Full) pin. A "1" written to P_{25} enables the $\overline{IBF}$ pin (the pin outputs the inverse of the IBF Status Bit. A "0" written to P_{25} disables the $\overline{IBF}$ pin (the pin remains low). This pin can be used to indicate that the UPI is ready for data.

Data Bus Buffer Interrupt Capability



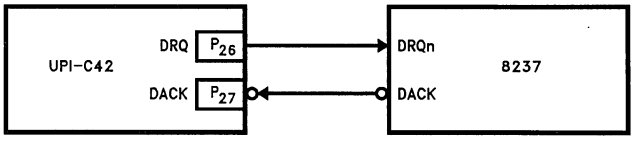
EN FLAGS Op Code: 0F5H

1	1	1	1	0	1	0	1
D_7							D_0

5. P_{26} and P_{27} are port pins or DMA handshake pins for use with a DMA controller. These pins default to port pins on Reset.

   If the "EN DMA" instruction has been executed, P_{26} becomes the DRQ (DMA Request) pin. A "1" written to P_{26} causes a DMA request (DRQ is activated). DRQ is deactivated by DACK•RD, DACK•WR, or execution of the "EN DMA" instruction.

DMA Handshake Capability



If "EN DMA" has been executed, P_{27} becomes the $\overline{DACK}$ (DMA ACKnowledge) pin. This pin acts as a chip select input for the Data Bus Buffer registers during DMA transfers.

EN DMA Op Code: 0E5H

1	1	1	0	0	1	0	1

D_7 D_0

6. When EA is enabled on the UPI, the program counter is placed on Port 1 and the lower four bits of Port 2 (MSB = P_{23}, LSB = P_{10}). On the UPI this information is multiplexed with PORT DATA (see port timing diagrams at end of this data sheet).
7. The UPI-C42 supports the Quick Pulse Programming Algorithm, but can also be programmed with the Intelligent Programming Algorithm. (See the Programming Section.)

UPI-C42 FEATURES

Programmable Memory Size Increase

The user programmable memory on the UPI-C42 will be increased from the 2K available in the NMOS product by 2X to 4K. The larger user programmable memory array will allow the user to develop more complex peripheral control micro-code. P2.3 (port 2 bit 3) has been designated as the extra address pin required to support the programming of the extra 2K of user programmable memory.

The new instruction SEL PMB1 (73h) allows for access to the upper 2K bank (locations 2048–4095). The additional memory is completely transparent to users not wishing to take advantage of the extra memory space. No new commands are required to access the lower 2K bytes. The SEL PMB0 (63h) has also been added to the UPI-C42 instruction set to allow for switching between memory banks.

Extended Memory Program Addressing (Beyond 2K)

For programs of 2K words or less, the UPI-C42 addresses program memory in the conventional manner. Addresses beyond 2047 can be reached by executing a program memory bank switch instruction (SEL PMB0, SEL PMB1) followed by a branch instruction (JMP or CALL). The bank switch feature extends the range of branch instructions beyond their normal 2K range and at the same time prevents the user from inadvertently crossing the 2K boundary.

PROGRAM MEMORY BANK SWITCH

The switching of 2K program memory banks is accomplished by directly setting or resetting the most significant bit of the program counter (bit 11); see Figure 5. Bit 11 is not altered by normal incrementing of the program counter, but is loaded with the contents of a special flip-flop each time a JMP or CALL instruction is executed. This special flip-flop is set by executing an SEL PMB1 instruction and reset by SEL PMB0. Therefore, the SEL PMB instruction may be executed at any time prior to the actual bank switch which occurs during the next branch instruction encountered. Since all twelve bits of the program counter, including bit 11, are stored in the stack, when a Call is executed, the user may jump to subroutines across the 2K boundary and the proper PC will be restored upon return. However, the bank switch flip-flop will not be altered on return.

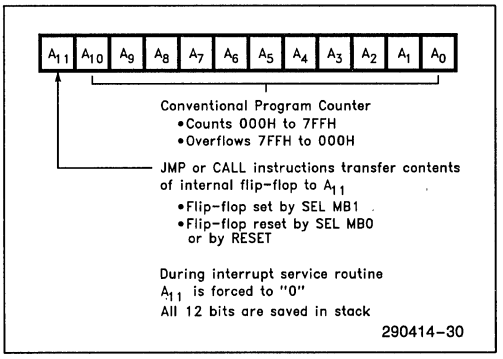


Figure 5. Program Counter

INTERRUPT ROUTINES

Interrupts always vector the program counter to location 3 or 7 in the first 2K bank, and bit 11 of the program counter is held at "0" during the interrupt service routine. The end of the service routine is signaled by the execution of an RETR instruction. Interrupt service routines should therefore be contained entirely in the lower 2K words of program memory. The execution of a SEL PMB0 or SEL PMB1 instruction within an interrupt routine is not recommended since it will not alter PC11 while in the routine, but will change the internal flip-flop.

Hardware A20 Gate Support

This feature has been provided to enhance the performance of the UPI-C42 when being used in a keyboard controller application. The UPI-C42 design has included on chip logic to support a hardware GATEA20 feature which eliminates the need to provide firmware to process A20 command sequences,

thereby providing additional user programmable memory space. This feature is enabled by the A20EN instruction and remains enabled until the device is reset. It is important to note that the execution of the A20EN instruction redefines Port 2, bit 1 as a pure output pin with read only characteristics. The state of this pin can be modified only through a valid "D1" command sequence (see Table 1). Once enabled, the A20 logic will process a "D1" command sequence (write to output port) by setting/resetting the A20 bit on port 2, bit 1 (P2.1) without requiring service from the internal CPU. The host can directly control the status of the A20 bit. At no time during this host interface transaction will the IBF flag in the status register be activated. Table 1 gives several possible GATEA20 command/data sequences and UPI-C42 responses.

Table 1. D1 Command Sequences

A0	R/W	DB Pins	IBF	A20	Comments
1	W	D1h	0	n(1)	Set A20 Sequence
0	W	DFh	0	1	Only DB1 Is Processed
1	W	FFH(2)	0	n	
1	W	D1h	0	n	Clear A20 Sequence
0	W	DDh	0	0	
1	W	FFh	0	n	
1	W	D1h	0	n	Double Trigger Set
1	W	D1h	0	n	Sequence
0	W	DFh	0	1	
1	W	FFh	0	n	
1	W	D1h	0	n	Invalid Sequence
1	W	XXh(3)	1	n	No Change in State
0	W	DDh	1	n	of A20 Bit

NOTES:
1. Indicates that P2.1 remains at the previous logic level.
2. Only FFh commands in a valid A20 sequence have no effect on IBF. An FFh issued at any other time will activate IBF.
3. Any command except D1.

The above sequences assume that the GATEA20 logic has been enabled via the A20EN instruction. As noted, only the value on DB 1 (data bus, bit 1) is processed. This bit will be directly passed through to P2.1 (port 2, bit 1).

SUSPEND

The execution of the suspend instruction (82h or E2h) causes the UPI-C42 to enter the suspend mode. In this mode of operation the oscillator is not running and the internal CPU operation is stopped. The UPI-C42 consumes ≤ 40 μA in the suspend mode. This mode can only be exited by RESET. CPU operation will begin from PC = 000h when the UPI-C42 exits from the suspend power down mode.

Suspend Mode Summary

- Oscillator Not Running
- CPU Operation Stopped
- Ports Tristated with Weak ($\sim$2–10 μA) Pull-Up
- Micropower Mode ($I_{CC} \leq 40$ μA)
- This mode is exited by RESET

5

Table 2 covers all suspend mode pin states. In addition to the suspend power down mode, the UPI-C42 will also support the NMOS power down mode as outlined in Chapter 4 of the UPI-42AH users manual.

Table 2. Suspend Mode Pin States

Pins	Suspend
Ports 1 and 2 Outputs Inputs	 Tristate Weak Pull-Up Disabled
DBB(1) Outputs Inputs	 Normal Normal
System Control (RD#, WR#, CS#, A0)	Disabled
Reset#	Enabled
Crystal Osc. (XTAL1, XTAL2)	Disabled
Test 0, Test 1	Disabled
Prog	High
Sync	High
EA	Disabled, No Pull-Up
SS#	Disabled, Weak Pull-Up
I_{CC}	<40 μA

NOTES:
1. DBB outputs are Tristate unless CS# and RD# are active. DBB inputs are disabled unless CS# and WR# are active.
2. A "disabled" input will not cause current to be drawn regardless of input level (within the supply range).
3. Weak pull-ups have current capability of typically 5 μA.

NEW UPI-C42 INSTRUCTIONS

The UPI-C42 will support several new instructions to allow for the use of new C42 features. These instructions are not necessary to the user who does not wish to take advantage of any new C42 functionality. The C42 will be completely compatible with all current NMOS code/applications. In order to use new features, however, some code modifications will be necessary. All new instructions can easily be inserted into existing code by use of the ASM-48 macro facility as shown in the following example:

```
Macname MACRO
        DB  63H
        ENDM
```

New Instructions

The following is a list of additions to the UPI-42 instruction set. These instructions apply only to the UPI-C42. These instructions must be added to existing code in order to use any new functionality.

SEL PMB0 Select Program Memory Bank 0

```
OPCODE  0110 0011 (63h)
```

PC Bit 11 is set to zero on next JMP or CALL instruction. All references to program memory fall within the range of 0–2047 (0–7FFh).

SEL PMB1 Select Program Memory Bank 1

```
OPCODE  0111 0011 (73h)
```

PC Bit 11 is set to one on next JMP or CALL instruction. All references to program memory fall within the range of 2048–4095 (800h–FFFh).

ENA20 Enables Auto A20 hardware

```
OPCODE  0011 0011 (33h)
```

Enables on chip logic to support Hardware A20 Gate feature. Will remain enabled until device is reset.

This circuitry gives the host direct control of port 2 bit 1 (P2.1) without intervention by the internal CPU. When this opcode is executed, P2.1 becomes a dedicated output pin. The status of this pin is read-able but can only be altered through a valid "D1" command sequence (see Table 1).

SUSPEND Invoke Suspend Power Down Mode

OPCODE 1000 0010 (82h) or 1110 0010 (E2h)

Enables device to enter micro power mode. In this mode the external oscillator is off, CPU operation is stopped, and the Port pins are tristated. This mode can only be exited via a RESET signal.

PROGRAMMING AND VERIFYING THE UPI-C42

The UPI-C42 programming will differ from the NMOS device in three ways. First, the C42 will have a 4K user programmable array. The UPI-C42 will also be programmed using the Intel Quick-Pulse Programming Algorithm. Finally, port 2 bit three (P2.3) will be used during program as the extra address pin required to program the upper 2K bank of additional memory. None of these differences have any effect on the full CHMOS to NMOS device compatibility. The extra memory is fully transparent to the user who does not need, or want, to use the extra memory space of the UPI-C42.

In brief, the programming process consists of: activating the program mode, applying an address, latching the address, applying data, and applying a programming pulse. Each word is programmed completely before moving on to the next and is followed by a verification step. The following is a list of the pins used for programming and a description of their functions:

Pin	Function
XTAL 2	Clock Input
$\overline{\text{Reset}}$	Initialization and Address Latching
Test 0	Selection of Program or Verify Mode
EA	Activation of Program/Verify Signature Row/Security Bit Modes
BUS	Address and Data Input Data Output During Verify
P_{20-23}	Address Input
V_{DD}	Programming Power Supply
PROG	Program Pulse Input

WARNING
An attempt to program a missocketed UPI-C42 will result in severe damage to the part. An indication of a properly socketed part is the appearance of the SYNC clock output. The lack of this clock may be used to disable the programmer.

The Program/Verify sequence is:

1. Insert 87C42 in programming socket
2. CS = 5V, V_{CC} = 5V, V_{DD} = 5V, RESET = 0V, A_0 = 0V, TEST 0 = 5V, clock applied or internal oscillator operating, BUS floating, PROG = 5V.
3. TEST 0 = 0V (select program mode)
4. EA = 12.75V (active program mode)
5. V_{CC} = 6.25V (programming supply)
6. V_{DD} = 12.75V (programming power)
7. Address applied to BUS and P_{20-23}
8. $\overline{\text{RESET}}$ = 5V (latch address)
9. Data applied to BUS
10. PROG = 5V followed by one 100 μs pulse to 0V
11. TEST 0 = 5V (verify mode)
12. Read and verify data on BUS
13. TEST 0 = 0V
14. $\overline{\text{RESET}}$ = 0V and repeat from step 6
15. Programmer should be at conditions of step 1 when the 87C42 is removed from socket

Please follow the Quick-Pulse Programming flow chart for proper programming procedure shown in Figure 6.

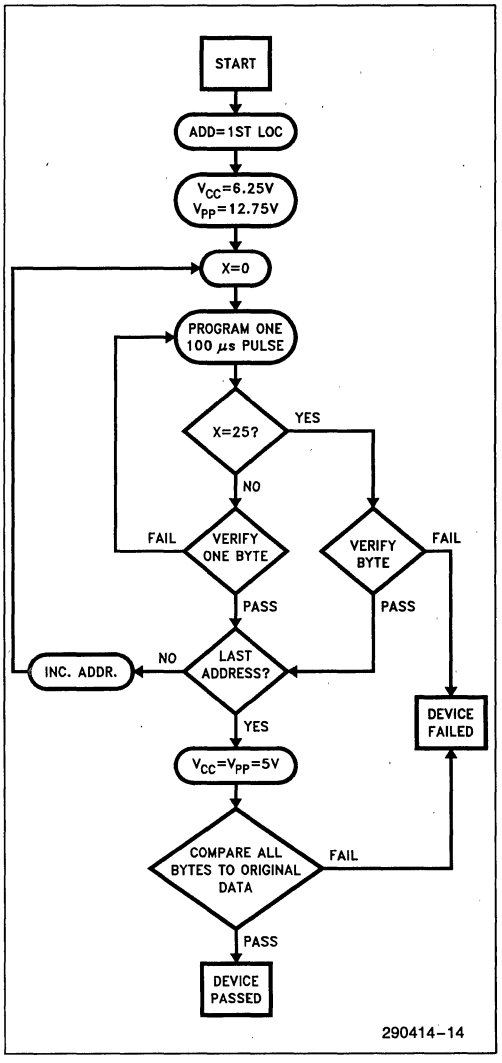


Figure 6. Quick-Pulse Programming Algorithm

Quick-Pulse Programming Algorithm

As previously stated, the UPI-C42 will be programmed using the Quick-Pulse Programming Algorithm, developed by Intel to substantially reduce the thorughput time in production programming.

The Quick-Pulse Programming Algorithm uses initial pulses of 100 μs followed by a byte verification to determine when the address byte has been successfully programmed. Up to 25 100 μs pulses per byte are provided before a failure is recognized. A flow chart of the Quick-Pulse Programming Algorithm is shown in Figure 6.

The entire sequence of program pulses and byte verifications is performed at V_{CC} = 6.25V and V_{DD} = 12.75V. When programming has been completed, all bytes should be compared to the original data with V_{CC} = V_{DD} = 5V.

A verify should be performed on the programmed bits to ensure that they have been correctly programmed. The verify is performed with T0 = 5V, V_{DD} = 5V, EA = 12.75V, SS# = 5V, PROG = 5V, A0 = 0V, and CS# = 5V.

In addition to the Quick-Pulse Programming Algorithm, the UPI-C42 OPT is also compatible with Intel's Int$_e$ligent Programming Algorithm which is used to program the NMOS UPI-42AH OTP devices.

The entire sequence of program pulses and byte verifications is performed at V_{CC} = 6.25V and V_{DD} = 12.75V. When the int$_e$ligent Programming cycle has been completed, all bytes should be compared to the original data with V_{CC} = 5.0, V_{DD} = 5V.

Verify

A verify should be performed on the programmed bits to determine that they have been correctly programmed. The verify is performed with T0 = 5V, V_{DD} = 5V, EA = 12.75V, $\overline{SS}$ = 5V, PROG = 5V, A0 = 0V, and $\overline{CS}$ = 5V.

SECURITY BIT

The security bit is a single EPROM cell outside the EPROM array. The user can program this bit with the appropriate access code and the normal programming procedure, to inhibit any external access to the EPROM contents. Thus the user's resident program is protected. There is no direct external access to this bit. However, the security byte in the signature row has the same address and can be used to check indirectly whether the security bit has been programmed or not. The security bit has no effect on the signature mode, so the security byte can always be examined.

SECURITY BIT PROGRAMMING/ VERIFICATION

Programming

a. Read the security byte of the signature mode. Make sure it is 00H.

b. Apply access code to appropriate inputs to put the device into security mode.

c. Apply high voltage to EA and V_{DD} pins.

d. Follow the programming procedure as per the Quick-Pulse Programming Algorithm with known data on the databus. Not only the security bit, but also the security byte of the signature row is programmed.

e. Verify that the security byte of the signature mode contains the same data as appeared on the data bus. (If DB0–DB7 = high, the security byte will contain FFH.)

f. Read two consecutive known bytes from the EPROM array and verify that the wrong data are retrieved in at least one verification. If the EPROM can still be read, the security bit may have not been fully programmed though the security byte in the signature mode has.

Verification

Since the security bit address overlaps the address of the security byte of the signature mode, it can be used to check indirectly whether the security bit has been programmed or not. Therefore, the security bit verification is a mere read operation of the security byte of the signature row (0FFH = security bit programmed; 00H = security bit unprogrammed). Note that during the security bit programming, the reading of the security byte does not necessarily indicate that the security bit has been successfully programmed. Thus, it is recommended that two consecutive known bytes in the EPROM array be read and the wrong data should be read at least once, because it is highly improbable that random data coincides with the correct ones twice.

SIGNATURE MODE

The UPI-C42 has an additional 64 bytes of EPROM available for Intel and user signatures and miscellaneous purposes. The 64 bytes are partitioned as follows:

A. **Test code/checksum**—This can accommodate up to 25 bytes of code for testing the internal nodes that are not testable by executing from the external memory. The test code/checksum is present on ROMs, and OTPs.

B. **Intel signature**—This allows the programmer to read from the UPI-41AH/42AH/C42 the manufacturer of the device and the exact product name. It facilitates automatic device identification and will be present in the ROM and OTP versions. Location 10H contains the manufacturer code. For Intel, it is 89H. Location 11H contains the device code.

   The code is 43H and 42H for the 8042AH/80C42 and OTP 8742AH/87C42, respectively. The code is 44H for any device with the security bit set by Intel.

C. **User signature**—The user signature memory is implemented in the EPROM and consists of 2 bytes for the customer to program his own signature code (for identification purposes and quick sorting of previously programmed materials).

D. **Test signature**—This memory is used to store testing information such as: test data, bin number, etc. (for use in quality and manufacturing control).

E. **Security byte**—This byte is used to check whether the security bit has been programmed (see the security bit section).

F. **UPI-C42 Intel Signature**—Applies only to CHMOS device. Location 20H contains the manufacturer code and location 21H contains the device code. The Intel UPI-C42 manufacturer's code is 99H. The device ID's are 82H for the OTP version and 83H for the ROM version. The device ID's are the same for the UPI-L42.

The signature mode can be accessed by setting P10 = 0, P11–P17 = 1, and then following the programming and/or verification procedures. The location of the various address partitions are as shown in Table 3.

SYNC MODE

The Sync Mode is provided to ease the design of multiple controller circuits by allowing the designer to force the device into known phase and state time. The Sync Mode may also be utilized by automatic test equipment (ATE) for quick, easy, and efficient synchronizing between the tester and the DUT (device under test).

Sync Mode is enabled when $\overline{SS}$ pin is raised to high voltage level of +12 volts. To begin synchronization, T0 is raised to 5 volts at least four clock cycles after $\overline{SS}$. T0 must be high for at least four X2 clock cycles to fully reset the prescaler and time state generators. T0 may then be brought down during low state of X2. Two clock cycles later, with the rising edge of X2, the device enters into Time State 1, Phase 1. $\overline{SS}$ is then brought down to 5 volts 4 clocks later after T0. RESET is allowed to go high 5 tCY (75 clocks) later for normal execution of code.

Table 3. Signature Mode Table

	Address	Device Type	No. of Bytes
Test Code/Checksum	0 0FH 16H 1EH	ROM/OTP	25
Intel Signature	10H 11H	ROM/OTP	2
User Signature	12H 13H	OTP	2
Test Signature	14H 15H	ROM/OTP	2
Security Byte	1FH or 3FH	ROM/OTP	2
UPI-C42 Intel Signature	20H 21H	ROM/OTP	2
User Defined UPI-C42 OTP EPROM Space	22H 3EH	ROM/OTP	30

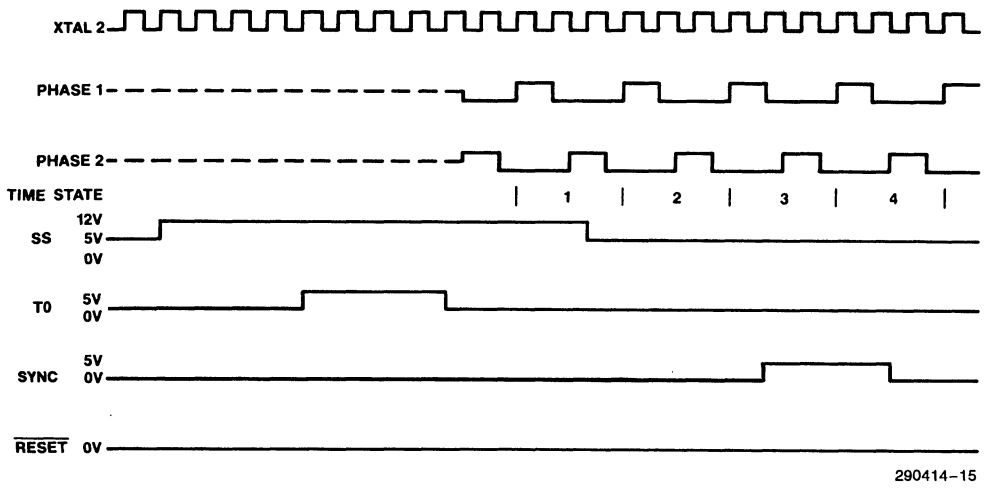
ACCESS CODE

The following table summarizes the access codes required to invoke the Sync Mode, Signature Mode, and the Security Bit, respectively. Also, the programming and verification modes are included for comparison.

Modes		Control Signals							Data Bus								Port 2	Access Code Port 1							
		T0	RST	SS	EA	PROG	V_{DD}	V_{CC}	0	1	2	3	4	5	6	7	0 1 2 3	0	1	2	3	4	5	6	7
Programming Mode		0	0	1	HV	1	V_{DDH}	V_{CC}	Address								Addr	a_0	a_1	X	X	X	X	X	X
		0	1	1	HV	STB	V_{DDH}	V_{CC}	Data In								Addr								
Verification Mode		0	0	1	HV	1	V_{CC}	V_{CC}	Address								Addr	a_0	a_1	X	X	X	X	X	X
		1	1	1	HV	1	V_{CC}	V_{CC}	Data Out								Addr								
Sync Mode		STB High	0	HV	0	X	V_{CC}	V_{CC}	X	X	X	X	X	X	X	X	X X X	X	X	X	X	X	X	X	X
Signature Mode	Prog	0	0	1	HV	1	V_{DDH}	V_{CC}	Addr. (see Sig Mode Table)								0 0 0	0	1	1	1	1	X	X	1
		0	1	1	HV	STB	V_{DDH}	V_{CC}	Data In								0 0 0								
	Verify	0	0	1	HV	1	V_{CC}	V_{CC}	Addr. (see Sig Mode Table)								0 0 0								
		1	1	1	HV	1	V_{CC}	V_{CC}	Data Out								0 0 0								
Security Bit/Byte	Prog	0	0	1	HV	1	V_{DDH}	V_{CC}	Address								0 0 0								
		0	1	1	HV	STB	V_{DDH}	V_{CC}	Data In								0 0 0								
	Verify	0	0	1	HV	1	V_{CC}	V_{CC}	Address								0 0 0								
		1	1	1	HV	1	V_{CC}	V_{CC}	Data Out								0 0 0								

NOTE:
1. a_0 = 0 or 1; a_1 = 0 or 1. a_0 must = a_1.

SYNC MODE TIMING DIAGRAMS



Minimum Specifications
SYNC Operation Time, t_{SYNC} = 3.5 XTAL 2 Clock cycles. Reset Time, t_{RS} = 4 t_{CY}.

NOTE:
The rising and falling edges of T0 should occur during low state of XTAL 2 clock.

APPLICATIONS

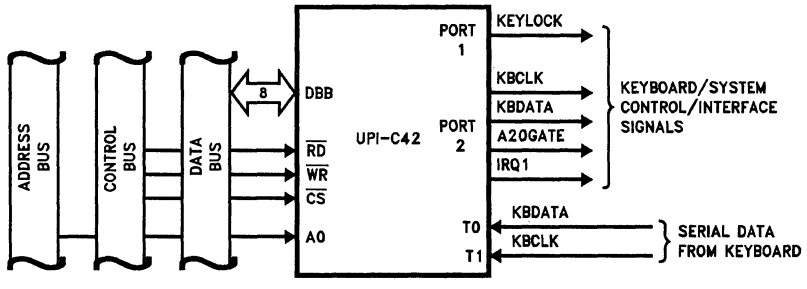


Figure 7. UPI-C42 Keyboard Controller

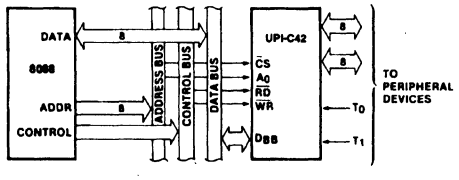


Figure 8. 8088-UPI-C42 Interface

APPLICATIONS (Continued)

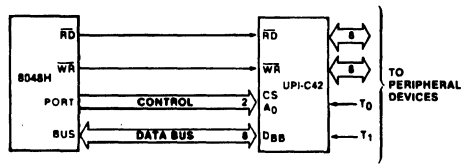


Figure 9. 8048H-UPI-C42 Interface

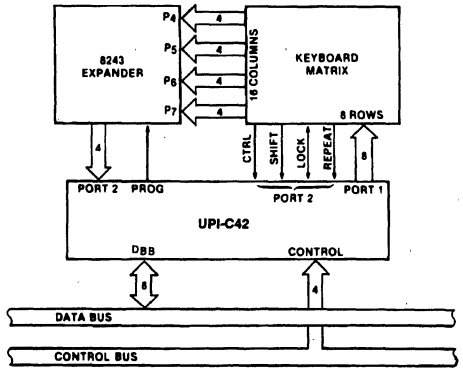


Figure 10. UPI-C42-8243 Keyboard Scanner

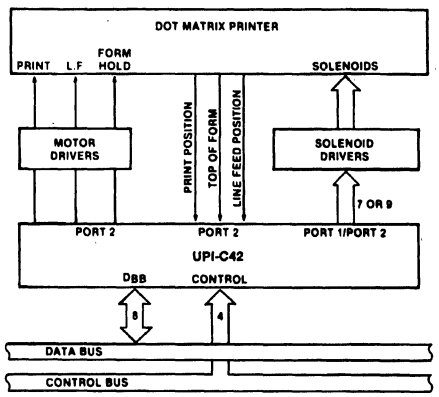


Figure 11. UPI-C42 80-Column Matrix Printer Interface

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias . . . . 0°C to +70°C
Storage Temperature . . . . . −65°C to +150°C
Voltage on Any Pin with
Respect to Ground . . . . . −0.5V to +7V
Power Dissipation . . . . . 1.5 W

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**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.*

DC CHARACTERISTICS T_A = 0°C to +70°C, V_{CC} = V_{DD} = +5V ±10%; +3.3V ±10% UPI-L42

Symbol	Parameter	UPI-C42		UPI-L42		Units	Notes
		Min	Max	Min	Max		
V_{IL}	Input Low Voltage	−0.5	0.8	−0.3	+0.8	V	All Pins
V_{IH}	Input High Voltage (Except XTAL2, RESET)	2.0	V_{CC}	2.0	V_{CC} + 0.3	V	
V_{IH1}	Input High Voltage (XTAL2, RESET)	3.5	V_{CC}	2.0	V_{CC} + 0.3	V	
V_{OL}	Output Low Voltage (D_0–D_7)		0.45		0.45	V	I_{OL} = 2.0 mA UPI-C42 I_{OL} = 1.3 mA UPI-L42
V_{OL1}	Output Low Voltage ($P_{10}P_{17}$, $P_{20}P_{27}$, Sync)		0.45		0.45	V	I_{OL} = 1.6 mA UPI-C42 I_{OL} = 1 mA UPI-L42
V_{OL2}	Output Low Voltage (PROG)		0.45		0.45	V	I_{OL} = 1.0 mA UPI-C42 I_{OL} = 0.7 mA UPI-L42
V_{OH}	Output High Voltage (D_0–D_7)	2.4		2.4		V	I_{OH} = −400 μA UPI-C42 I_{OH} = −260 μA UPI-L42
V_{OH1}	Output High Voltage (All Other Outputs)	2.4		2.4			I_{OH} = −50 μA UPI-C42 I_{OH} = −25 μA UPI-L42
I_{IL}	Input Leakage Current (T_0, T_1, RD, WR, CS, A_0, EA)		±10		±10	μA	$V_{SS} \le V_{IN} \le V_{CC}$
I_{OFL}	Output Leakage Current (D_0–D_7, High Z State)		±10		±10	μA	V_{SS} + 0.45 ≤ $V_{OUT} \le V_{CC}$
I_{LI}	Low Input Load Current ($P_{10}P_{17}$, $P_{20}P_{27}$)	−50	−250	−35	−175	μA	Port Pins Min V_{IN} = 2.4V Max V_{IN} = 0.45V
I_{LI1}	Low Input Load Current (RESET, SS)		−40		−40	μA	$V_{IN} \le V_{IL}$
I_{HI}	Port Sink Current ($P_{10}P_{17}$, $P_{20}P_{27}$)				5.0	mA	V_{CC} = 3.0V V_{IH} = 5.0V
I_{DD}	V_{DD} Supply Current		4		2.5	mA	

5

DC CHARACTERISTICS

T_A = 0°C to +70°C, V_{CC} = V_{DD} = +5V ±10%; +3.3V ±10% UPI-L42 (Continued)

Symbol	Parameter	UPI-C42		UPI-L42		Units	Notes
		Min	Max	Min	Max		
I_{CC} + I_{DD}	Total Supply Current: Active Mode @ 12.5 MHz		30		20	mA	Typical 14 mA UPI-C42, 9 mA UPI-L42
	Suspend Mode		40		26	μA	Osc. Off(1, 4)
I_{DD} Standby	Power Down Supply Current		5		3.5	mA	NMOS Compatible Power Down Mode
I_{IH}	Input Leakage Current (P_{10}–P_{17}, P_{20}–P_{27})		100		100	μA	V_{IN} = V_{CC}
C_{IN}	Input Capacitance		10		10	pF	T_A = 25°C (1)
C_{IO}	I/O Capacitance		20		20	pF	T_A = 25°C (1)

NOTE:
1. Sampled, not 100% tested.

DC CHARACTERISTICS—PROGRAMMING (UPI-C42 AND UPI-L42)

T_A = 25°C ±5°C, V_{CC} = 6.25V ±0.25V, V_{DD} = 12.75V ±0.25V

Symbol	Parameter	Min	Max	Units
V_{DDH}	V_{DD} Program Voltage High Level	12.5	13	V(1)
V_{DDL}	V_{DD} Voltage Low Level	4.75	5.25	V
V_{PH}	PROG Program Voltage High Level	2.0	5.5	V
V_{PL}	PROG Voltage Low Level	−0.5	0.8	V
V_{EAH}	Input High Voltage for EA	12.0	13.0	V(2)
V_{EAL}	EA Voltage Low Level	−0.5	5.25	V
I_{DD}	V_{DD} High Voltage Supply Current		50.0	mA
I_{EA}	EA High Voltage Supply Current		1.0	mA(4)

NOTES:
1. Voltages over 13V applied to pin V_{DD} will permanently damage the device.
2. V_{EAH} must be applied to EA before V_{DDH} and removed after V_{DDL}.
3. V_{CC} must be applied simultaneously or before V_{DD} and must be removed simultaneously or after V_{DD}.
4. Sampled, not 100% tested.

AC CHARACTERISTICS

T_A = 0°C to +70°C, V_{SS} = 0V, V_{CC} = V_{DD} = +5V ±10%; +3.3V ±10% for the UPI-L42

NOTE:
All AC Characteristics apply to both the UPI-C42 and UPI-L42

DBB READ

Symbol	Parameter	Min	Max	Units
t_{AR}	CS, A_0 Setup to RD ↓	0		ns
t_{RA}	CS, A_0 Hold After RD ↑	0		ns
t_{RR}	RD Pulse Width	160		ns
t_{AD}	CS, A_0 to Data Out Delay		130	ns
t_{RD}	RD ↓ to Data Out Delay	0	130	ns
t_{DF}	RD ↑ to Data Float Delay		85	ns

DBB WRITE

Symbol	Parameter	Min	Max	Units
t_{AW}	CS, A_0 Setup to WR ↓	0		ns
t_{WA}	CS, A_0 Hold After WR ↑	0		ns
t_{WW}	WR Pulse Width	160		ns
t_{DW}	Data Setup to WR ↑	130		ns
t_{WD}	Data Hold After WR ↑	0		ns

AC CHARACTERISTICS

T_A = 0°C to +70°C, V_{SS} = 0V, V_{CC} = V_{DD} = +5V ±10%; +3.3V ±10% for the UPI-L42 (Continued)

CLOCK

Symbol	Parameter	Min	Max	Units
t_{CY} UPI-C42/UPI-L42	Cycle Time	1.2	9.20	μs(1)
t_{CYC} UPI-C42/UPI-L42	Clock Period	80	613	ns
t_{PWH}	Clock High Time	30		ns
t_{PWL}	Clock Low Time	30		ns
t_R	Clock Rise Time		10	ns
t_F	Clock Fall Time		10	ns

NOTE:
1. t_{CY} = 15/f(XTAL)

AC CHARACTERISTICS DMA

Symbol	Parameter	Min	Max	Units
t_{ACC}	DACK to WR or RD	0		ns
t_{CAC}	RD or WR to DACK	0		ns
t_{ACD}	DACK to Data Valid	0	130	ns
t_{CRQ}	RD or WR to DRQ Cleared		110	ns(1)

NOTE:
1. C_L = 150 pF.

AC CHARACTERISTICS PORT 2

Symbol	Parameter	f(t_{CY})(3)	Min	Max	Units
t_{CP}	Port Control Setup Before Falling Edge of PROG	1/15 t_{CY} − 28	55		ns(1)
t_{PC}	Port Control Hold After Falling Edge of PROG	1/10 t_{CY}	125		ns(2)
t_{PR}	PROG to Time P2 Input Must Be Valid	8/15 t_{CY} − 16		650	ns(1)
t_{PF}	Input Data Hold Time		0	150	ns(2)
t_{DP}	Output Data Setup Time	2/10 t_{CY}	250		ns(1)
t_{PD}	Output Data Hold Time	1/10 t_{CY} − 80	45		ns(2)
t_{PP}	PROG Pulse Width	6/10 t_{CY}	750		ns

NOTES:
1. C_L = 80 pF.
2. C_L = 20 pF.
3. t_{CY} = 1.25 μs.

AC CHARACTERISTICS—PROGRAMMING (UPI-C42 AND UPI-L42)

T_A = 25°C ±5°C, V_{CC} = 6.25V ±0.25V, V_{DDL} = +5V ±0.25V, V_{DDH} = 12.75V ±0.25V

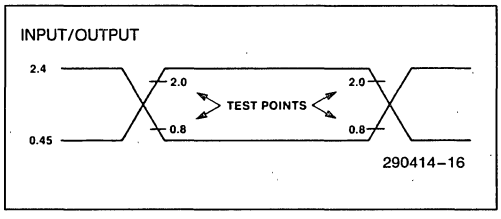
(87C42/87L42 ONLY)

Symbol	Parameter	Min	Max	Units
t_{AW}	Address Setup Time to RESET ↑	$4t_{CY}$		
t_{WA}	Address Hold Time after RESET ↑	$4t_{CY}$		
t_{DW}	Data in Setup Time to PROG ↓	$4t_{CY}$		
t_{WD}	Data in Hold Time after PROG ↑	$4t_{CY}$		
t_{PW}	Initial Program Pulse Width	95	105	μs
t_{TW}	Test 0 Setup Time for Program Mode	$4t_{CY}$		
t_{WT}	Test 0 Hold Time after Program Mode	$4t_{CY}$		
t_{DO}	Test 0 to Data Out Delay		$4t_{CY}$	
t_{WW}	RESET Pulse Width to Latch Address	$4t_{CY}$		
t_r, t_f	PROG Rise and Fall Times	0.5	100	μs
t_{CY}	CPU Operation Cycle Time	2.5	3.75	μs
t_{RE}	RESET Setup Time before EA ↑	$4t_{CY}$		
t_{OPW}	Overprogram Pulse Width	2.85	78.75	ms(1)
t_{DE}	EA High to V_{DD} High	$1t_{CY}$		

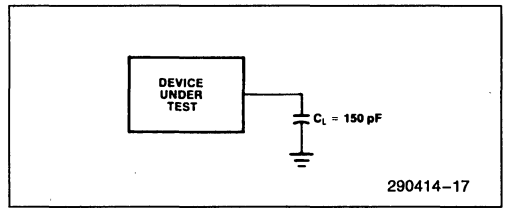
NOTES:
1. This variation is a function of the iteration counter value, X.
2. If TEST 0 is high, t_{DO} can be triggered by RESET ↑.

5

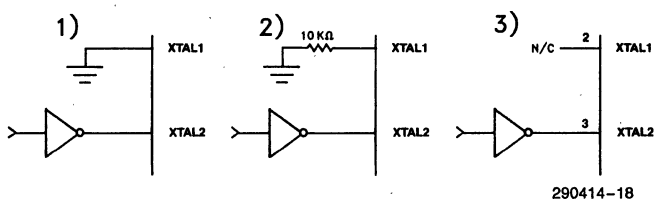
AC TESTING INPUT/OUTPUT WAVEFORM



AC TESTING LOAD CIRCUIT

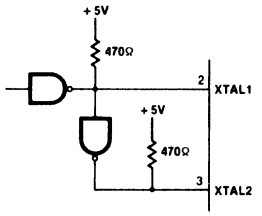


DRIVING FROM AN EXTERNAL SOURCE



290414-18

NOTE:
See XTAL1 Configuration Table.

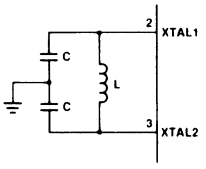


290414-19

Rise and Fall Times Should Not Exceed 10 ns. Resistors to V_{CC} are Needed to Ensure V_{IH} = 3.5V if TTL Circuitry is Used.

LC OSCILLATOR MODE

L	C	NOMINAL
45 H	20 pF	5.2 MHz
120 H	20 pF	3.2 MHz



$$f = \frac{1}{2\pi\sqrt{LC'}}$$

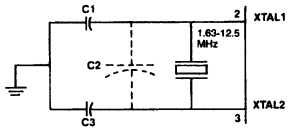
$$C' = \frac{C + 3Cpp}{2}$$

Cpp ≅ 5–10 pF
Pin-to-Pin Capacitance

290414-20

Each C Should be Approximately 20 pF, including Stray Capacitance.

CRYSTAL OSCILLATOR MODE



290414-21

C1 5 pF (STRAY 5 pF)
C2 (CRYSTAL + STRAY) 8 pF
C3 20–30 pF INCLUDING STRAY

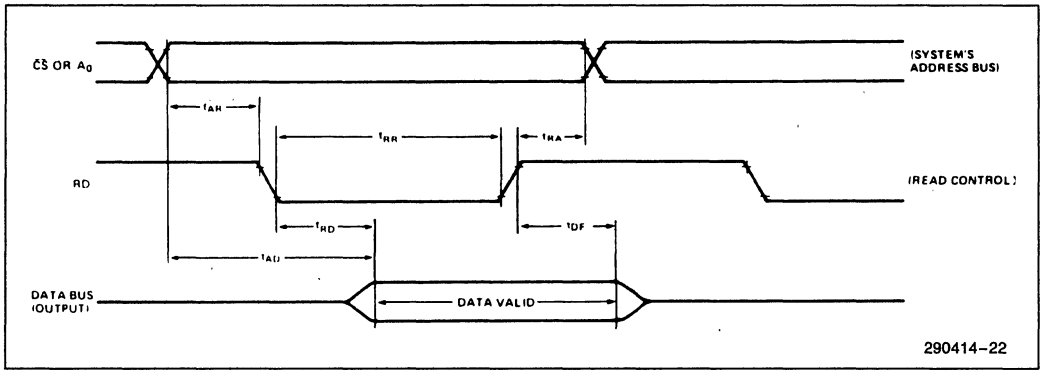
Crystal Series Resistance Should be Less Than 30Ω at 12.5 MHz.

XTAL1 Configuration Table

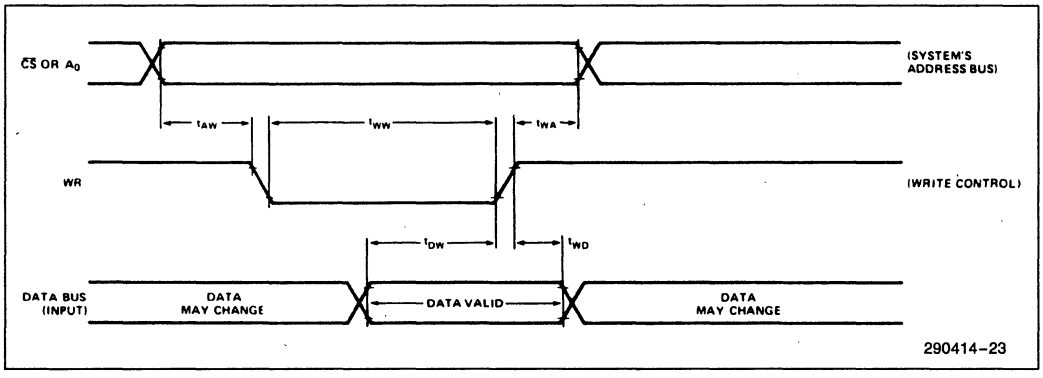
XTAL1 Connection		
1) to Ground	**2) 10 KΩ Resistor to Ground**	**3) Not Connected**
Not recommended for CHMOS designs. Causes approximately 16 mA of additional current flow through the XTAL1 pin on UPI-C42 and approximately 11 mA of additional current through XTAL1 on the UPI-L42.	Recommended configuration for designs which will use both NMOS and CHMOS parts. This configuration limits the additional current through the XTAL1 pin to approximately 1 mA, while maintaining compatibility with the NMOS device.	Low power configuration recommended for CHMOS only designs to provide lowest possible power consumption. This configuration will not work with the NMOS device.

WAVEFORMS

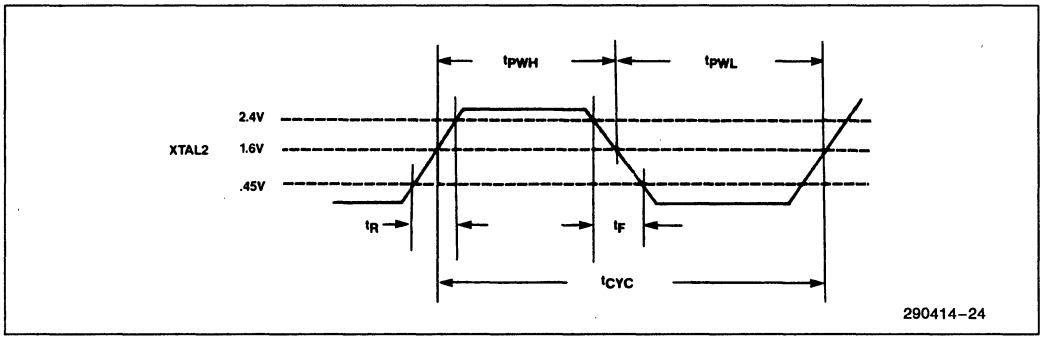
READ OPERATION—DATA BUS BUFFER REGISTER



WRITE OPERATION—DATA BUS BUFFER REGISTER

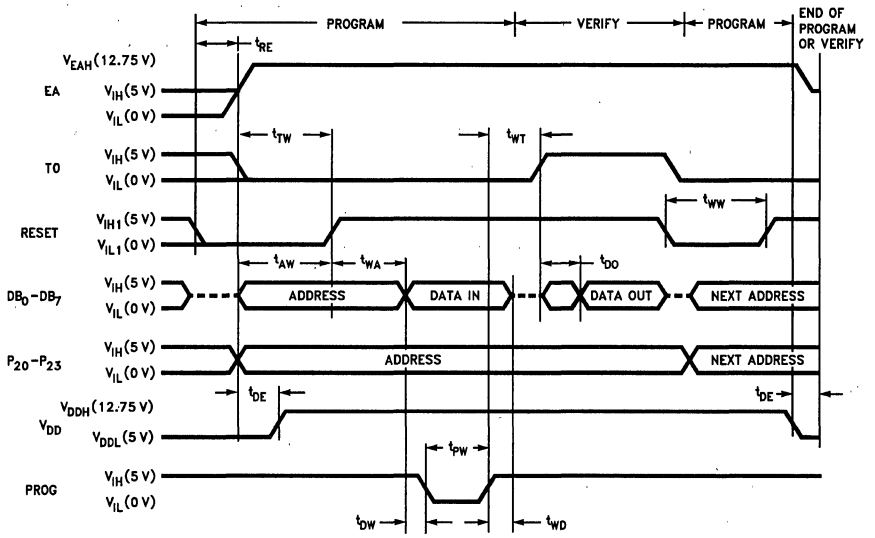


CLOCK TIMING



WAVEFORMS (Continued)

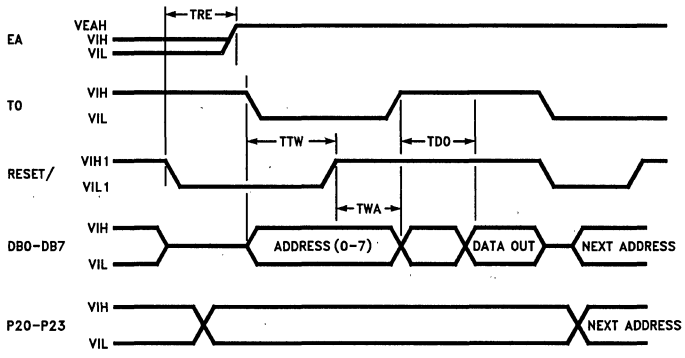
COMBINATION PROGRAM/VERIFY MODE



290414-25

NOTES:
1. A_0 must be held low (0V) during program/verify modes.
2. For V_{IH}, V_{IH1}, V_{IL}, V_{IL1}, V_{DDH}, and V_{DDL}, please consult the D.C. Characteristics Table.
3. When programming the 87C42, a 0.1 μF capacitor is required across V_{DD} and ground to suppress spurious voltage transients which can damage the device.

VERIFY MODE

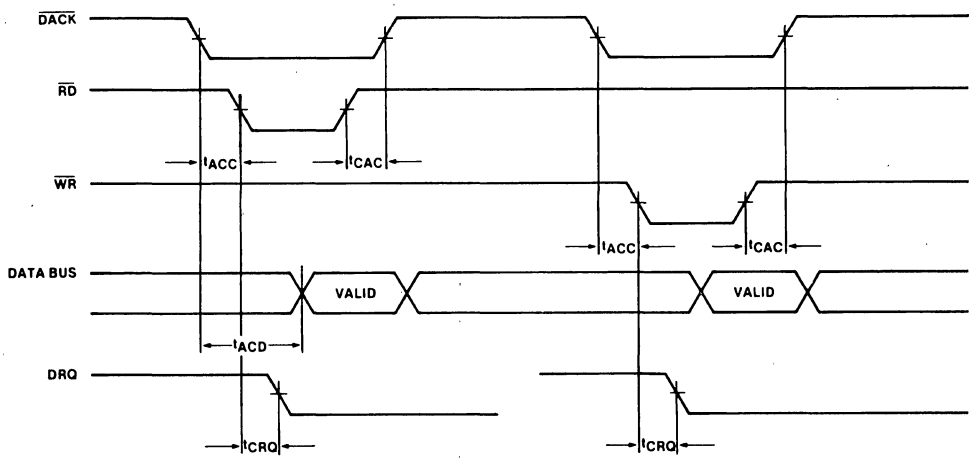


290414-26

NOTES:
1. PROG must float if EA is low.
2. PROG must float or = 5V when EA is high.
3. P_{10}-P_{17} = 5V or must float.
4. P_{24}-P_{27} = 5V or must float.
5. A_0 must be held low during programming/verify modes.

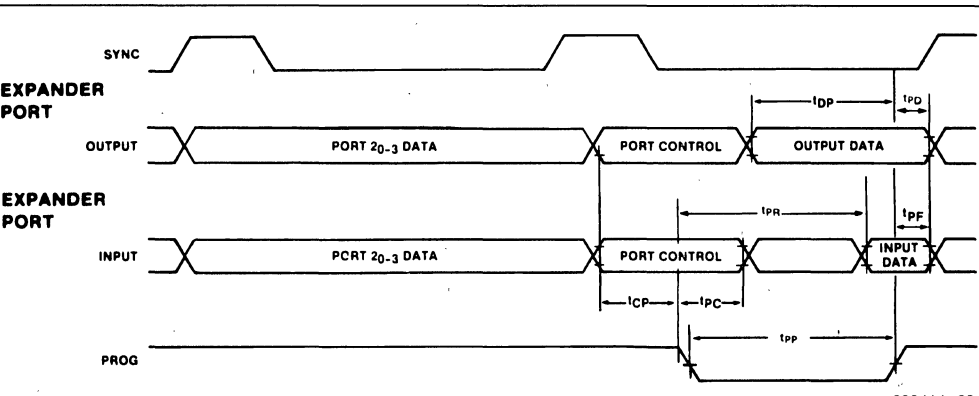
WAVEFORMS (Continued)

DMA



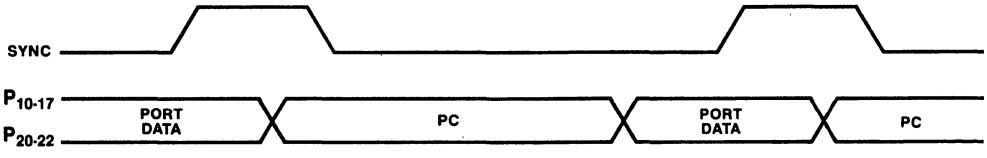
290414-27

PORT 2



290414-28

PORT TIMING DURING EXTERNAL ACCESS (EA)



290414-29

On the Rising Edge of SYNC and EA is Enabled, Port Data is Valid and can be Strobed. On the Trailing Edge of Sync the Program Counter Contents are Available.

Table 4. UPI Instruction Set

Mnemonic	Description	Bytes	Cycles
ACCUMULATOR			
ADD A, Rr	Add register to A	1	1
ADD A, @Rr	Add data memory to A	1	1
ADD A, #data	Add immediate to A	2	2
ADDC A, Rr	Add register to A with carry	1	1
ADDC A, @Rr	Add data memory to A with carry	1	1
ADDC A, #data	Add immediate to A with carry	2	2
ANL A, Rr	AND register to A	1	1
ANL, A @Rr	AND data memory to A	1	1
ANL A, #data	AND immediate to A	2	2
ORL A, Rr	OR register to A	1	1
ORL, A, @Rr	OR data memory to A	1	1
ORL A, #data	OR immediate to A	2	2
XRL A, Rr	Exclusive OR register to A	1	1
XRL A, @Rr	Exclusive OR data memory to A	1	1
XRL A, #data	Exclusive OR immediate to A	2	2
INC A	Increment A	1	1
DEC A	Decrement A	1	1
CLR A	Clear A	1	1
CPL A	Complement A	1	1
DA A	Decimal Adjust A	1	1
SWAP A	Swap nibbles of A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through carry	1	1
INPUT/OUTPUT			
IN A, Pp	Input port to A	1	2
OUTL Pp, A	Output A to port	1	2
ANL Pp, #data	AND immediate to port	2	2
ORL Pp, #data	OR immediate to port	2	2
IN A, DBB	Input DBB to A, clear IBF	1	1
OUT DBB, A	Output A to DBB, set OBF	1	1
MOV STS, A	A_4–A_7 to Bits 4–7 of Status	1	1
MOVD A, Pp	Input Expander port to A	1	2
MOVD Pp, A	Output A to Expander port	1	2
ANLD Pp, A	AND A to Expander port	1	2
ORLD Pp, A	OR A to Expander port	1	2

Mnemonic	Description	Bytes	Cycles
DATA MOVES			
MOV A, Rr	Move register to A	1	1
MOV A, @Rr	Move data memory to A	1	1
MOV A, #data	Move immediate to A	2	2
MOV Rr, A	Move A to register	1	1
MOV @Rr, A	Move A to data memory	1	1
MOV Rr, #data	Move immediate to register	2	2
MOV @Rr, #data	Move immediate to data memory	2	2
MOV A, PSW	Move PSW to A	1	1
MOV PSW, A	Move A to PSW	1	1
XCH A, Rr	Exchange A and register	1	1
XCH A, @Rr	Exchange A and data memory	1	1
XCHD A, @Rr	Exchange digit of A and register	1	1
MOVP A, @A	Move to A from current page	1	2
MOVP3, A, @A	Move to A from page 3	1	2
TIMER/COUNTER			
MOV A, T	Read Timer/Counter	1	1
MOV T, A	Load Timer/Counter	1	1
STRT T	Start Timer	1	1
STRT CNT	Start Counter	1	1
STOP TCNT	Stop Timer/Counter	1	1
EN TCNTI	Enable Timer/Counter Interrupt	1	1
DIS TCNTI	Disable Timer/Counter Interrupt	1	1
CONTROL			
*EN A20	Enable A20 Logic	1	1
EN DMA	Enable DMA Handshake Lines	1	1
EN I	Enable IBF Interrupt	1	1
DIS I	Diable IBF Interrupt	1	1
EN FLAGS	Enable Master Interrupts	1	1
*SEL PMB0	Select Program memory bank 0	1	1
*SEL PMB1	Select Program memory bank 1	1	1
SEL RB0	Select register bank 0	1	1
SEL RB1	Select register bank 1	1	1

* UPI-C42/UPI-L42 Only.

Table 4. UPI Instruction Set (Continued)

Mnemonic	Description	Bytes	Cycles
CONTROL (Continued)			
*SUSPEND	Invoke Suspend Power-down mode	1	2
NOP	No Operation	1	1
REGISTERS			
INC Rr	Increment register	1	1
INC @Rr	Increment data memory	1	1
DEC Rr	Decrement register	1	1
SUBROUTINE			
CALL addr	Jump to subroutine	2	2
RET	Return	1	2
RETR	Return and restore status	1	2
FLAGS			
CLR C	Clear Carry	1	1
CPL C	Complement Carry	1	1
CLR F0	Clear Flag 0	1	1
CPL F0	Complement Flag 0	1	1
CLR F1	Clear F1 Flag	1	1
CPL F1	Complement F1 Flag	1	1

Mnemonic	Description	Bytes	Cycles
BRANCH			
JMP addr	Jump unconditional	2	2
JMPP @A	Jump indirect	1	2
DJNZ Rr, addr	Decrement register and jump	2	2
JC addr	Jump on Carry = 1	2	2
JNC addr	Jump on Carry = 0	2	2
JZ addr	Jump on A Zero	2	2
JNZ addr	Jump on A not Zero	2	2
JT0 addr	Jump on T0 = 1	2	2
JNT0 addr	Jump on T0 = 0	2	2
JT1 addr	Jump on T1 = 1	2	2
JNT1 addr	Jump on T1 = 0	2	2
JF0 addr	Jump on F0 Flag = 1	2	2
JF1 addr	Jump on F1 Flag = 1	2	2
JTF addr	Jump on Timer Flag = 1, Clear Flag	2	2
JNIBF addr	Jump on IBF Flag = 0	2	2
JOBF addr	Jump on OBF Flag = 1	2	2
JBb addr	Jump on Accumula- for Bit	2	2

*UPI-C42/UPI-L42 Only.

REVISION SUMMARY

The following has been changed since Revision -003:

1. Delete all references to standby power down mode.

The following has been changed since Revision -002:

1. Added information on keyboard controller product family.
2. Added I_{HI} specification for the UPI-L42.

The following has been changed since Revision -001:

1. Added UPI-L42 references and specification.

5

UPI-452
CHMOS PROGRAMMABLE I/O PROCESSOR

83C452 - 8K × 8 Mask Programmable Internal ROM

80C452 - External ROM/EPROM

- **83C452/80C452:3.5 to 14 MHz Clock Rate**
- **Software Compatible with the MCS-51 Family**
- **128-Byte Bi-Directional FIFO Slave Interface**
- **Two DMA Channels**
- **256 × 8-Bit Internal RAM**
- **34 Additional Special Function Registers**
- **40 Programmable I/O Lines**
- **Two 16-Bit Timer/Counters**
- **Boolean Processor**
- **Bit Addressable RAM**
- **8 Interrupt Sources**
- **Programmable Full Duplex Serial Channel**
- **64K Program Memory Space**
- **64K Data Memory Space**
- **68-Pin PGA and PLCC**
  (See Packaging Spec., Order: #231369)

The Intel UPI-452 (Universal Peripheral Interface) is a 68 pin CHMOS Slave I/O Processor with a sophisticated bi-directional FIFO buffer interface on the slave bus and a two channel DMA processor on-chip. The UPI-452 is the newest member of Intel's UPI family of products. It is a general-purpose slave I/O Processor that allows the designer to grow a customized interface solution.

The UPI-452 contains a complete 80C51 with twice the on-chip data and program memory. The sophisticated slave FIFO module acts as a buffer between the UPI-452 internal CPU and the external host CPU. To both the external host and the internal CPU, the FIFO module looks like a bi-directional bottomless buffer that can both read and write data. The FIFO manages the transfer of data independent of the UPI-452 core CPU and generates an interrupt or DMA request to either CPU, host or internal, as a FIFO service request.

The FIFO consists of two channels:the Input FIFO and the Output FIFO. The division of the FIFO module array, 128 bytes, between Input channel and Output channel is programmable by the user. Each FIFO byte has an additional logical ninth bit to distinguish between a data byte and a Data Stream Command byte. Additionally, Immediate Commands allow direct, interrupt driven, bi-directional communication between the UPI-452 internal CPU and external host CPU, bypassing the FIFO.

The on-chip DMA processor allows high speed data transfers from one writeable memory space to another. As many as 64K bytes can be transferred in a single DMA operation. Three distinct memory spaces may be used in DMA operations; Internal Data Memory, External Data Memory, and the Special Function Registers (including the FIFO IN, FIFO OUT, and Serial Channel Special Functions Registers).

The complete document for this product is available from Intel's Literature Center at 1-800-548-4725.

November 1994
Order Number: 231428-006

82091AA
ADVANCED INTEGRATED PERIPHERAL (AIP)

- **Single-Chip PC Compatible I/O Solution for Notebook and Desktop Platforms:**
  - **82078 Floppy Disk Controller Core**
  - **Two 16550 Compatible UARTs**
  - **One Multi-Function Parallel Port**
  - **IDE Interface**
  - **Integrated Back Power Protection**
  - **Integrated Game Port Chip Select**
  - **5V or 3.3V Supply Operation with 5V Tolerant Drive Interface**
  - **Full Power Management Support**
  - **Supports Type F DMA Transfers for Faster I/O Performance**
  - **No Wait-State Host I/O Interface**
  - **Programmable Interrupt Interfaces**
  - **Single Crystal/Oscillator Clock (24 MHz)**
  - **Software Detectable Device ID**
  - **Comprehensive Powerup Configuration**
- **The 82091AA is 100 Percent Compatible with EISA, ISA and AT**
- **Host Interface Features**
  - **8-Bit Zero Wait-State ISA Bus Interface**
  - **DMA with Type F Transfers**
  - **Five Programmable ISA Interrupt Lines**
  - **Internal Address Decoder**
- **Parallel Port Features**
  - **All IEEE Standard 1284 Protocols Supported (Compatibility, Nibble, Byte, EPP, and ECP)**
  - **Peak Bi-Directional Transfer Rate of 2 MB/sec**
  - **Provides Interface for Low-Cost Engineless Laser Printer**
  - **16-Byte FIFO for ECP**
  - **Interface Backpower Protection**
- **Floppy Disk Controller Features**
  - **100 Percent Software Compatible with Industry Standard 82077SL and 82078**
  - **Integrated Analog Data Separator 250K, 300K, 500K, and 1 MBits/sec**
  - **Programmable Powerdown Command**
  - **Auto Powerdown and Wakeup Modes**
  - **Integrated Tape Drive Support**
  - **Perpendicular Recording Support for 4 MB Drives**
  - **Programmable Write Pre-Compensation Delays**
  - **256 Track Direct Address, Unlimited Track Support**
  - **16-Byte FIFO**
  - **Supports 2 or 4 Drives**
- **16550 Compatible UART Features**
  - **Two Independent Serial Ports**
  - **Software Compatible with 8250 and 16450 UARTs**
  - **16-Byte FIFO per Serial Port**
  - **Two UART Clock Sources, Supports MIDI Baud Rate**
- **IDE Interface Features**
  - **Generates Chip Selects for IDE Drives**
  - **Integrated Buffer Control Logic**
  - **Dual IDE Interface Support**
- **Power Management Features**
  - **Transparent to Operating Systems and Applications Programs**
  - **Independent Power Control for Each Integrated Device**
- **100-Pin QFP Package**
  (See Packaging Spec. 240800)

5

The 82091AA Advanced Integrated Peripheral (AIP) is an integrated I/O solution containing a floppy disk controller, 2 serial ports, a multi-function parallel port, an IDE interface, and a game port on a single chip. The integration of these I/O devices results in a minimization of form factor, cost and power consumption. The

floppy disk controller is the 82078 core. The serial ports are 16550 compatible. The parallel port supports all of the IEEE Standard 1284 protocols (ECP, EPP, Byte, Compatibility, and Nibble). The IDE interface supports 8- or 16-bit programmed I/O and 16-bit DMA. The Host Interface is an 8-bit ISA interface optimized for type "F" DMA and no wait-state I/O accesses. Improved throughput and performance, the 82091AA contains six 16-byte FIFOs–two for each serial port, one for the parallel port, and one for the floppy disk controller. The 82091AA also includes power management and 3.3V capability for power sensitive applications such as notebooks. The 82091AA supports both motherboard and add-in card configurations.

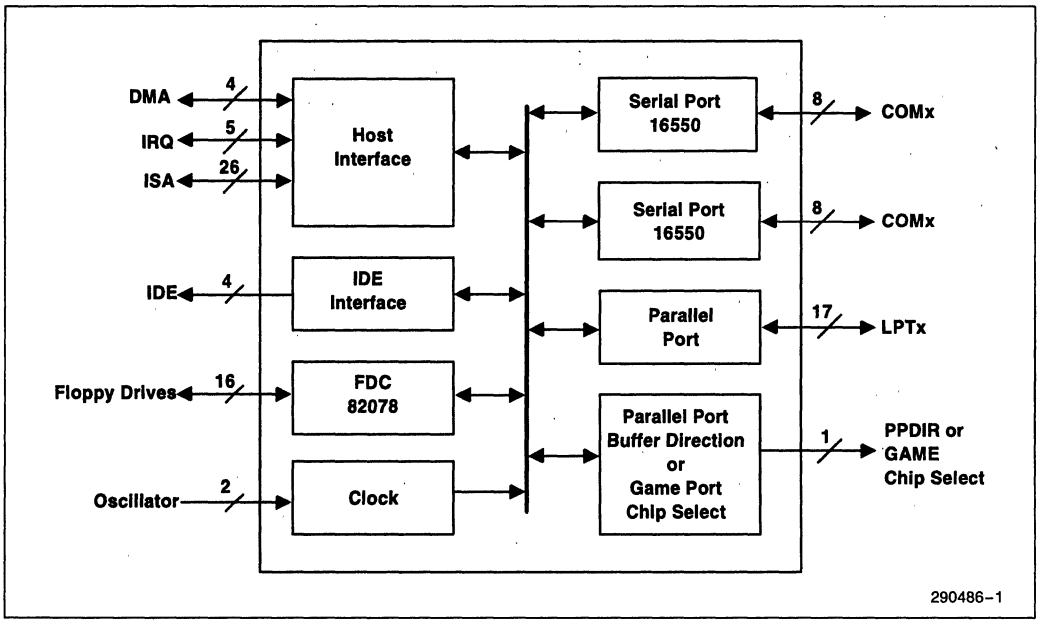


Figure 1. 82091AA Advanced Integrated Peripheral Block Diagram

82091AA
ADVANCED INTEGRATED PERIPHERAL (AIP)

CONTENTS

CONTENTS

PAGE

CONTENTS

CONTENTS

CONTENTS

1.0. OVERVIEW

The major functions of the 82091AA are shown in Figure 1. A brief description of each of these functions is presented in this section.

Host Interface

The 82091AA host interface is an 8-bit direct-drive (24 mA) ISA Bus/X-Bus interface that permits the CPU to access its registers through read/write operations in I/O space. These registers may be accessed by programmed I/O and/or DMA bus cycles. With the exception of the IDE Interface, all functions on the 82091AA require only 8-bit data accesses. The 16-bit access required for the IDE Interface is supported through the appropriate chip selects and data buffer enables from the 82091AA.

Figure 2 shows an example system implementation with the 82091AA located on an ISA Bus add-in card. This add-in card could also be used in a PCI-based system as shown in Figure 3. For motherboard implementations, the 82091AA can be located on the X-Bus as shown in Figure 4.

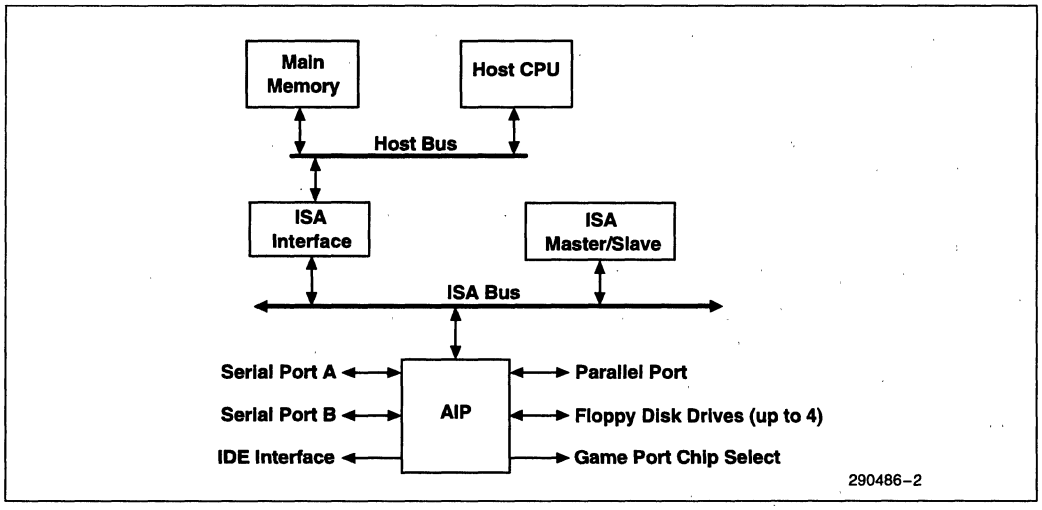


Figure 2. Block Diagram of the 82091AA on the ISA Bus

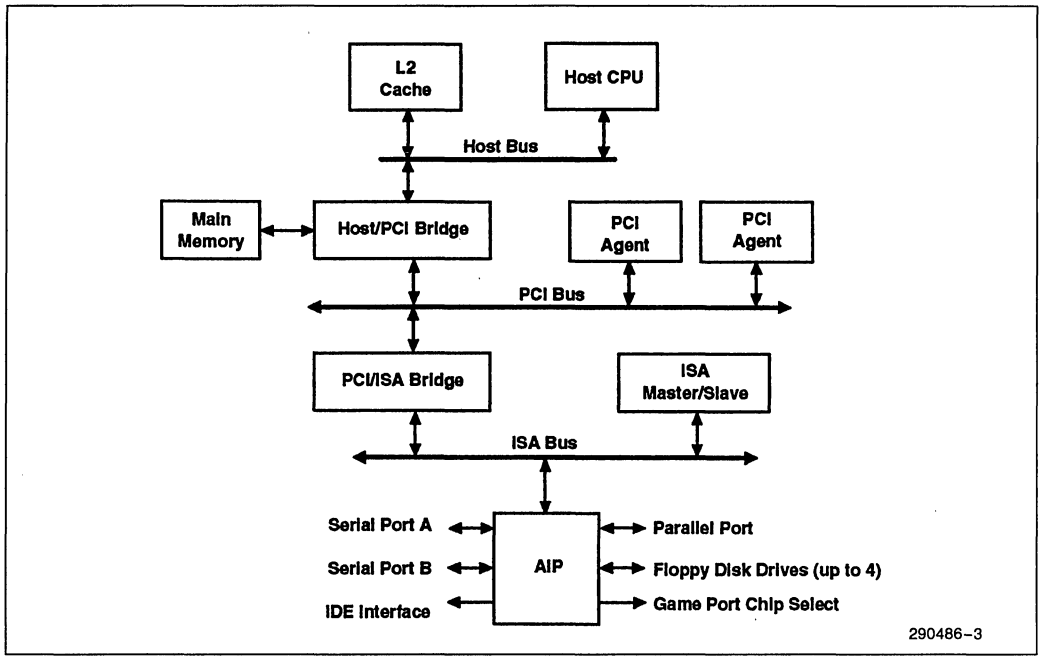


Figure 3. Block Diagram of the 82091AA in a PCI System

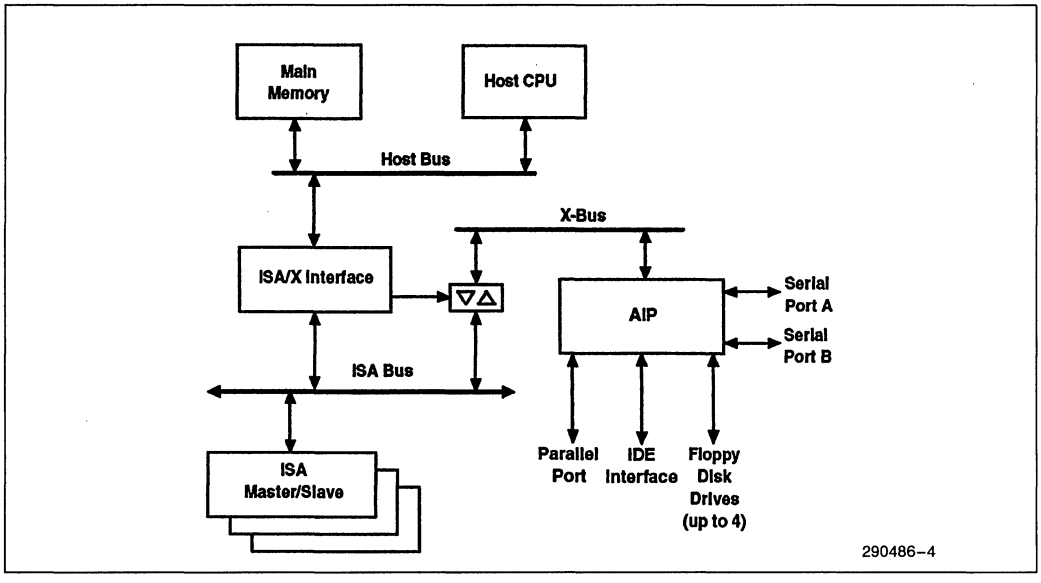


Figure 4. Block Diagram of the 82091AA on the X-Bus

5

Floppy Disk Controller

The 82091AA's enhanced floppy disk controller (FDC) incorporates several new features allowing for easy implementation in both the portable and desktop markets. It provides a low cost, small form factor solution targeted for 5.0V and 3.3V platforms. The FDC supports up to four drives.

The 82091AA's FDC implements these new features while remaining functionally compatible with 82078/82077SL/82077AA/8272A floppy disk controllers. Together, with a 24-MHz crystal, a resistor package and a device chip select, these devices allow for the most integrated solution available. The integrated analog PLL data separator has better performance than most board level discrete PLL implementations and can be operated at 1 Mbps/500 Kbps/300 Kbps/250 Kbps. A 16-byte FIFO substantially improves system performance and is ideal for multimaster systems (e.g., EISA).

Serial Ports

The 82091AA contains two independent serial ports that provide asynchronous communications that are equivalent to two 16550 UARTs. The serial ports have identical circuitry and provide the serial communication interface to a peripheral device or modem via Serial Port A and Serial Port B. Each serial port can be configured for one of eight address assignments. The standard PC/AT compatible logical address assignments for COM1, COM2, COM3, and COM4 are supported.

The serial ports perform serial-to-parallel conversion on data characters received from a peripheral device or modem, and parallel-to-serial conversion on data characters received from the host. The serial ports can operate in either FIFO mode or non-FIFO mode. In FIFO mode, a 16-byte transmit FIFO holds data from the host to be transmitted on the serial link and a 16-byte receive FIFO that buffers data from the serial link until read by the host.

The serial ports contain programmable baud rate generators that divide the internal reference clock by divisors of 1 to ($2^{16} - 1$), and produce a 16x clock for driving the transmitter and receiver logic. The internal reference clock can be programmed to support MIDI. The serial ports have complete modem-control capability and a prioritized interrupt system.

Parallel Port

The 82091AA provides a multi-function parallel port that transfers information between the host and peripheral device (e.g., printer). The parallel port interface contains nine control/status lines and an 8-bit data bus. The standard PC/AT compatible logical address assignments for LPT1, LPT2, and LPT3 are supported. The parallel port can be configured for one of four modes and supports the following IEEE Standard 1284 parallel interface protocol standards:

Parallel Port Mode	Parallel Interface Protocol
ISA-Compatible Mode	Compatibility, Nibble
PS/2-Compatible Mode	Byte
EPP Mode	EPP
ECP Mode	ECP

For ISA-Compatible and PS/2-Compatible modes, software controls the handshake signals on the parallel port interface to transfer data between the host and peripheral device. Status and Control registers permit software to monitor the state of the peripheral device and generate handshake sequences.

The EPP parallel port interface protocol increases throughput by specifying an automatic handshake sequence. In EPP mode, the 82091AA parallel port automatically generates this handshake sequence in hardware to transfer data between the host and peripheral device.

In addition to a hardware handshake on the parallel port interface, the ECP protocol specification also defines DMA and FIFO capability. To minimize processor overhead data transfer to/from a peripheral device, the 82091AA parallel port, in ECP mode, provides a 16-byte FIFO with DMA capability.

IDE Interface

The 82091AA supports the IDE (Integrated Drive Electronics) interface by providing chip selects and lower data byte control. Two chip selects are used to access registers on the IDE device. Separate lower and upper byte data control signals are provided. With these control signals, minimal external logic is needed to implement 16-bit IDE I/O and DMA interfaces.

Game Port

The 82091AA provides a game port chip select signal for use when the 82091AA is in an add-in card application. This function is assigned to I/O address location 201h. Note that when the 82091AA is located on the motherboard, this feature is not available.

Power Management

82091AA power management provides a mechanism for saving power when the device or a portion of the device is not being used. By programming the appropriate 82091AA registers, software can invoke power management to the entire 82091AA or selected modules within the 82091AA (e.g., floppy disk controller, serial port, or parallel port). There are two methods for applying power management—direct powerdown or auto powerdown. Direct powerdown turns off the clock to a particular module immediately placing that module into a powerdown state. This method removes the clock regardless of the activity or status of the module. When auto powerdown is invoked, the module enters a powerdown state (clock is turned off) after certain conditions are met and the module is in an idle state.

1.1. 3.3V/5V Operating Modes

The 82091AA can operate at a power supply of 3.3V, 5V or a mix of 3.3V and 5V. The mixed power supply mode provides 5V interfaces for the floppy disk controller and parallel port while all other 82091AA interfaces and internal logic (including the floppy disk controller and parallel port internal circuitry) operate at 3.3V. The mixed mode permits 5V floppy disk drives and parallel port peripherals to be used in a 3.3V system without external buffering.

NOTE:
3.3V operation is available only in the 82091AA.

2.0. SIGNAL DESCRIPTION

This section describes the 82091AA signals. The interface signals are shown in Figure 5 and described in the following tables. Signal descriptions are organized by functional group.

Note that the "#" symbol at the end of a signal name indicates the active, or asserted, state occurs when the signal is at a low voltage level. When "#" is not present after the signal name, the signal is asserted when at the high voltage level.

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

The following notations are used to describe pin types:

I Input Pin
O Output Pin
I/O Bi-Directional Pin

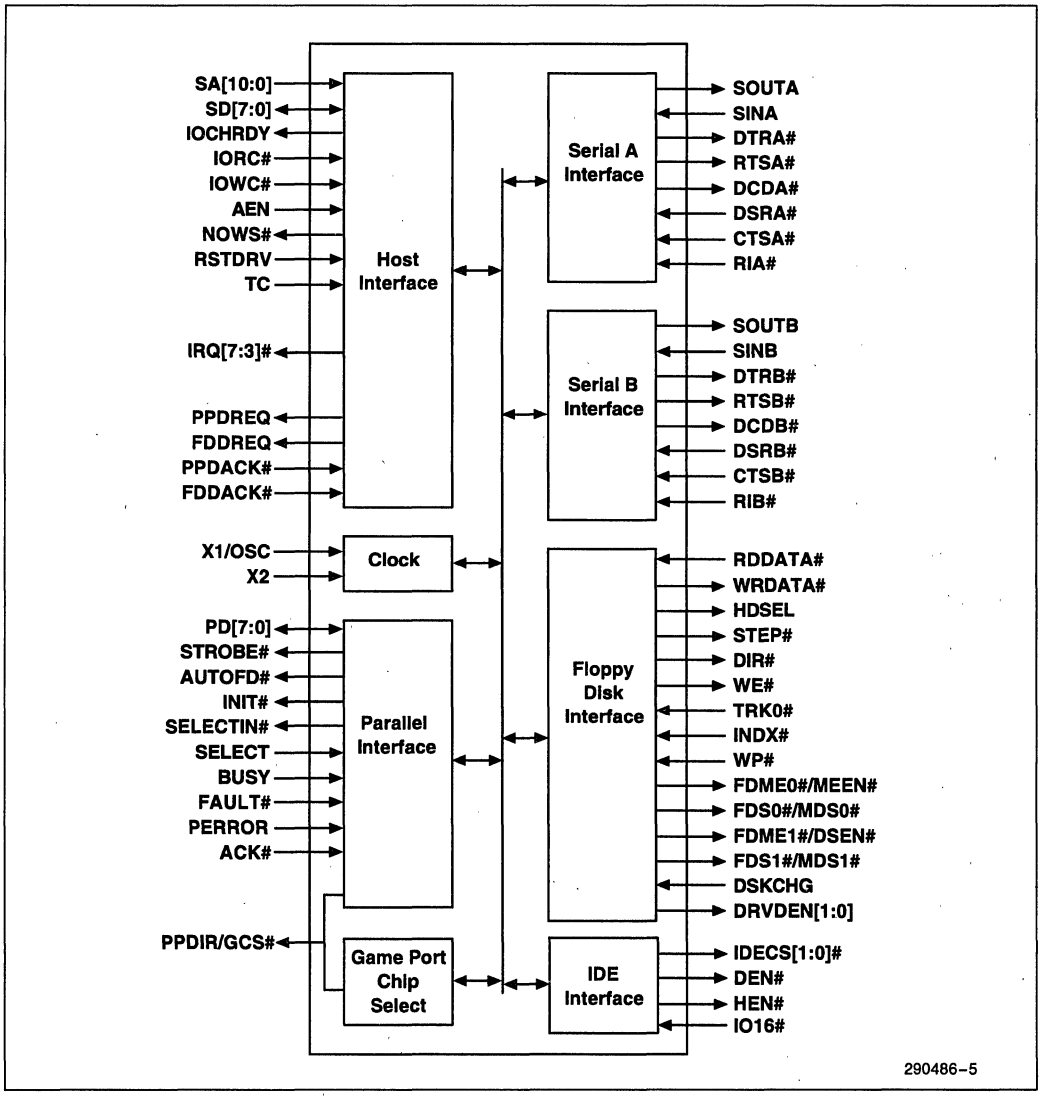


Figure 5. 82091AA Signals

2.1 Host Interface Signals

Signal Name	Type	Description
ISA SIGNALS		
SA[10:0]	I	**SYSTEM ADDRESS BUS:** The 82091AA decodes the standard ISA I/O address space using SA[9:0]. SA10 is used along with SA[9:0] to decode the extended register set of the ECP parallel port. SA[10:0] connects directly to the ISA system address bus.
SD[7:0]	I/O	**SYSTEM DATA BUS:** SD[7:0] is a bi-directional data bus. Data is written to and read from the 82091AA on these signal lines. SD[7:0] connect directly to the ISA system data bus.
IORC#	I	**I/O READ COMMAND STROBE:** IORC# is an I/O access read control signal. When a valid internal address is decoded by the 82091AA and IORC# is asserted, data at the decoded address location is driven onto the SD[7:0] signal lines.
IOWC#	I	**I/O WRITE COMMAND STROBE:** IOWC# is an I/O access write control signal. When a valid internal address is decoded by the 82091AA and IOWC# is asserted, data on the SD[7:0] signal lines is written into the decoded address location at the rising edge of IOWC#.
NOWS#	O	**NO WAIT-STATES:** End data transfer signal. The 82091AA asserts NOWS# when a valid internal address is decoded by the 82091AA and the IORC# or IOWC# signal is asserted. This reduces the total bus cycle time by eliminating the wait-states associated with the default 8-bit I/O cycles. NOWS# is not asserted for IDE accesses or DMA accesses. This is an open drain output pin.
IOCHRDY	O	**I/O CHANNEL READY:** The 82091AA uses this signal for parallel port data transfers when the parallel port is in EPP mode. In this case, the 82091AA negates IOCHRDY to extend the cycle to allow for completion of transfers to/from the peripheral attached to the parallel port. When the parallel port is in EPP mode, the 82091AA negates IOCHRDY to lengthen the ISA Bus cycle if the parallel port BUSY signal is asserted. The 82091AA also uses IOCHRDY during hardware configuration time (see Section 4.0, AIP Configuration). If IOWC#/IORC# is asserted to the 82091AA during hardware configuration time, the 82091AA negates IOCHRDY until hardware configuration time is completed. This is an open drain output pin.
AEN	I	**ADDRESS ENABLE:** AEN is used during DMA cycles to prevent the 82091AA from misinterpreting DMA cycles from valid I/O cycles. When negated, AEN indicates that the 82091AA may respond to address and I/O commands addressed to the 82091AA. When asserted, AEN informs the 82091AA that a DMA transfer is occurring. When AEN is asserted and a xDACK# signal is asserted, the 82091AA responds to the cycle as a DMA cycle.
RSTDRV	I	**RESET DRIVE:** RSTDRV forces the 82091AA to a known state. All 82091AA registers are set to their default state.
X1/OSC	I	**CRYSTAL1/OSCILLATOR:** Main clock input signal can be a 24 MHz crystal connected across X1 and X2 or a 24 MHz TTL level clock input connected to X1.
X2	I	**CRYSTAL2:** This signal pin is connected to one side of the crystal when a crystal oscillator is used to provide the main clock. If an external oscillator/clock is connected to X1, this pin is not used and left unconnected.

2.1 Host Interface Signals (Continued)

Signal Name	Type	Description
DMA SIGNALS		
FDDREQ	O	**FLOPPY DISK CONTROLLER DMA REQUEST:** The 82091AA asserts FDDREQ to request service from a DMA controller for the FDC module. This signal is enabled/disabled by bit 3 of the Digital Output Register (DOR). When disabled, FDDREQ is tri-stated.
FDDACK#	I	**FLOPPY DISK CONTROLLER DMA ACKNOWLEDGE:** The DMA controller asserts this signal to acknowledge the FDC DMA request. When asserted, the IORC# and IOWC# inputs are enabled during DMA transfers. This signal is enabled/disabled by bit 3 of the DOR.
PPDREQ	O	**PARALLEL PORT DMA REQUEST:** Parallel port DMA service request to the system DMA controller. This signal is only used when the parallel port is in ECP hardware mode and is always negated when the parallel port is not in this mode. In ECP hardware mode DMA requests are enabled/disabled by bit 3 of the ECP Extended Control Register (ECR). When disabled, PPDREQ is tri-stated.
PPDACK#	I	**PARALLEL PORT DMA ACKNOWLEDGE:** The DMA controller asserts this signal to acknowledge the parallel port DMA request. When asserted the IORC# and IOWC# inputs are enabled during DMA transfers. This signal is enabled/disabled by bit 3 of the ECR Register.
TC	I	**TERMINAL COUNT:** The system DMA controller asserts TC to indicate it has reached the last programmed data transfer. TC is accepted only when FDDACK# or PPDACK# is asserted.
INTERRUPT SIGNALS		
IRQ3, IRQ4	O	**INTERRUPT 3 AND 4:** IRQ3 and IRQ4 are associated with the serial ports and can be programmed (via the AIPCFG2 Register) to be either active high or active low. These signals can be configured for a particular serial channel via hardware configuration (at powerup) or by software configuration. **Under Hardware Configuration** IRQ3 is used as a serial port interrupt if the serial port is configured at address locations 2F8h–2FFh or 2E8h–2EFh. IRQ4 is used as a serial port interrupt if the serial port is configured at address locations 3F8h–3FFh or 3E8h–3EFh. **Under Software configuration** IRQ3 and IRQ4 are independently configured (i.e., the IRQ does not automatically track the communication port address assignment). These interrupts are enabled/disabled globally via bit 3 of the serial port Modem Control Register (MCR) and for specific conditions via the Interrupt Enable Register (IER). IRQ3 and IRQ4 are tri-stated when not enabled.
IRQ5, IRQ7	O	**INTERRUPT REQUEST 5:** IRQ5 and IRQ7 are associated with the parallel port and can be programmed (via AIPCFG2 Register) to be either active high or active low. Either IRQ5 or IRQ7 is enabled/disabled via PCFG1 Register to signal a parallel port interrupt. The interrupt not selected is disabled and tri-stated. During hardware configuration (see Section 4.0, AIP Configuration), IRQ5 is used if the parallel port is assigned to 278h–27Fh and IRQ7 is used if the parallel port interrupt is assigned to either 3BCh–3BFh or 378h–37Fh.

2.1 Host Interface Signals (Continued)

Signal Name	Type	Description
INTERRUPT SIGNALS (Continued)		
IRQ6	O	**INTERRUPT REQUEST 6:** IRQ6 is associated with the floppy disk controller and can be programmed (via the AIPCFG2 Register) to be either active high or active low. In non-DMA mode this signal is asserted to signal when a data transfer is ready. IRQ6 is also asserted to signal the completion of the execution phase for certain FDC commands. This signal is enabled/disabled by the DMAGATE bit in the Digital Output Register of the FDC. The signal is tri-stated when disabled.

2.2 Floppy Disk Controller Interface

Signal Name	Type	Description
RDDATA#	I	**READ DATA:** Serial data from the disk drive.
WRDATA#	O	**WRITE DATA:** MFM serial data to the disk drive. Precompensation value is selectable through software.
HDSEL	O	**HEAD SELECT:** Selects which side of a disk is to be accessed. When asserted (low), side 1 is selected. When negated (high), side 0 is selected.
STEP#	O	**STEP:** STEP# supplies step pulses (asserted) to the drive to move the head between the tracks during a seek operation.
DIR#	O	**DIRECTION:** Controls the direction the head moves when a step signal is present. The head moves toward the center when DIR# is asserted and away from the center when negated.
WE#	O	**WRITE ENABLE:** WE# is a disk drive control signal. When asserted, WE# enables the head to write to the disk.
TRK0#	I	**TRACK0:** The disk drive asserts this signal to indicate that the head is on track 0.
INDX#	I	**INDEX:** The disk drive asserts this signal to indicate the beginning of the track.
WP#	I	**WRITE PROTECT:** The disk drive asserts this signal to indicate that the disk drive is write-protected.
DSKCHG	I	**DISK CHANGE:** The disk drive asserts this signal to indicate that the drive door has been opened. The state of this signal input is available in the Digital Input Register (DIR#).
DRIVDEN0 DRIVDEN1	O	**DRIVE DENSITY:** These signals are used by the disk drive to configure the drive for the appropriate media density. These signals are controlled by the FDC's Drive Specification Command.

2.2 Floppy Disk Controller Interface (Continued)

Signal Name	Type	Description
FDME1#/ DSEN#(1)	O	**FLOPPY DRIVE MOTOR ENABLE 1, IDLE, OR DRIVE SELECT ENABLE:** This signal pin has two functions(1). FDME1# is the motor enable for drive 1. FDME1# is directly controlled via the Digital Output Register (DOR) and is a function of the mapping based on the BOOTSEL bits in the Tape Drive Register (TDR). The Drive Select Enable (DSEN#) function is only used in a four floppy drive system (see Appendix A, FDC Four Drive Support).
FDS1#/ MDS1(1)	O	**FLOPPY DRIVE SELECT1, POWERDOWN, OR MOTOR DRIVE SELECT 1:** This signal pin has two functions(1). FDS1# is the floppy drive select for drive 1. FDS1# is controlled by the select bits in the DOR and is a function of the mapping based on the BOOTSEL bits in the TDR. The Motor Drive Select 1 (MDS1) function is only used in a four floppy drive system (see Appendix A, FDC Four Drive Support).
FDME0#/ MEEN#(1)	O	**FLOPPY DRIVE MOTOR ENABLE 0 OR MOTOR ENABLE ENABLE:** This signal pin has two functions(1). FDME0# is the motor enable for drive 0. FDME0# is directly controlled via the Digital Output Register (DOR) and is a function of the mapping based on the BOOTSEL bits in the Tape Drive Register (TDR). The Motor Enable Enable (MEEN#) function is only used in a four floppy drive system (see Appendix A, FDC Four Drive Support).
FDS0#/ MDS0(1)	O	**FLOPPY DRIVE SELECT 0 OR MOTOR DRIVE SELECT 0:** This signal pin has two functions(1). FDS0# is the floppy drive select for drive 0. This output is controlled by the drive select bits in the DOR and is a function of the mapping based on BOOTSEL bits in the TDR. The Motor Drive Select 0 (MDS0) function is only used in a four floppy drive system (see Appendix A, FDC Four Drive Support).

NOTE:
1. The function selected for these pins is based on the FDDQTY bit in the FCFG1 Register as shown in the following table.

Signal Pin	2 Drive System (FDDQTY = 0)	4 Drive System (FDDQTY = 1)
FDME1#/DSEN#	FDME1#	DSEN#
FDS1#/MDS1#	FDS1#	MDS1
FDME0#/MEEN#	FDME0#	MEEN#
FDS0#/MDS0	FDS0#	MDS0

When FDDQTY = 1, these signal pins are used to control an external decoder for a four floppy disk drive system as described in Appendix A, FDC Four Drive Support.

2.3 Serial Port Interface

Serial Port A signal names end in the letter A and Serial Port B signal names end in the letter B. Serial Port A and B signals have the same functionality.

Signal Name	Type	Description
CTSA#, CTSB#	I	**CLEAR TO SEND:** When asserted, this signal indicates that the modem or data set is ready to exchange data. The CTS# signal is a modem status input whose condition the CPU can determine by reading the CTS bit in Modem Status Register (MSR) for the appropriate serial port. The CTS bit is the compliment of the CTS# signal. The DCTS bit in the MSR indicates whether the CTS# input has changed state since the previous reading of the MSR. CTS# has no effect on the transmitter.
DCDA#, DCDB#	I	**DATA CARRIER DETECT:** When asserted, this signal indicates that the data carrier has been detected by the modem or data set. The DCD# signal is a modem status whose condition the CPU can determine by reading the DCD bit in the MSR for the appropriate serial port. The DCD bit is the compliment of the DCD# signal. The DDCD bit in the MSR indicates whether the DCD# input has changed state since the previous reading of the MSR. DCD# has no effect on the transmitter.
DSRA#, DSRB#	I	**DATA SET READY:** When asserted, this signal indicates that the modem or data set is ready to establish the communications link with the serial port module. The DSR# signal is a modem status whose condition the CPU can determine by reading the DSR bit in the MSR for the appropriate serial channel. The DSR bit is the compliment of the DSR# signal. The DSR bit in the MSR indicates whether the DSR# input has changed state since the previous reading of the MSR. DSR# has no effect on the transmitter.
DTRA#, DTRB#	I/O	**DATA TERMINAL READY:** DTRA#/DTRB# are outputs during normal system operations. When asserted, this signal indicates to the modem or data set that the serial port module is ready to establish a communications link. The DTR# signal can be asserted via the Modem Control Register (MCR). A hard reset negates this signal. **Hardware Configuration** These signals are only inputs during hardware configuration time (RSTDRV asserted and for a short time after RSTDRV is negated). (See Section 4.0, AIP Configuration.)
RIA#, RIB#	I	**RING INDICATOR:** When asserted, this signal indicates that a telephone ringing signal has been received by the modem or data set. The RI# signal is a modem status input whose condition the CPU can determine by reading the RI bit in the MSR for the appropriate serial channel. The RI bit is the compliment of the RI# signal. The TERI bit in the MSR indicates whether the RI# input has changed from low to high since the previous reading of the MSR.

2.3 Serial Port Interface (Continued)

Signal Name	Type	Description
RTSA#, RTSB#	I/O	**REQUEST TO SEND:** RTSA#/RTSB# are outputs during normal system operations. When asserted, this signal informs the modem or data set that the serial port module is ready to exchange data. The RTS# signal can be asserted via the RTS bit in the Modem Control Register. A hard reset negates this signal. **Hardware Configuration** These signals are only inputs during hardware configuration time (RSTDRV asserted and for a short time after RSTDRV is negated). (See Section 4.0, AIP Configuration.)
SINA, SINB	I	**SERIAL INPUT:** Serial data input from the communications link. (Peripheral device, modem, or data set.)
SOUTA, SOUTB	I/O	**SERIAL OUTPUT:** SOUTA/SOUTB are serial data outputs to the communications link during normal system operations. (Peripheral device, modem, or data set.) The SOUT signal is set to a marking state (logic 1) after a hard reset. **Test Mode** In test mode (selected via the SACFG2 or SBCFG2 Registers), the baudout from the baud rate generator is output on SOUTx. **Hardware Configuration** These signals are only inputs during hardware configuration time (RSTDRV asserted and for a short time after RSTDRV is negated). (See Section 4.0, AIP Configuration.)

2.4 IDE Interface

Signal Name	Type	Description
IO16#	I	**16-BIT I/O:** This signal is driven by I/O devices on the ISA Bus to indicate support for 16-bit I/O bus cycles. The IDE interface asserts this signal to the 82091AA to indicate support for 16-bit transfers. For IDE transfers, the 82091AA asserts HEN# when IO16# is asserted.
IDECS[1:0]#	I/O	**IDE CHIP SELECT:** IDECS[1:0]# are outputs during normal system operation and are chip selects for the IDE interface. IDECS[1:0]# select the Command Block Registers of the IDE device and are decoded from SA[9:3] and AEN. **Hardware Configuration** These signals are only inputs during hardware configuration time (RSTDRV asserted). (See Section 4.0, AIP Configuration.)

2.4 IDE Interface (Continued)

Signal Name	Type	Description
DEN#	I/O	**DATA ENABLE:** DEN# is an output during normal system operations and is a data enable for an external data buffer for all 82091AA and IDE accesses. The SD[7:0] signals can be connected directly to the ISA. In this case, the DEN# signal is not used. However, an external buffer can be used to isolate the SD[7:0] signals from the 240 pF loading of the ISA Bus. With an external buffer implementation, DEN# controls the external buffers for transfers to/from the ISA Bus. **Hardware Configuration** This signal is only an input during hardware configuration time (RSTDRV asserted). (See Section 4.0, AIP Configuration.)
HEN#	I/O	**IDE UPPER DATA TRANSCEIVER ENABLE:** HEN# is an output during normal system operations and is a high byte data transceiver enable signal for the IDE hard disk drive interface. HEN# is asserted for I/O accesses to the IDE data register when the drive asserts IO16#. **Hardware Configuration** This signal is only an input during hardware configuration time (RSTDRV asserted). (See Section 4.0, AIP Configuration.)

2.5 Parallel Port External Buffer Control/Game Port

Signal Name	Type	Description
PPDIR/GCS#	I/O	**PARALLEL PORT DIRECTION (PPDIR) or GAME PORT CHIP SELECT (GCS#):** This signal is an output during normal operations and provides the PPDIR and GCS# functions as follows: **PPDIR** This signal pin functions as a parallel port direction control output when the 82091AA is configured for software motherboard mode (SWMB). For configuration details, see Section 4.0, AIP Configuration. If external buffers are used on PD[7:0], PPDIR can be used to control the buffer direction. The 82091AA drives this signal low when PD[7:0] are outputs and the 82091AA drives this signal high when PD[7:0] are inputs. Note that if a configuration mode other than SWMB is selected, this signal pin is a game port chip select and does not track the PD[7:0] signal direction. **GCS#** This signal pin functions as a game port chip select output when 82091AA configuration is set for Software Add-In (SWAI), Hardware Basic (HWB), or Hardware Extended (HWE) modes. When the host accesses I/O address 201h, GCS# is asserted. **Hardware Configuration** This signal is only an input during hardware configuration time (RSTDRV asserted). (See Section 4.0, AIP Configuration.)

2.6 Parallel Port Interface

The 82091AA parallel port is a multi-function interface that can be configured for one of four hardware modes (see Section 4.0, AIP Configuration). The hardware modes are ISA-Compatible, PS/2-Compatible, EPP, and ECP modes. These parallel port modes support the compatibility, nibble, byte, EPP and ECP parallel interface protocols described in the IEEE 1284 standard. The operation and use of the interface signal pins are a function of the parallel port hardware mode selected and the protocol used.

Table 1 shows a matrix of the 82091AA parallel port signal names and corresponding signal names for each of the protocols. Sections 2.6.1–2.6.5 provide a signal description for the five interface protocols. Note that the 82091AA hardware operations are the same for Compatibility and Nibble protocols. The signals, however, are controlled and used differently via software and the peripheral device.

Table 1. Parallel Port Signal Name Cross Reference

82091AA Signal Names	Compatibility Protocol Signal Names	Nibble Protocol Signal Names	Byte Protocol Signal Names	EPP Protocol Signal Names	ECP Protocol Signal Names
STROBE#	Strobe#	—	HostCLK	Write#	HostClk
BUSY	Busy	PtrBusy	PtrBusy	Wait#	PeriphAck
ACK#	Ack#	PtrClk	PtrClk	Intr	PeriphClk#
SELECT	Select	Xflag	Xflag	Xflag	Xflag
PERROR	PError	AckDataReq	AckDataReq	AckDataReq	AckReverse#
FAULT#	Fault#	DataAvail#	DataAvail#	DataAvail#	PeriphRequest#
INIT#	Init#	—	—	Init#	ReverseRequest#
AUTOFD#	AutoFd#	HostBusy	HostBusy	DStrb#	HostAck
PD[7:0]	Data[8:1]	—	Data[8:1]	Data[8:1]	Data[8:1]
SELECTIN#	SelectIn#	—	—	AStrb#	ECP Mode

NOTE:
Not all parallel port signal pins are used for certain parallel port interface protocols. These signals are labeled "—".

2.6.1 COMPATIBILITY PROTOCOL SIGNAL DESCRIPTION

Except for the data bus, the 82091AA and compatibility protocol signal names are the same. For the data bus, the 82091AA signal names PD[7:0] corresponds to the compatibility protocol signal names Data[8:1].

82091AA Signal Name	Type	Compatibility Protocol Signal Name and Description
STROBE#	O	**STROBE:** The host asserts STROBE# to latch data into the peripheral device's input latch. This signal is controlled via the PCON Register.
BUSY	I	**BUSY:** BUSY is asserted by the peripheral to indicate that the peripheral device is not ready to receive data. The status of this signal line is reported in the PSTAT Register.
ACK#	I	**ACKNOWLEDGE:** The printer asserts this signal to indicate that it has received the data and is ready for new data. The status of this signal line is reported in the PSTAT Register.
SELECT	I	**SELECT:** SELECT is asserted by the peripheral device to indicate that the device is on line. The status of this signal line is reported in the PSTAT Register.
PERROR	I	**PAPER ERROR:** The peripheral device asserts PERROR to indicate that it has encountered an error in the paper path. The exact meaning varies from peripheral device to peripheral device. The status of this signal line is reported in the PSTAT Register.
FAULT#	I	**FAULT:** FAULT# is asserted by the peripheral device to indicate that an error has occurred. The status of this signal line is reported in the PSTAT Register.
INIT#	O	**INITIALIZE:** The host asserts INIT# to issue a hardware reset to the peripheral device. This signal is controlled via the PCON Register.
AUTOFD#	O	**AUTO FEED:** AUTOFD# is asserted by the host to put the peripheral device into auto-line feed mode. This means that when software asserts this signal, the printer is instructed to advance the paper one line for each carriage return encountered. This signal is controlled via the PCON Register.
PD[7:0]	O	**DATA:** Forward channel data.
SELECTIN#	O	**SELECT INPUT:** SELECTIN# is asserted by the host to select a peripheral device. This signal is controlled via the PCON Register.

2.6.2 NIBBLE PROTOCOL SIGNAL DESCRIPTION

The Nibble protocol assigns the following signal operation to the parallel port pins. The name in bold at the beginning of the signal description column is the Nibble protocol signal name. The terms **assert** and **negate** are used in accordance with the 82091AA signal name as described at the beginning of Section 2.0. For example, AUTOFD# (HostBusy) asserted refers to AUTOFD# (HostBusy) at a low level.

82091AA Signal Name	Type	Nibble Protocol Signal Name and Description
STROBE#	O	**STROBE:** The host controls this signal via the PCON Register and STROBE# should be held negated by the host.
BUSY	I	**PRINTER BUSY (PtrBusy):** The peripheral drives this signal to transfer data bits 3 and 7 sequentially. The status of this signal line is reported in the PSTAT Register.
ACK#	I	**PRINTER CLOCK (PtrClk):** The peripheral device asserts ACK# (PtrClk) to indicate to the host that data is available. The signal is subsequently asserted to qualify data being sent to the host. The status of this signal line is reported in the PSTAT Register. If interrupts are enabled via the PCON Register, the assertion of this signal causes a host interrupt to be generated.
SELECT	I	**XFLAG:** The peripheral device drives this signal to transfer data bits 1 and 5 sequentially. The status of this signal line is reported in the PSTAT Register.
PERROR	I	**ACKNOWLEDGE DATA REQUEST (AckDataReq):** This signal is initially high. The peripheral device drives this signal low to acknowledge HostBusy assertion. PERROR is subsequently used to transfer data bits 2 and 6 sequentially. The status of this signal line is reported in the PSTAT Register.
FAULT#	I	**DATA AVAILABLE (DataAvail):** The peripheral device asserts FAULT# (DataAvail) to indicate data availability. Subsequently used to transfer data bits 0 and 4 sequentially. The status of this signal line is reported in the PSTAT Register.
INIT#	O	**INITIALIZE:** The host controls this signal via the PCON Register.
AUTOFD#	O	**HOST BUSY (HostBusy):** The host negates AUTOFD# (HostBusy) in response to ACK# being asserted. This signal is subsequently driven low to enable the peripheral to transfer data to the host. AUTOFD# is then driven high to acknowledge receipt of byte data. This signal is controlled via the PCON Register.
PD[7:0]	O	**DATA:** This 8-bit output data path to the peripheral Host data is written to the peripheral attached to the parallel port interface on these signal lines.
SELECTIN#	O	**SELECT INPUT:** This signal is controlled by the PCON Register.

2.6.3 BYTE MODE SIGNAL DESCRIPTION

The Byte protocol assigns the following signal operation to the parallel port pins. The name in bold at the beginning of the signal description column is the Byte protocol signal name. The terms **assert** and **negate** are used in accordance with the 82091AA signal name as described at the beginning of Section 2.0. For example, STROBE# (HostClk) asserted refers to STROBE# (HostClk) at a low level.

82091AA Signal Name	Type	Byte Protocol Signal Name and Description
STROBE#	O	**HOST CLOCK (HostClk):** This signal is strobed low by the host to acknowledge receipt of data. Note that the peripheral must not interpret this as a latch strobe for forward channel data.
BUSY	I	**PRINTER BUSY (PtrBusy):** The peripheral device asserts BUSY (PtrBusy) to provide forward channel peripheral busy status. The status of this signal line is reported in the PSTAT Register.
ACK#	I	**PRINTER CLOCK (PtrClk):** The peripheral device asserts ACK# (PtrClk) to indicate to the host that data is available. The signal is subsequently asserted to qualify data being sent to the host. The status of this signal line is reported in the PSTAT Register. If interrupts are enabled via the PCON Register, the assertion of this signal causes a host interrupt to be generated.
SELECT	I	**XFLAG:** SELECT (XFLAG) is asserted by the peripheral device to indicate that the device is on line. The status of this signal line is reported in the PSTAT Register.
PERROR	I	**ACKNOWLEDGE DATA REQUEST (AckDataReq):** This signal is initially high. The peripheral device drives this signal low to acknowledge HostBusy assertion. The status of this signal line is reported in the PSTAT Register.
FAULT#	I	**DATA AVAILABILITY (DataAvail):** The peripheral device asserts FAULT# (DataAvail) to indicate data availability. The status of this signal line is reported in the PSTAT Register.
INIT#	O	**INITIALIZE:** The host controls this signal via the PCON Register and INIT# should be held in the negated state.
AUTOFD#	O	**HOST BUSY (HostBusy):** The host negates AUTOFD# (HostBusy) in response to ACK# being asserted.The signal is subsequently driven low to enable the peripheral to transfer data to the host. AUTOFD# is then driven high to acknowledge receipt of nibble data. This signal is controlled via the PCON Register.
PD[7:0]	O	**DATA:** This 8-bit data bus is used for bi-directional data transfer.
SELECTIN#	I/O	**SELECT INPUT:** This signal is controlled by the PCON Register.

2.6.4 ENHANCED PARALLEL PORT (EPP) PROTOCOL SIGNAL DESCRIPTION

EPP protocol assigns the following signal operation to the parallel port pins. The name in bold at the beginning of the signal description column is the EPP mode signal name. The terms **assert** and **negate** are used in accordance with the 82091AA signal name as described at the beginning of Section 2.0. For example, BUSY (Wait#) asserted refers to BUSY (Wait#) being high.

82091AA Signal Name	Type	EPP Protocol Signal Name and Description
STROBE#	O	**WRITE (Write#):** STROBE# (Write#) indicates an address or data read/write operation to the peripheral. The 82091AA drives this signal low for a write and high for a read.
BUSY	I	**WAIT (Wait#):** The peripheral sets BUSY (Wait#) low to indicate that the device is not ready. When BUSY signal is low, the 82091AA negates IOCHRDY on the ISA Bus to lengthen the I/O cycles. The peripheral device sets BUSY (Wait#) high to indicate that transfer of data or address is completed.
ACK#	I	**INTERRUPT REQUEST (Intr):** The peripheral asserts ACK# (Intr) to generate an interrupt the host. When this signal is low and interrupts are enabled via bit 4 of the PCON Register, the 82091AA generates an interrupt request (via either IRQ5 or IRQ7) to the host.
SELECT	I	**SELECT:** SELECT is asserted by the peripheral device to indicate that the device is on line. The status of this signal line is reported in the PSTAT Register.
PERROR	I	**PAPER ERROR:** The peripheral device asserts PERROR to indicate that it has encountered an error in the paper path. The exact meaning varies from peripheral device to peripheral device. The status of this signal line is reported in the PSTAT Register.
FAULT#	I	**FAULT:** FAULT# is asserted by the peripheral device to indicate that an error has occurred. The status of this signal line is reported in the PSTAT Register.
INIT#	O	**INITIALIZE:** The host asserts INIT# to issue a hardware reset to the peripheral device. This signal is controlled via the PCON Register.
AUTOFD#	O	**DATA STROBE (DStrb#):** The 82091AA asserts AUTOFD# (DStrb#) to indicate that valid data is present on PD[7:0] and is used by the peripheral to latch data during write cycles. For reads, the 82091AA reads in data from PD[7:0] when this signal is asserted.
PD[7:0]	I/O	**DATA:** This 8-bit bi-directional bus provides addresses or data during the write cycles and supplies addresses or data to the 82091AA during the read cycles.
SELECTIN#	O	**ADDRESS STROBE (AStrb#):** The 82091AA asserts SELECTIN# (AStrb#) to indicate that a valid address is present on PD[7:0] and is used by the peripheral to latch addresses during write cycles. For reads, the 82091AA reads in an address from PD[7:0] when this signal is asserted.

2.6.5 EXTENDED CAPABILITIES PORT (ECP) PROTOCOL SIGNAL DESCRIPTION

ECP protocol assigns the following signal operation to the parallel port pins. The name in bold at the beginning of the signal description column is the ECP protocol signal name. The terms **assert** and **negate** are used in accordance with the 82091AA signal name as described at the beginning of Section 2.0. For example, STROBE# (HostClk) asserted refers to STROBE# (HostClk) being low.

82091AA Signal Name	Type	ECP Protocol Signal Name and Description
STROBE#	O	**HOST CLOCK (HostClk):** In the forward direction, the 82091AA asserts STROBE# (HostClk) to instruct the peripheral to latch the data on PD[7:0]. During write operations, the peripheral should latch data on the rising edge of STROBE# (HostClk). STROBE# (HostClk) handshakes with BUSY (PeriphAck) during write operations and is negated after the 82091AA detects BUSY (PeriphAck) asserted. STROBE# (HostClk) is not asserted by the 82091AA again until BUSY (PeriphAck) is detected negated. For read operations (reverse direction), STROBE# (HostClk) is not used.
BUSY	I	**PERIPHERAL ACKNOWLEDGE (PeriphAck):** The peripheral device asserts this signal during a host write operation to acknowledge receipt of data. The peripheral device then negates the signal after STROBE# is detected high to terminate the transfer. For host write operations (forward direction), this signal handshakes with STROBE# (HostClk). During a host read operation (reverse direction), BUSY (PeriphAck) is normally low and is driven high by the peripheral to identify Run Length Encoded (RLE) data.
ACK#	I	**PERIPHERAL CLOCK (PeriphClk):** During a peripheral to host transfer (reverse direction), ACK# (PeriphClk) is asserted by the peripheral to indicate data is valid on the data bus and then negated after AUTOFD# is detected high. This signal handshakes with AUTOFD# to transfer data.
SELECT	I	**XFLAG (Xflag):** This signal is asserted by the peripheral to indicate that it is on-line. The status of this signal line is reported in the PSTAT Register.
PERROR	I	**ACKNOWLEDGE REVERSE (AckReverse#):** PERROR (AckReverse#) is driven low by the peripheral to acknowledge a reverse transfer request by the host. This signal handshakes with INIT# (ReverseRequest#). The status of this signal line is reported in the PSTAT Register.
FAULT#	I	**PERIPHERAL REQUEST (PeriphRequest#):** The peripheral asserts FAULT# (PeriphRequest#) to request a reverse transfer. The status of this signal line is reported in the PSTAT Register.
INIT#	O	**REVERSE REQUEST (ReverseRequest#):** The host controls this signal via the PCON Register to indicate the transfer direction. The host asserts this signal to request a reverse transfer direction and negates the signal for a forward transfer direction.
AUTOFD#	O	**HOST ACKNOWLEDGE (HostAck):** The 82091AA asserts AUTOFD# (HostAck) to request data from the peripheral (reverse direction). This signal handshakes with ACK# (PeriphClk). AUTOFD# (HostAck) is negated when the peripheral indicates valid state of the data bus (i.e., ACK# is detected asserted). In the forward direction, AUTOFD# (HostAck) indicates whether PD[7:0] contain an address/RLE or data. The 82091AA asserts this signal to identify an address/RLE transfer and negates it to identify a data transfer.
PD[7:0]	I/O	**DATA:** PD[7:0] is a bi-directional data bus that transfers data, addresses, or RLE data.
SELECTIN#	O	**ECP MODE (ECPmode):** The host (via the PCON Register) negates this signal during ECP mode operation.

2.7 Hard Reset Signal Conditions

Table 1 shows the state of all 82091AA output and bi-directional signals during hard reset (RSTDRV asserted). The strapping options described in Section 4.0, AIP Configuration are sampled when the 82091AA is hard reset.

Table 2. Output and I/O Signal States During a Hard Reset

Signal Name	State
ACK#	—
AEN	—
AUTOFD#	Tri-state
BUSY	—
CTS[A,B]#	—
DCD[A,B]#	High
DEN#	High(1)
DIR#	High
DRVDEN[1:0]0	Low
DSKCHG#	—
DTR[A,B]#	High(1)
FAULT#	—
FDDACK#	—
FDDREQ	Tri-state
FDME0#/MEEN#	High
FDME1#/DSEN#	High
FDS0#/MDS0	High
FDS1#/MDS1	High

Signal Name	State
HDSEL	High
HEN#	High(1)
IDECS[1,0]#	High(1)
INDX#	—
INIT#	Low
IO16#	—
IOCHRDY	Tri-state(2)
IORC#	—
IOWC#	—
IRQ[7:3]	Tri-state
NOWS#	Tri-state
PD[7:0]	Low
PERROR	—
PPDACK#	—
PPDREQ	Tri-state
PPDIR/GCS#	High(1)
RDDATA	—
RI[A,B]#	—

Signal Name	State
RSTDRV	—
RTS[A,B]#	High(1)
SA[10,0]	—
SD[7:0]	Tri-state
SELECT	—
SELECTIN#	Tri-state
SIN[A,B]	—
SOUT[A,B]	High(1)
STEP#	High
STROBE#	Tri-state
TC	—
TRK0#	—
WE#	High
WP#	—
WRDATA#	High
X1/OSC	—
X2	—

NOTES:
1. During and immediately after a hard reset, this signal is an input for hardware configuration. After the hardware configuration time, these signals go to the state specified in the table.
2. If IORC# or IOWC# is asserted, IOCHRDY will be asserted by the IOCHRDY.
3. Dashes represent input signals.

2.8 Power And Ground

Signal Name	Type	Description
V_{SS}	I	**GROUND:** The ground reference for the 82091AA.
V_{CC}	I	**POWER:** The 5V/3.3V(1) modes are selected via strapping options at power-up (see Section 4.2, hardware Configuration). When strapping options (V_{SEL}) are set to 5V, the V_{CC} pins must be connected to 5V. When strapping options are set to 3.3V, the V_{CC} pins must be connected to 3.3V.
V_{CCF}	I	**POWER:** The 5V/3.3V(1) power supply for the 82091AA. In 5V or 3.3V power supply modes (non-mixed mode), the voltage applied to V_{CCF} is the same voltage as applied to V_{CC}. For mixed mode operations, 5V is applied to V_{CCF}. This voltage provides 5V reference for the parallel port and floppy disk controller interfaces. Note that in mixed mode, 3.3V is applied to V_{CC}.

NOTE:
1. 3.3V operation is available only in the 82091AA.

3.0 I/O ADDRESS ASSIGNMENTS

The 82091AA assigns CPU I/O address locations to its game port chip select, IDE interface, serial ports, parallel port, floppy disk controller, and the 82091AA configuration registers as indicated in Table 3. Except for the game port chip select (address 201h), address assignments are configurable. For example, the serial port can be assigned to one of eight address blocks. The parallel port can be assigned to one of three address blocks, and the IDE interface and floppy disk controller can be assigned to one of two address blocks. These address assignments are made during 82091AA configuration (either hardware configuration at powerup or a hard reset, or software configuration by programming the 82091AA configuration registers). In addition, the 82091AA configuration registers can be located at one of two address blocks during hardware configuration.

All of the 82091AA address locations are located in the host I/O address space. The address block assignments are shown in Table 3. The first hex address in the Address Block column represents the base address for that particular block.

5

Table 3. AIP Address Assignments

Address Block (ISA Bus)	Assignment
170–177h	IDE Interface—Secondary Address Block
1F0–1F7h	IDE Interface—Primary Address Block
201h	Game Port Chip Select
220–227h	Serial Port
228–22Fh	Serial Port
238–23Fh	Serial Port
26E–26Fh	82091AA Configuration Registers—Primary Address Block (022–023h on X-Bus)
278–27Fh	Parallel Port
2E8–2EFh	Serial Port
2F8–2FFh	Serial Port
338–33Fh	Serial Port
370–377h	Floppy Disk Controller—Secondary Address Block (376h and 377h are Shared with the IDE Drive Interface Secondary Address)
378–37Fh	Parallel Port
398–399h	82091AA Configuration Registers—Secondary Address Block (024–025h on X-Bus)
3BC–3BFh	Parallel Port (All Mopes Except EPP)
3E8–3EFh	Serial Port
3F0–3F7h	Floppy Disk Controller—Primary Address (3F6h and 3F7h are Shared with the IDE Drive Interface Primary Address)
3F8–3FFh	Serial Port
678–67Ah	Parallel Port (ECP Mode Peripheral Interface Protocol)
778–77Ah	Parallel Port (ECP Mode Peripheral Interface Protocol)
7BC–7BEh	Parallel Port (ECP Mode Peripheral Interface Protocol)

NOTES:
1. The 82091AA does not contain IDE registers. However, the 82091AA provides the address block assignments for accessing the IDE registers that are located in the IDE device.
2. The standard PC/AT* compatible logical I/O address assignments are supported. For example, COM1 (3F8–3FFh) and COM2 (2F8–2FFh) are part of the serial port assignments and LPT1 (3BC–3BFh), LPT2 (378–37Fh), and LPT3 (278–27Fh) are part of the parallel port assignments.

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

4.0 AIP CONFIGURATION

82091AA configuration consists of setting up overall device operations along with certain functions pertaining to the individual 82091AA modules (parallel port, serial ports, floppy disk controller, and IDE interface). Overall device operations include selecting the clock frequency, power supply voltage, and address assignment for the configuration registers. Overall device operations also enable/disable access to the configuration registers and provide interrupt signal level control. For the individual modules, 82091AA configuration includes module address assignment, interrupt control, module enable/disable, powerdown control, test mode control, module reset, and certain functions specific to each module. The remainder of the functions unique to each module are handled via the individual module registers.

Two methods are provided for configuring the 82091AA—hardware configuration via strapping options at powerup (or whenever RSTDRV is asserted) and software configuration by programming the configuration registers. (For information on hardware configuration, see Section 4.2, Hardware Configuration. For information on software configuration, see Section 4.1, Configuration Registers.)

NOTE:

1. There are four hardware configuration modes—SWMB (Software Motherboard), SWAI (Software Add-In), HWB (Hardware Basic), and HWE (Hardware Extended). Some of these modes can be used without the need for programming the 82091AA configuration registers. Other modes use both hardware configuration strapping options and programming the configuration registers to set up the 82091AA.
2. The 82091AA's operating power supply voltage level, 82091AA clock frequency, and address assignment for the 82091AA configuration registers can only be configured by hardware configuration.

4.1 Configuration Registers

82091AA Configuration Space contains 13 configuration registers. Four of the registers (Product and Revision Identification Registers and the 82091AA Configuration 1 and 2 Registers) provide control and status information for the entire chip. In addition, two registers each for the floppy disk controller, parallel port, serial port A, and serial port B and one register for the IDE interface provide certain module status and control information. The 82091AA configuration registers are indirectly addressed by first writing to the 82091AA Configuration Index Register as described in Section 4.1.1. Thus, the 13 configuration registers occupy two address locations in the host's I/O address space—one for indirectly selecting the specific configuration register and the other for transfering register data. All 82091AA configuration registers are 8-bits wide and are accessed as byte quantities.

Some of the 82091AA Configuration registers described in this section contain reserved bits. These bits are labeled "R". Software must deal correctly with fields that are reserved. On reads, software must use appropriate masks to extract the defined bits and not rely on reserved bits being any particular value. On writes, software must ensure that the values of reserved bit positions are preserved. That is, the value of reserved bit positions must first be read, merged with the new values for other bit positions, and then written back.

In addition to reserved bits within a register, the 82091AA configuration space contains address locations that are labeled "Reserved" (Table 5). While the 82091AA responds to accesses to these I/O addresses by completing the host cycle, writing to a reserved I/O address can result in unintended device operations. Values read from a reserved I/O address should not be used to permit future expansion and upgrades.

During a hard reset (RSTDRV asserted), the 82091AA sets its configuration registers to pre-determined **default** states. The default values are indicated in the individual register descriptions. The following nomenclature is used for register access attributes:

RO **Read Only.** If a register is read only, writes have no effect.

R/W **Read/Write.** A register with this attribute can be read and written. Note that individual bits in some read/write registers may be read only.

4.1.1 CFGINDX, CFGTRGT—CONFIGURATION INDEX REGISTER AND TARGET PORT

I/O Address: Hardware Configurable (see Table 4)
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

CFGINDX and CFGTRGT are used to access 82091AA configuration space where all of the 82091AA configuration registers are located. CFGINDX and CFGTRGT are located in the host I/O address space and the address locations are hardware configurable as shown in Table 4. CFGINDX is an 8-bit register that contains the address index of the 82091AA configuration register to be accessed. CFGTRGT is a port for reading data from or writing data to the configuration register whose index address matches the address stored in the CFGINDX Register. Thus, to access a configuration register, CFGINDX must first be programmed with the index address. A software example is provided in this section demonstrating how to access the configuration registers.

Table 4. Configuration Register Access Addresses

Address Selection	X-Bus Implementation		ISA Bus Implementation	
	Index	Target	Index	Target
Primary Address	22h	23h	26Eh	26Fh
Secondary Address	24h	25h	398h	399h

Table 5 summarizes the 82091AA configuration space. Following the table, is a detailed description of each register. The register descriptions are arranged in the order that they appear in Table 5.

Bit	Description
7:0	**82091AA Configuration Register Address Index:** Bits[7:0] correspond to SD[7:0].

Software Configuration

Access Addresses for the two Software Configuration Modes:

	Index	Target
For SWMB Mode Primary Address:	22h	23h
For SWMB Mode Secondary Address:	24h	25h
For SWAI, HWE, and HWB Modes Primary Address:	26Eh	26Fh
For SWAI, HWE, and HWB Modes Secondary Address:	398h	399h

The following pseudo code sequence could be used to access the configuration registers under SWMB primary address:

Configuration register write: OUT 22h, ConfigRegAddr
OUT 23h, ConfigRegData

Configuration register read: OUT 22h, ConfigRegAddr
IN 23h

Table 5. AIP Configuration Registers

82091AA Configuration Address Index	Abbreviation	Register Name	Access
00h	AIPID	Product Identification	RO
01h	AIPREV	Revision Identification	RO
02h	AIPCFG1	82091AA Configuration 1	R/W
03h	AIPCFG2	82091AA Configuration 2	R/W
04–0Fh	—	Reserved	—
10h	FCFG1	FDC Configuration	R/W
11h	FCFG2	FDC Power Management and Status	R/W
12–1Fh	—	Reserved	—
20h	PCFG	Parallel Port Configuration	R/W
21h	PCFG2	Parallel Port Power Management and Status	R/W
22–2Fh	—	Reserved	—
30h	SACFG1	Serial Port A Configuration	R/W
31h	SACFG2	Serial Port A Power Management and Status	R/W
32–3Fh	—	Reserved	—
40h	SBCFG1	Serial Port B Configuration	R/W
41h	SBCFG2	Serial Port B Power Management and Status	R/W
42–4Fh	—	Reserved	—
50h	ICFG	IDE Configuration	R/W
51–FFh	—	Reserved	—

NOTE:
Writing to a reserved I/O address should not be attempted and can result in unintended device operations.

4.1.2 AIPID—AIP IDENTIFICATION REGISTER

Index Address: 00h
Default Value: A0h
Attribute: Read Only
Size: 8 bits

Bit	Description
7:0	**AIP IDENTIFICATION (AIPID):** A value of A0h is assigned to the 82091AA. This 8-bit register combined with the 82091AA Revision Identification Register uniquely identifies the device.

4.1.3 AIPREV—AIP REVISION IDENTIFICATION

Index Address: 01h
Default Value: 00h
Attribute: Read Only
Size: 8 bits

This register contains two fields that identify the revision of the 82091AA device. The revision number will be incremented for every stepping, even if change is invisible to software.

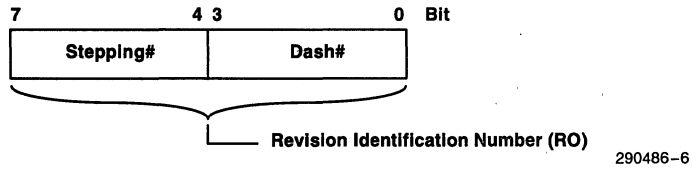


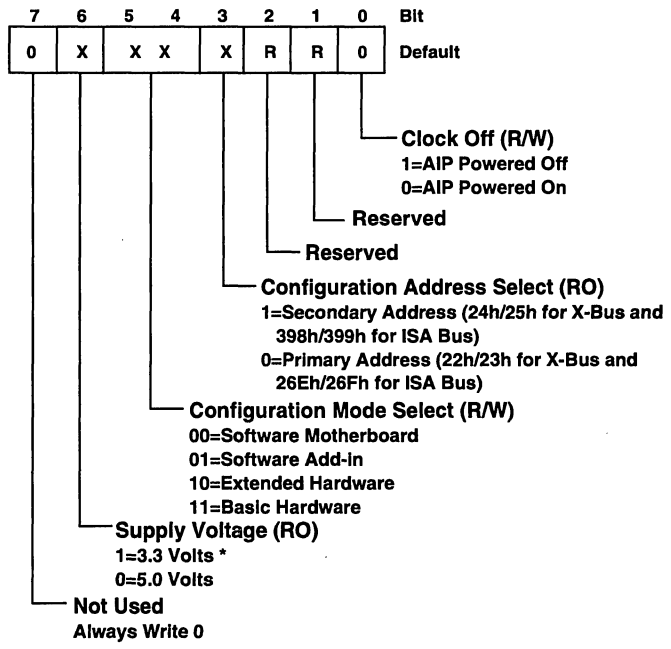
Figure 6. AIP Revision Identification Register

Bit	Description
7:4	**STEP NUMBER:** Contains the hexadecimal representation of the device stepping.
3:0	**DASH NUMBER:** Contains the hexadecimal representation of the dash number of the device stepping.

4.1.4 AIPCFG1—AIP CONFIGURATION 1 REGISTER

Index Address: 02h
Default Value: Depends upon hardware strap
Attribute: Read/Write
Size: 8 bits

The AIPCFG1 Register enables/disables master clock circuitry for power management, enables/disables access to the configuration registers, and selects the 82091AA configuration mode. This register provides status for certain hardware configuration selections—the 82091AA clock frequency, power supply voltage, and address assignment for the configuration registers (address locations of the INDEX and TARGET Registers).



NOTES:
*3.3V operation is available only in the 82091AA.
X = Value is determined by hardware strapping options as described in Section 4.2, Hardware Configuration.

Figure 7. AIP Configuration 1 Register

5

Bit	Description
7	**NOT USED:** Always write to 0.
6	**VOLTAGE SELECT (VSEL):** This bit indicates whether 3.3V or 5V has been selected for the operating power supply voltage during hardware configuration. A 1 indicates that 3.3V is selected and a 0 indicates that 5V is selected. This bit is read only and writes have no effect. **NOTE:** 3.3V operation is available only in the 82091AA.
5:4	**CONFIGURATION MODE SELECT (CFGMOD):** These bits indicate the configuration mode for the 82091AA. After a hard reset, these bits reflect the mode selected by hardware configuration. If configuration register access is not locked out during hardware configuration, software can change the configuration mode by writing to this field. For configuration mode details, (see Section 4.2, Hardware Configuration). **Bits[5:4] Configuration Mode** 0 0 Software Motherboard (SWMB) 0 1 Software Add-in (SWAI) 1 0 Extended Hardware (HWE) 1 1 Basic Hardware (HWB)
3	**CONFIGURATION ADDRESS SELECT (CFGADS):** This read only bit indicates the address assignment for the 82091AA configuration registers as selected by hardware configuration. Hardware configuration selects between primary addresses (22h/23h and 26Eh/26Fh) and secondary addresses (24h/25h and 398h/399h) for accessing the 82091AA configuration registers. When CFGADS = 0, the primary addresses are selected and when CFGADS = 1, the secondary addresses are selected.
2	**RESERVED**
1	**RESERVED**
0	**CLOCK OFF (CLKOFF):** The CLKOFF bit is used to implement clock circuitry power management. When CLKOFF = 0, the main clock circuitry is powered on. When CLKOFF = 1, the main clock circuitry is powered off. This capability is independent of the 82091AA's powerdown state. Note that auto powerdown mode and powerdown have no effect over the power state of the clock circuitry.

4.1.5 AIPCFG2—AIP CONFIGURATION 2 REGISTER

Index Address: 03h
Default Value: 0000 0RRR
Attribute: Read/Write
Size: 8 bits

This register selects the active signal level for IRQ[7:3]. The interrupt signals can be individually programmed for either active high or active low drive characteristics. The active high mode is ISA (non-share) compatible and has tri-state drive characteristic. The active low mode is EISA (sharable) compatible and has an open collector drive characteristic.

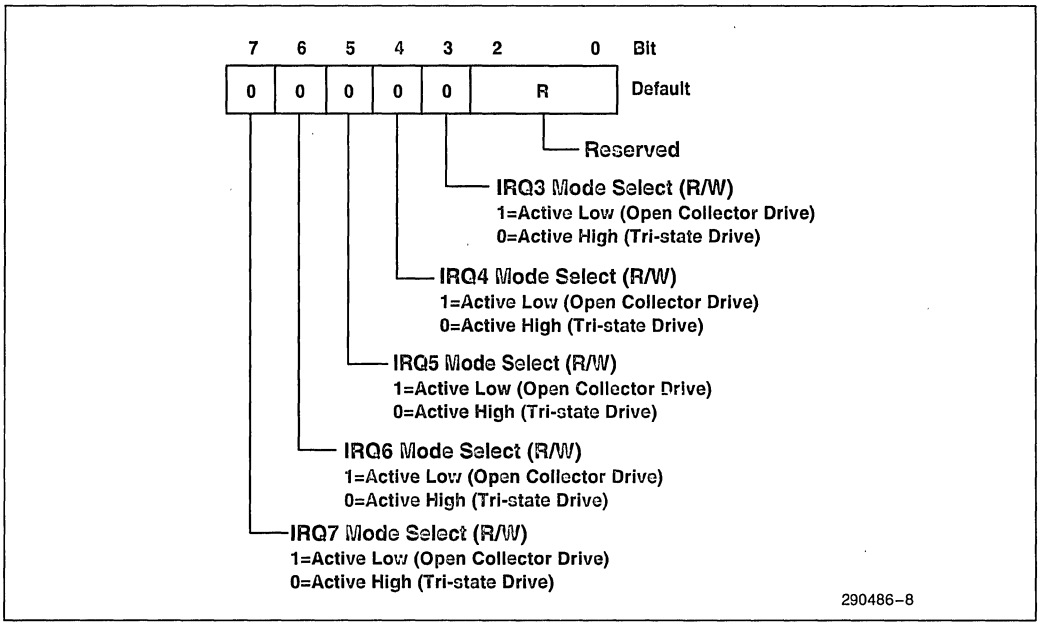


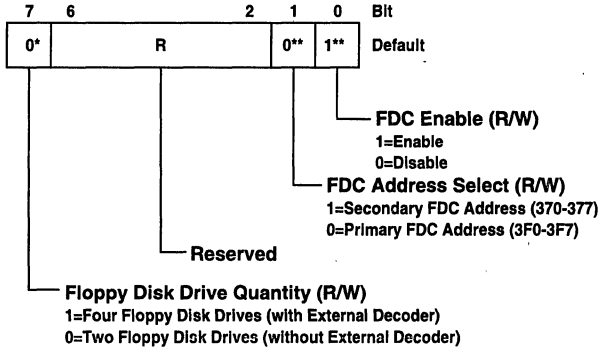
Figure 8. AIP Configuration 2 Register

Bit	Description
7	**IRQ7 MODE SELECT (IRQ7MOD):** When IRQ7MOD = 0, IRQ7 is an active high tri-state drive signal. When IRQ7MOD = 1, IRQ7 is an active low open collector drive signal.
6	**IRQ6 MODE SELECT (IRQ6MOD):** When IRQ6MOD = 0, IRQ6 is an active high tri-state drive signal. When IRQ6MOD = 1, IRQ6 is an active low open collector drive signal.
5	**IRQ5 MODE SELECT (IRQ5MOD):** When IRQ5MOD = 0, IRQ5 is an active high tri-state drive signal. When IRQ5MOD = 1, IRQ5 is an active low open collector drive signal.
4	**IRQ4 MODE SELECT (IRQ4MOD):** When IRQ4MOD = 0, IRQ4 is an active high tri-state drive signal. When IRQ4MOD = 1, IRQ4 is an active low open collector drive signal.
3	**IRQ3 MODE SELECT (IRQ3MOD):** When IRQ3MOD = 0, IRQ3 is an active high tri-state drive signal. When IRQ3MOD = 1, IRQ3 is an active low open collector drive signal.
2:0	**RESERVED**

4.1.6 FCFG1—FDC CONFIGURATION REGISTER

Index Address: 10h
Default Value: 0RRR RR01
Attribute: Read/Write
Size: 8 bits

This register selects between a 2 and 4 floppy drive system, selects primary/secondary ISA address range for the FDC, and enables/disables the FDC. All bits in this register are read/write.



NOTES:
*Default shown is for SWMB, SWAI, and HWB hardware configuration modes. For HWE, the default is determined by hardware strapping options as described in Section 4.2, Hardware Configuration.
**Default shown is for SWMB and SWAI configuration modes. For HWB and HWE configuration modes, the default is determined by hardware strapping options as described in Section 4.2, Hardware Configuration.

Figure 9. FDC Configuration Register

Bit	Description
7	**FLOPPY DISK DRIVE QUANTITY (FDDQTY):** This bit selects between two and four floppy disk drive capability. When FDDQTY = 0, the 82091AA can control two floppy disk drives directly without an external decoder. When FDDQTY = 1, the 82091AA can control four floppy disk drives with an external decoder. When FDDQTY = 1, the PDEN feature in the powerdown command is disabled. For further details, see Appendix A, FDC Four Drive Support. This bit can be configured by hardware extended configuration (HWE) at powerup. For all other hardware configuration modes (SWMB, SWAI, and HWB), the floppy disk drive quantity is not configurable by hardware strapping options and defaults to 2 drives.
6:2	**RESERVED**
1	**FLOPPY DISK CONTROLLER ADDRESS SELECT (FADS):** When FADS = 0, the primary FDC address (3F0–3F7) is selected. When FADS = 1, the secondary FDC address (370–377) is selected. For SWMB and SWAI configuration modes, the default is 0 (primary address). For HWB and HWE hardware configuration modes, the default is determined by signal pin strapping options.
0	**FLOPPY DISK CONTROLLER ENABLE (FEN):** This bit enables/disables the FDC. When FEN = 1, the FDC is enabled. When FEN = 0, the FDC module is disabled. For SWMB and SWAI configuration modes, the default is 1 (enabled). For HWB and HWE hardware configuration modes, the default is determined by signal pin strapping options. Note that, when the FDC is disabled, IRQ6 and FDDREQ are tri-stated.

4.1.7 FCFG2—FDC POWER MANAGEMENT AND STATUS REGISTER

Index Address: 11h
Default Value: RRRR 0000
Attribute: Read/Write
Size: 8 bits

This register enables/disables FDC auto powerdown and can place the FDC into direct powerdown. The register also provides FDC idle status and FDC reset control.

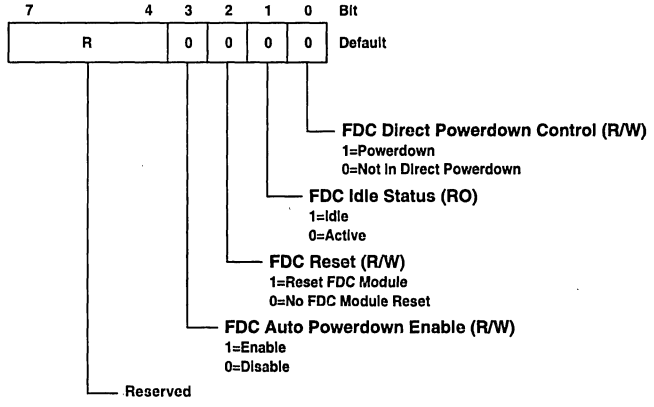


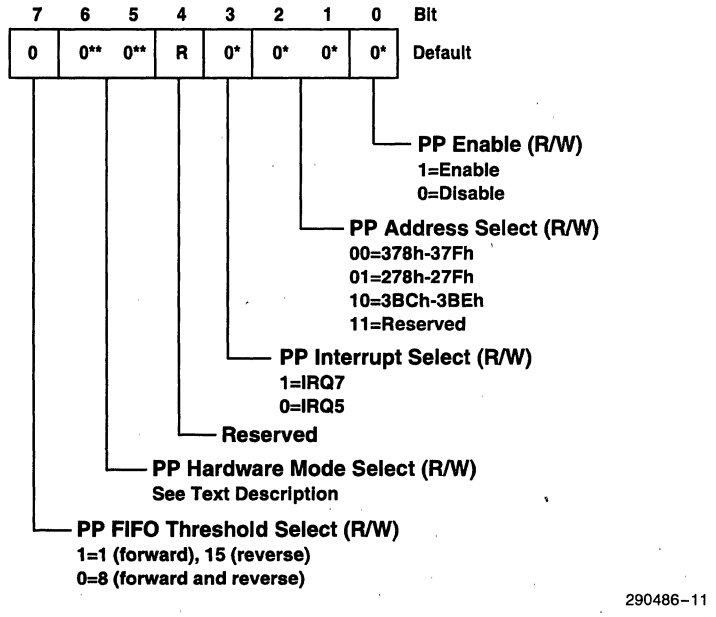
Figure 10. FDC Power Management and Status Register

Bit	Description
7:4	**RESERVED**
3	**FLOPPY DISK AUTO POWERDOWN ENABLE (FAPDN):** This bit is used to enable/disable auto powerdown for the FDC. When FAPDN = 1, the FDC will enter auto powerdown when the required conditions are met. When FAPDN = 0, FDC auto powerdown is disabled.
2	**FLOPPY DISK CONTROLLER RESET (FRESET):** FRESET is a reset for the FDC. When FRESET = 1, the FDC is reset (i.e., all programming and current state information is lost). FRESET = 1 has the same affect on the FDC as a hard reset (asserting the RSTDRV signal). When resetting the FDC via this configuration bit, the software must toggle this bit and ensure the reset active time (FRESET = 1) of 1.13 μs minimum is met.
1	**FLOPPY DISK CONTROLLER IDLE STATUS (FIDLE):** When the FDC is in the idle state, this bit is set to 1 by the 82091AA hardware. In the idle state the FDC's Main Status Register (MSR) = 80h, IRQ6 = inactive, and the head unload timer has expired. When the FDC exits its idle state, this bit is set to 0. This bit is read only.
0	**FLOPPY DISK CONTROLLER POWERDOWN (FDPDN):** When FDPDN is set to 1, the FDC is placed in direct powerdown. Once in powerdown the following procedure should be used to bring the FDC out of powerdown: • Write this bit low • Apply a hardware reset (via bit 2 of this register) or a software reset (via either bit 2 of the FDC's DOR or bit 7 of the FDC's DSR). **NOTE:** A hard reset via the RSTDRV pin also removes the FDC powerdown.

4.1.8 PCFG1—PARALLEL PORT CONFIGURATION REGISTER

Index Address: 20h
Default Value: 000R 0000
Attribute: Read/Write
Size: 8 bits

The PCFG1 Register enables/disables the parallel port, selects the parallel port address, and selects the parallel port interrupt. This register also selects the hardware operation mode for the parallel port.



NOTES:
*Default shown is for SWMB and SWAI configuration modes. For HWB and HWE modes, the default is determined by hardware configuration options as described in Section 4.2, Hardware Configuration.
**Default shown is for SWMB, SWAI, and HWB configuration modes. For HWE mode, the default is determined by hardware configuration options as described in Section 4.2, Hardware Configuration.

Figure 11. Parallel Port Configuration Register

Bit	Description
7	**PARALLEL PORT FIFO THRESHOLD SELECT (PTHRSEL):** This bit controls the FIFO threshold and only affects parallel port operations when the parallel port is in ECP mode or ISA-Compatible FIFO mode. When PTHRSEL = 1, the FIFO threshhold is 1 in the forward direction and 15 in the reverse direction. When PTHRSEL = 0, the FIFO threshold is 8 in both directions. This bit can only be programmed when the parallel port is in ISA-Compatible or PS/2-Compatible mode. These modes can be selected via bits[6:5] of this register or the ECP Extended Control Register (ECR). **NOTE:** In the reverse direction, a threshold of 15/8 means that a request (DMA or Interrupt is enabled) is generated when 15/8 bytes are in the FIFO. In the forward direction, a threshold of 1/8 means that a request is generated when 1/8 byte locations are available.
6:5	**PARALLEL PORT HARDWARE MODE SELECT (PPHMOD):** This field selects the parallel port hardware mode. The ISA-Compatible mode is for compatibility and nibble mode peripheral interface protocols. The PS/2-Compatible mode is for the byte mode peripheral interface protocol. The EPP and ECP modes are for the EPP and ECP mode peripheral interface protocols, respectively. This field can be configured by strapping options at powerup for hardware extended configuration (HWE) mode only. For all other hardware configuration modes (SWMB, SWAI, and HWB), the default is 00 (ISA-Compatible). **Bits [6:5] — Read — Write** 0 0 — ISA-Compatible — ISA-Compatible(1) 0 1 — PS/2-Compatible — PS/2-Compatible(1) 1 0 — EPP — EPP(1, 3) 1 1 — ECP(2) — Reserved; do not write(2) **NOTES:** 1. ISA-Compatible, PS/2-Compatible, and EPP modes are selected via this field or hardware configuration. In addition, ISA-Compatible and PS/2-Compatible modes can be selected via the ECP Extended Control Register (ECR). When the ECR is programmed for one of these two modes (ECR[7:5] = 000, 001), this field is updated to match the selected mode. 2. ECP Mode can not be entered by programming this field. ECP Mode can only be selected through the ECR. When the ECR is programmed for ECP mode, the 82091AA sets this field to 11. 3. Parallel port interface signals controlled by the PCON Register (SELECTIN#, INIT#, AUTOFD#, and STROBE#) should be negated before entering EPP mode.
4	**RESERVED**
3	**PARALLEL PORT IRQ SELECT (PIRQSEL):** When PIRQSEL = 1, IRQ7 is selected as the parallel port interrupt. When PIRQSEL = 0, IRQ5 is selected as the parallel port interrupt. This field can be configured by strapping options at powerup for HWB and HWE modes only. For all other hardware configuration modes (SWMB and SWAI), the default is 0 (IRQ5).

Bit	Description
2:1	**PARALLEL PORT ADDRESS SELECT (PADS):** This field selects the address for the parallel port as follows: **Bits[2:1] Address Parallel Port Hardware Mode** 0 0 378–37F All 0 1 278–27F All 1 0 3BC–3BE All except EPP 1 1 Reserved None, do not write This field can be configured by strapping options at powerup for HWB and HWE modes only. For all other hardware configuration modes (SWMB and SWAI), the default is 00 (378h–37Fh). Note that the SWMB and SWAI default settings for PIRQSEL (bit 3) and PADS (bits[2,1]) do not match a standard PC/AT* combination for address assignment and interrupt setting. However, for SWMB and SWAI, the parallel port defaults to a disabled condition and this register must be programmed to enable the parallel port (i.e., bit 0 set to 1). At this time, the selections for interrupt and address assignments should be made.
0	**PARALLEL PORT ENABLE (PEN):** When PEN = 0, the parallel port is disabled. When PEN = 1, the parallel port is enabled. This bit can be configured by hardware strapping options at powerup for HWB and HWE modes only. For all other hardware configuration modes (SWMB and SWAI), the default is 0 (disabled). Note that when the parallel port is disabled, IRQ[7,5] and PPDREQ are tri-stated.

4.1.9 PCFG2—PARALLEL PORT POWER MANAGEMENT AND STATUS REGISTER

Index Address: 21h
Default Value: RR0R 0000
Attribute: Read/Write
Size: 8 bits

This register enables/disables parallel port auto powerdown and can place the parallel port into a powerdown mode directly. The register also provides parallel port idle status, resets the parallel port, and reports FIFO underrun or overrun errors.

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

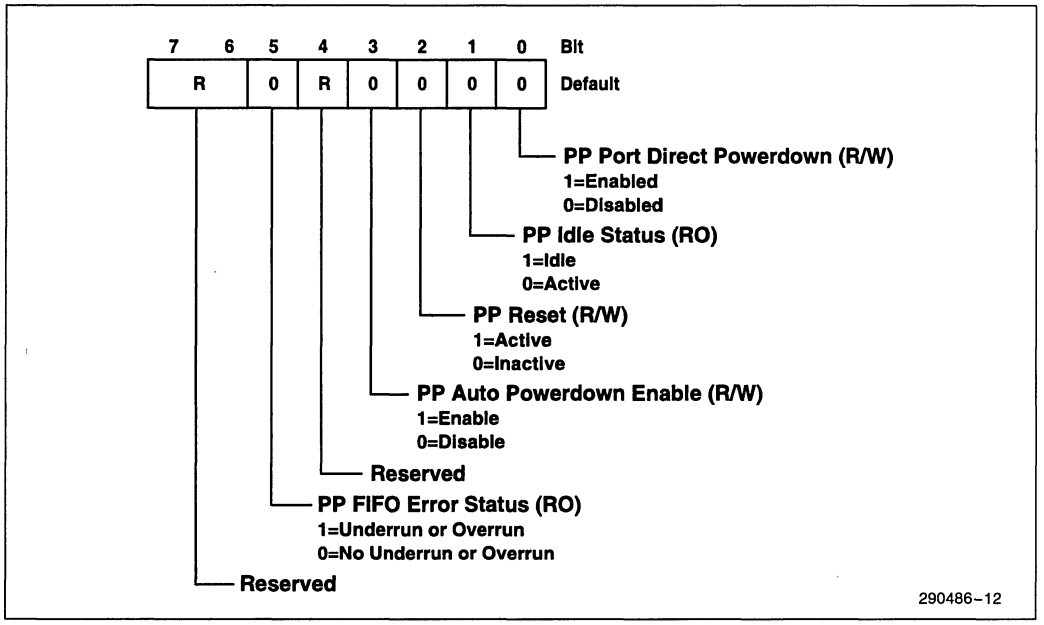


Figure 12. Parallel Port Power Management and Status Register

Bit	Description
7:6	**RESERVED**
5	**PARALLEL PORT FIFO ERROR STATUS (PFERR):** When PFERR = 1, a FIFO underrun or overrun condition has occurred. This bit is read only. Setting PRESET to 1 clears this bit to 0.
4	**RESERVED**
3	**PARALLEL PORT AUTO POWERDOWN ENABLE (PAPDN):** When PAPDN = 1, the parallel port can enter auto powerdown if the required auto powerdown conditions are met. When PAPDN = 0, auto powerdown is disabled.
2	**PARALLEL PORT RESET (PRESET):** When PRESET is set to 1, the parallel port is reset (i.e., all programming and current state information is lost). This is the same state the module would be in after a hard reset (RSTDRV asserted) to the 82091AA. When resetting the parallel port via this configuration bit, the software must toggle this bit and ensure the reset active time (PRESET = 1) of 1.13 μs minimum is met.
1	**PARALLEL PORT IDLE STATUS (PIDLE):** This bit reflects the idle state of the parallel port. When the parallel port is in an idle state (i.e., when the same conditions are met that apply to entering auto powerdown) the 82091AA sets this bit to 1. The parallel port idle state is defined as the FIFO empty and no activity on the parallel port interface. This bit is read only.
0	**PARALLEL PORT DIRECT POWERDOWN (PDPDN):** When PDPDN is set to 1, the parallel port enters direct powerdown. When PDPDN is set to 0, the parallel port is not in direct powerdown. Note that a parallel port module reset (PRESET bit in this register) also brings the parallel port out of the direct powerdown state.

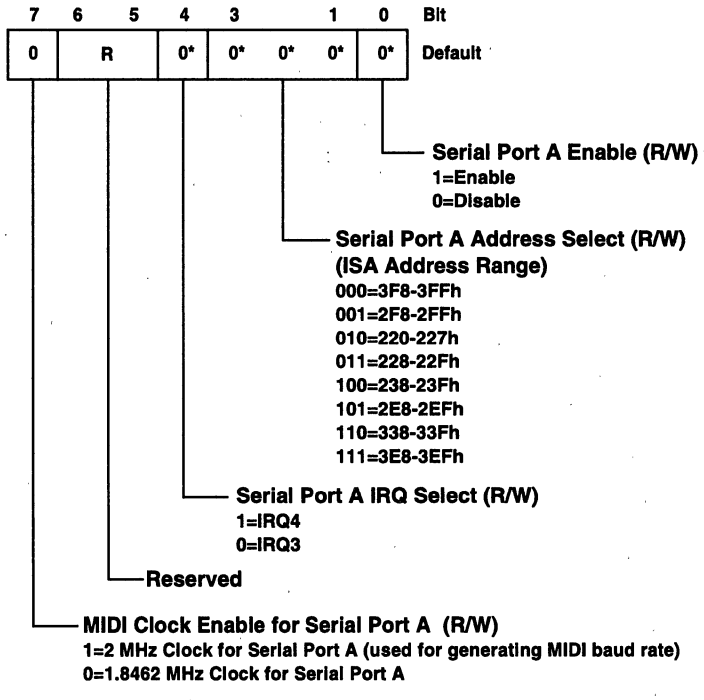
4.1.10 SACFG1—SERIAL PORT A CONFIGURATION REGISTER

Index Address: 30h
Default Value: 0RR0 0000
Attribute: Read/Write
Size: 8 bits

The SACFG1 register enables/disables Serial Port A, selects the Serial Port A address range, and selects between IRQ3 and IRQ4 as the Serial Port A interrupt. This register also selects the appropriate clock frequency for use with MIDI.

NOTES:

1. Through programming of this register and the SBCFG1 Register, the 82091AA permits serial ports A and B to be configured for the same interrupt assignment. However, software must take care in responding to interrupts correctly.
2. It is possible to enable and assign both serial ports to the same address through software. In this configuration, the 82091AA disables serial port B, but does not set serial port B into it's powerdown condition. Although this is a safe configuration for the 82091AA, it is not power conservative and is not recommended.



NOTE:
*Default shown is for SWMB and SWAI hardware configuration modes. For HWB and HWE modes, the default is determined by hardware strapping options as described in Section 4.2, Hardware Configuration.

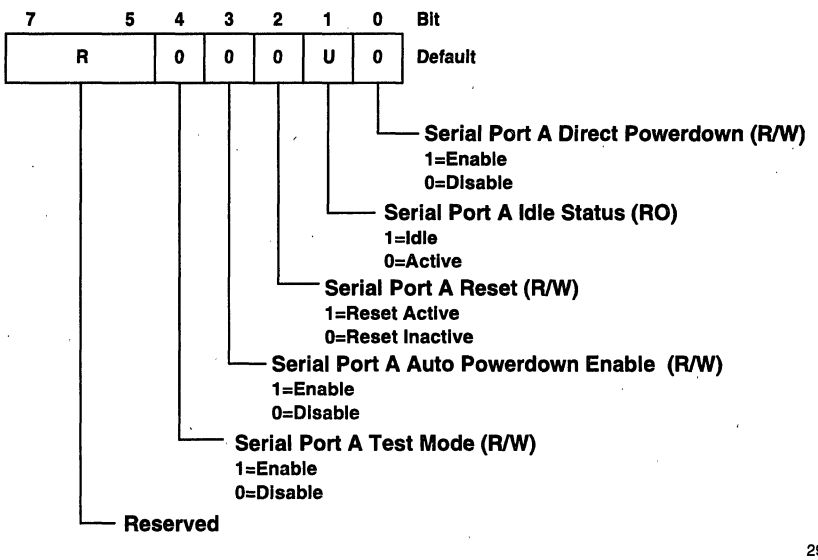
Figure 13. Serial Port A Configuration Register

Bit	Description
7	**MIDI CLOCK FOR SERIAL PORT A ENABLE (SAMIDI):** When SAMIDI = 1, the clock into Serial Port A is changed from 1.8462 MHz–2 MHz. The 2 MHz clock is needed to generate the MIDI baud rate. When SAMIDI = 0, the clock frequency is 1.8462 MHz.
6:5	**RESERVED**
4	**SERIAL PORT A IRQ SELECT (SAIRQSEL):** When SAIRQSEL = 0, IRQ3 is selected for the Serial Port A interrupt. When SAIRQSEL = 1, IRQ4 is selected for the Serial Port A interrupt. This bit can be configured by strapping options at powerup for HWB and HWE modes only. For SWMB and SWAI hardware configuration modes, the default is 0 (IRQ3). Note that, while the default address and IRQ assignments for SWMB and SWAI modes are the same for both serial ports, the serial ports are disabled and programming of this register is required for operation.
3:1	**SERIAL PORT A ADDRESS SELECT (SAADS):** This field selects the ISA address range for Serial Port A as follows: **Bits[3:1] — ISA Address Range** 0 0 0 — 3F8–3FFh 0 0 1 — 2F8–2FFh 0 1 0 — 220–227h 0 1 1 — 228–22Fh 1 0 0 — 238–23Fh 1 0 1 — 2E8–2EFh 1 1 0 — 338–33Fh 1 1 1 — 3E8–3EFh This field can be configured by strapping options at powerup for HWB and HWE modes only. For SWMB and SWAI hardware configuration modes, the default is 000 (3F8–3FFh). Note that, while the default address and IRQ assignments for SWMB and SWAI modes are the same for both serial ports, the serial ports are disabled and programming of this register is required for operation.
0	**SERIAL PORT A ENABLE (SAEN):** When SAEN = 1, Serial Port A is enabled. When SAEN = 0, Serial Port A is disabled. This bit can be configured by strapping options at powerup for HWB and HWE modes only. For SWMB and SWAI hardware configuration modes, the default is 0 (disabled).

4.1.11 SACFG2—SERIAL PORT A POWER MANAGEMENT AND STATUS REGISTER

Index Address: 31h
Default Value: RRR0 00U0
Attribute: Read/Write
Size: 8 bits

This register enables/disables the Serial Port A module auto powerdown and can place the module into a direct powerdown mode. The register also provides Serial Port A idle status, resets the Serial Port A module, and places Serial Port A into test mode.



NOTE:
U = Undefined

Figure 14. Serial Port A Power Management and Status Register

Bit	Description
7:5	**RESERVED**
4	**SERIAL PORT A TEST MODE (SATEST):** The serial port test mode provides user access to the output of the baud out generator. When SATEST = 1 (and the DLAB bit is 1 in the LCR), the Serial Port A test mode is enabled and the baud rate clock is output on the SOUTA pin (Figure 15). When SATEST = 0, the Serial Port A test mode is disabled.

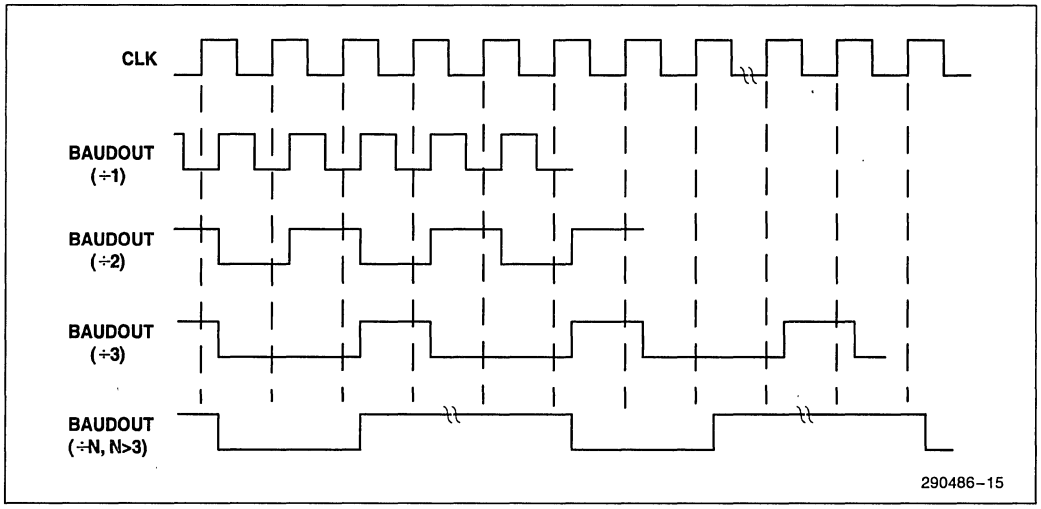


Figure 15. Test Mode Output (SOUTA and SOUTB)

Bit	Description
3	**SERIAL PORT A AUTO POWERDOWN ENABLE (SAAPDN):** This bit enables/disables auto powerdown. When SAAPDN = 1, Serial Port A can enter auto powerdown if the required conditions are met. The required conditions are that the transmit and receive FIFOs are empty and the timeout counter has expired. When SAAPDN = 0, auto powerdown is disabled.
2	**SERIAL PORT A RESET (SARESET):** When SARESET = 1, the Serial Port A module is reset (i.e. all programming and current state information is lost). This is the same state the module would be in after a hard reset (RSTDRV asserted). When resetting the serial port via this configuration bit, the software must toggle this bit and ensure the reset active time (SARESET = 1) of 1.13 μs minimum is met.
1	**SERIAL PORT A IDLE STATUS (SAIDLE):** When Serial Port A is in an idle state the 82091AA sets this bit to 1. Serial Port A is in the idle state when the transmit and receive FIFOs are empty and the timeout counter has expired. Note that these are the same conditions that apply to entering auto powerdown. When serial port A is not in an idle state, the 82091AA sets this bit to 0. Direct powerdown does not affect this bit and in auto powerdown SAIDLE is only set to a 1 if the receive and transmit FIFOs are empty. This bit is read only. During a hard reset (RSTDRV asserted), The 82091AA sets SAIDLE to 0. However, because the serial port is typically initialized by software before the idle conditions are met, the default state is shown as undefined.
0	**SERIAL PORT A DIRECT POWERDOWN (SADPDN):** When SADPDN = 1, Serial Port A is placed in direct powerdown mode. Setting this bit to 0 brings Serial Port A out of direct powerdown mode. Setting bit 2 (SARESET) of this register to 1 will also bring Serial Port A out of the direct powerdown mode. **NOTE:** Direct powerdown resets the receiver and transmitter portions of the serial port including the receive and transmit FIFOs. To ensure that the resetting of the FIFOs does not cause data loss, the SAIDLE bit should be 1 before placing the serial port into direct powerdown.

5

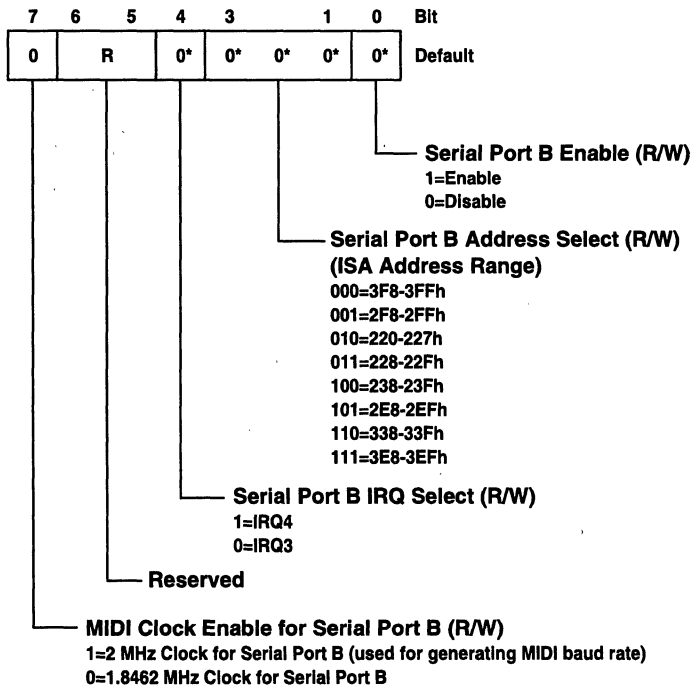
4.1.12 SBCFG1—SERIAL PORT B CONFIGURATION REGISTER

Index Address: 40h
Default Value: 0RR0 0000
Attribute: Read/Write
Size: 8 bits

The SBCFG1 register enables/disables Serial Port B, selects the Serial Port B address range, and selects between IRQ3 and IRQ4 as the Serial Port B interrupt. This register also selects the appropriate clock frequency for use with MIDI.

NOTES:

1. Through programming of this register and the SBCFG1 Register, the 82091AA permits serial ports A and B to be configured for the same interrupt assignment. However, software must take care in responding to interrupts correctly.
2. It is possible to enable and assign both serial ports to the same address through software. In this configuration, the 82091AA disables serial port B, but does not set serial port B into it's powerdown condition. Although this is a safe configuration for the 82091AA, it is not power conservative and is not recommended.



NOTE:
*Default shown is for SWMB and SWAI hardware configuration modes. For HWB and HWE modes, the default is determined by hardware strapping options as described in Section 4.2, Hardware Configuration.

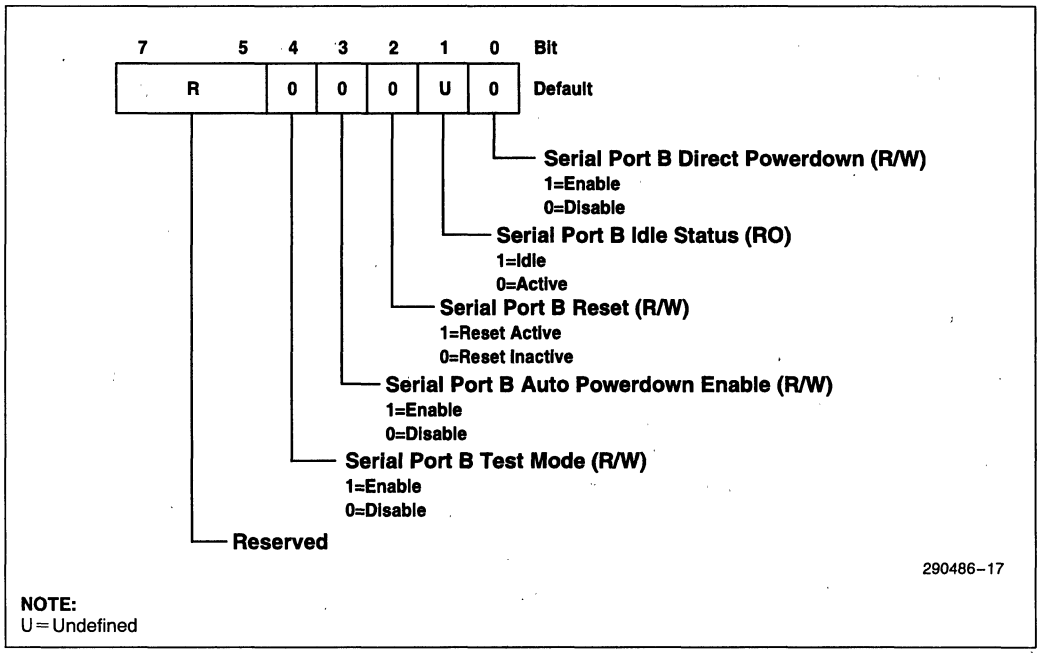
Figure 16. Serial Port B Configuration Register

Bit	Description
7	**MIDI CLOCK FOR SERIAL PORT B ENABLE (SBMIDI):** When SBMIDI = 1, the clock into Serial Port B is changed from 1.8462 MHz to 2 MHz. The 2 MHz clock is needed to generate the MIDI baud rate. When SBMIDI = 0, the clock frequency is 1.8462 MHz. The default value is 0.
6:4	**RESERVED**
4	**SERIAL PORT B IRQ SELECT (SBIRQSEL):** When SBIRQSEL = 0, IRQ3 is selected for the Serial Port B interrupt. When SBIRQSEL = 1, IRQ4 is selected for the Serial Port B interrupt. The default value is 0. This bit can be configured by strapping options at powerup for HWB and HWE modes only. For SWMB and SWAI configuration modes, the default is 0 (IRQ3). Note that, while the default address and IRQ assignments for SWMB and SWAI modes are the same for both serial ports, the serial ports are disabled and programming of this register is required for operation.
3:1	**SERIAL PORT B ADDRESS SELECT (SBADS):** This field selects the ISA address range for Serial Port B as follows: **Bits[3:1] — ISA Address Range** 0 0 0 — 3F8–3FFh 0 0 1 — 2F8–2FFh 0 1 0 — 220–227h 0 1 1 — 228–22Fh 1 0 0 — 238–23Fh 1 0 1 — 2E8–2EFh 1 1 0 — 338–33Fh 1 1 1 — 3E8–3EFh This field can be configured by strapping options at powerup for HWB and HWE modes only. For SWMB and SWAI configuration modes, the default is 000 (3F8–3FFh). Note that, while the default address and IRQ assignments for SWMB and SWAI modes are the same for both serial ports, the serial ports are disabled and programming of this register is required for operation.
0	**SERIAL PORT B ENABLE (SBEN):** When SBEN = 1, Serial Port B is enabled. When SAEN = 0, Serial Port B is disabled. This bit can be configured by strapping options at powerup for HWB and HWE modes only. For SWMB and SWAI configuration modes, the default is 0 (disabled).

4.1.13 SBCFG2—SERIAL PORT B POWER MANAGEMENT AND STATUS REGISTER

Index Address: 41h
Default Value: RRR0 00U0
Attribute: Read/Write
Size: 8 bits

This register enables/disables the Serial Port B module auto powerdown and can place the module into a powerdown mode directly. The register also provides Serial Port B idle status, resets the Serial Port B module, and enables/disables Serial Port B test mode.



NOTE:
U = Undefined

Figure 17. Serial Port B Power Management and Status Register

Bit	Description
7:5	**RESERVED**
4	**SERIAL PORT B TEST MODE (SBTEST):** The serial port test mode provides user access to the output of the baud out generator. When SBTEST = 1 (and the DLAB bit is 1 in the LCR), the Serial Port B test mode is enabled and the baud rate clock is output on the SOUTB pin (Figure 15). When SBTEST = 0, the Serial Port B test mode is disabled.
3	**SERIAL PORT B AUTO POWERDOWN ENABLE (SBAPDN):** This bit enables/disables auto powerdown. When SBAPDN = 1, Serial Port B can enter auto powerdown if the required conditions are met. The required conditions are that the transmit and receive FIFOs are empty and the timeout counter has expired. When SBAPDN = 0, auto powerdown is disabled.
2	**SERIAL PORT B RESET (SBRESET):** When SBRESET = 1, Serial Port B is reset (i.e., all programming and current state information is lost). This is the same state the module would be in after a hard reset (RSTDRV asserted). When resetting the serial port via this configuration bit, the software must toggle this bit and ensure the reset active time (SBRESET = 1) of 1.13 μs minimum is met.
1	**SERIAL PORT B IDLE STATUS (SBIDLE):** When Serial Port B is in an idle state the 82091AA sets this bit to 1. Serial Port B is in the idle state when the transmit and receive FIFOs are empty and the timeout counter has expired. Note that these are the same conditions that apply to entering auto powerdown. When serial port B is not in an idle state, the 82091AA sets this bit to 0. Direct powerdown does not affect this bit and in auto powerdown, this bit is only set to a 1 if the receive and transmit FIFOs are empty. This bit is read only. During a hard reset (RSTDRV asserted), the 82091AA sets this bit to 0. However, because the serial port is typically initialized by software before the idle conditions are met, the defaullt state is shown as undefined.
0	**SERIAL PORT B DIRECT POWERDOWN (SBDPDN):** When SBDPDN = 1, Serial Port B is placed in powerdown mode. Setting this bit to 0 brings the module out of direct powerdown mode. Setting bit 2 (SBRESET) of this register to 1 will also bring Serial Port B out of the direct powerdown mode. **NOTE:** Direct powerdown resets the receiver and transmitter portions of the serial port including the receive and transmit FIFOs. To ensure that the resetting of the FIFOs does not cause data loss, the SBIDLE bit should be 1 before placing the serial port into direct powerdown.

5

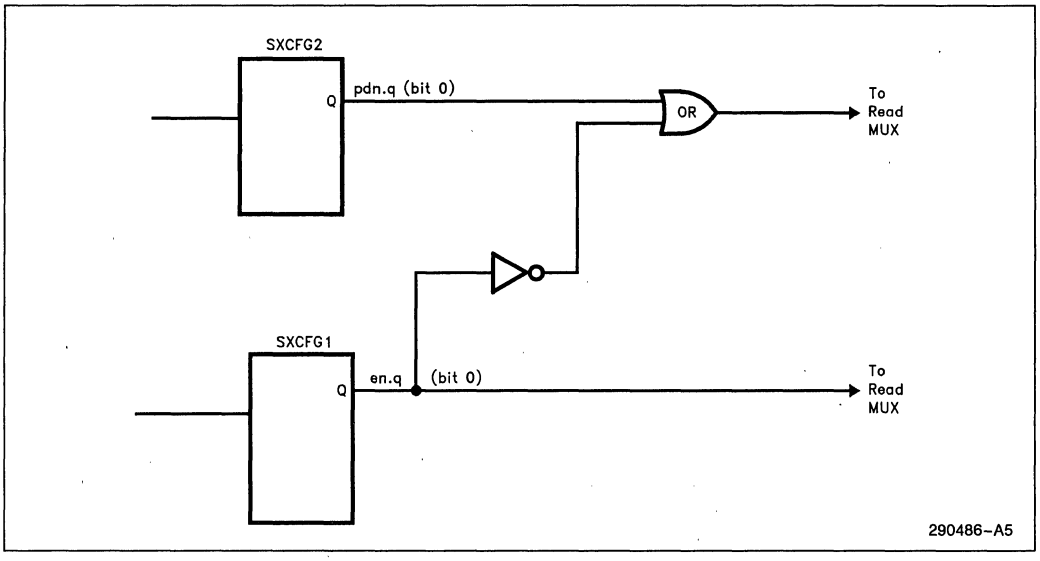
4.1.13.1 Serial Port A/B Configuration Register's SxEN and SxDPDN Bits

The bits which enable the serial ports (bit 0 in both the SACFG1 and SBCFG1 registers) and the bits which provide for serial port direct powerdown (bit 0 in both the SACFG2 and SBCFG2 registers) are not mutually exclusive. The partial circuit and truth table for the two bits shows that it is possible to enable serial port A using SACFG1, for example, yet still read the serial port A SACFG2 direct powerdown bit as a "1".

When the SxCFG1 register bit 0 (serial port x enable) is written as a "1" (enable), the SxCFG2 register bit 0 (serial port x powerdown) does not change from a "1" (powerdown enable) to a "0" (powerdown disable). As can be seen in the circuit diagram, the READ activity does not see the register directly. Instead, a MUXed output is seen by the READ activity. The truth table for the two bits shows it is possible to enable a serial port yet still read the powerdown bit for that same port as a "1", or enabled.

Truth Table for Reading the Enable/Powerdown Bit Status

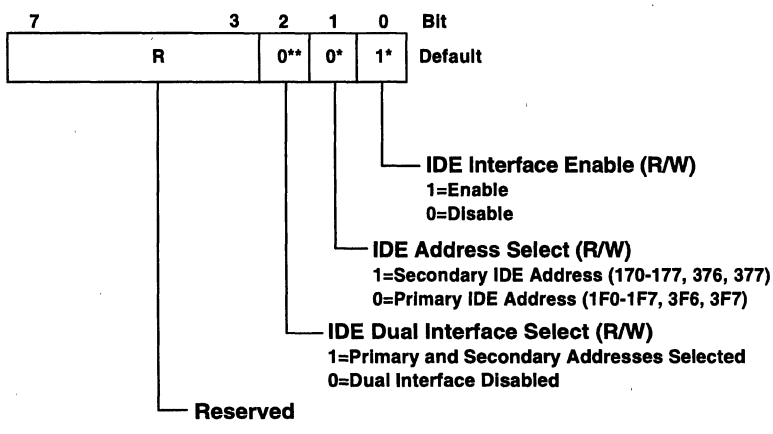
Write Activity		Read Activity	
SxCFG1 Enable	SxCFG2 Powerdown	SxCFG1 Enable	SxCFG2 Powerdown
0	0	0	1
1	0	1	0
1	1	1	1
0	1	0	1



4.1.14 IDECFG—IDE CONFIGURATION REGISTER

Index Address: 50h
Default Value: RRRR R001
Attribute: Read/Write
Size: 8 bits

The IDECFG Register sets up the 82091AA IDE interface. This register enables the IDE interface and selects the address for accessing the IDE.



NOTES:
* Default shown is for SWMB and SWAI configuration modes. For HWB and HWE hardware configuration modes, the default is determined by hardware strapping options as described in Section 4.2, Hardware Configuration.
** Not hardware configurable.

Figure 18. IDE Configuration Register

Bit	Description
7:3	**RESERVED**
2	**IDE DUAL SELECT (IDUAL):** When IDUAL = 0, the IDE address selection is determined by the IADS bit. When IDUAL = 1, both the primary and secondary IDE addresses are selected and the setting of the IADS bit does not affect IDE address selection.
1	**IDE ADDRESS SELECT (IADS):** When IADS = 0, the primary IDE address is selected (1F0h–1F7h, 3F6h, 3F7h). When IADS = 1, the secondary IDE address is selected (1F0h–1F7h, 376h, 377h). For all hardware configuration modes (SWMB, SWAI, HWB, and HWE), the default is determined by signal pin strapping options.
0	**IDE INTERFACE ENABLE (IEN):** When IEN = 0, the IDE interface is disabled (i.e., the IDE chip selects (IDECS[1:0]), DEN#, and HEN# are negated (remain inactive) for accesses to the IDE primary and secondary addresses). When IEN = 1, the IDE interface is enabled. For all hardware configuration modes (SWMB, SWAI, HWB, and HWE), the default is determined by signal pin strapping options.

4.2 Hardware Configuration

Hardware configuration provides a mechanism for configuring certain 82091AA operations at powerup. Four hardware configuration modes provide different levels of configuration depending on the type of application and the degree of hardware/software configuration desired. The hardware configuration modes are:

- Software Motherboard (SWMB)
- Software Add-In (SWAI)
- Hardware Extended (HWE)
- Hardware Basic (HWB)

These modes support a variety of system implementations. For example, with Hardware Basic (HWB) and Hardware Extended (HWE) modes, an extensive set of 82091AA configuration options are available for setting up the 82091AA at powerup. This permits the 82091AA to be used in systems without 82091AA software drivers. For many of these systems, access to the 82091AA configuration registers may not be necessary. As such, access to these registers can be disabled via hardware configuration. This option could be used to prevent software from inadvertently re-configuring the 82091AA.

NOTE:
If the 82091AA is configured in HWB or HWE configuration mode at powerup, and reconfiguration with software is desired, the 82091AA configuration mode must first be changed to SWAI configuration mode by writing the AIPCFG1 register. The 82091AA can then remain in SWAI configuration mode to accomodate software programmable configuration changes as desired.

Software Motherboard (SWMB) and Software Add-In (SWAI) modes provide a minimum hardware configuration in systems where software/firmware drivers are used for configuration. Because access to the 82091AA configuration registers after powerup/hardware configuration is needed, the SWMB and SWAI modes do not provide disabling access to these registers (i.e., the strapping of the HEN# signal has no effect).

The desired hardware configuration mode and options within the mode are selected by strapping certain 82091AA signal pins at powerup. These signal pins are sampled when the 82091AA receives a hard reset (via RSTDRV). This section describes how to select the configuration mode and options within the mode. The section also provides example hardware connection diagrams for the different modes.

4.2.1 SELECTING THE HARDWARE CONFIGURATION MODE

During powerup or a hard reset, four signal pins (DEN#, PPDIR/GCS#, DTRA, and HEN#) select the hardware configuration mode, I/O address assignment for the 82091AA configuration registers, and whether software access to these configuration registers is permitted. The following mnemonics and signal pins are assigned for these functions:

CFGMOD[1,0] **Hardware Configuration Mode.** The 82091AA samples the CFGMOD0 (DEN#) and CFGMOD1 (PPDIR/GCS#) signal pins to select one of the four hardware configuration modes as shown in Table 6.

CFGADS **82091AA Configuration Register Address Assignment.** The 82091AA samples the DTRA# signal (CFGADS function) to determine the address assignment of the 82091AA configuration registers as shown in Table 6. CFGADS works in conjunction with CFGDIS. Note that the 82091AA configuration register address assignment for Hardware Basic mode is not selectable.

CFGDIS **82091AA Configuration Register Disable.** The 82091AA samples CFGDIS (HEN# signal) to enable/disable access to the 82091AA configuration registers as shown in Table 6. Note that CFGDIS only affects the HWE and HWB modes.

NOTE:
For Extended Hardware Configuration, the time immediately following the RSTDRV pulse is required to complete the configuration time. If IORC#/IOWC# are asserted during this time, IOCHRDY will be negated (wait-states inserted) until the 82091AA configuration time expires.

Table 6. AIP Configuration Mode Register Address Assignment

CFGDIS (HEN#)	CFGMOD1 (PPDIR)	CFGMOD0 (DEN#)	CFGADS (DTRA#)	Configuration Mode	Configuration Register ISA Address (INDEX/TARGET)
X	0	0	0	SWMB	22h/23h
X	0	0	1	SWMB	24h/25h
X	0	1	0	SWAI	26Eh/26Fh
X	0	1	1	SWAI	398h/399h
0	1	0	0	HWE	26Eh/26Fh
0	1	0	1	HWE	398h/399h
1	1	0	X	HWE	Access Disabled
0	1	1	n/a	HWB	398h/399h
1	1	1	n/a	HWB	Access Disabled

4.2.2 SELECTING HARDWARE CONFIGURATION MODE OPTIONS

Within each hardware configuration mode, a number of options are available. For the HWB and HWE hardware configuration modes, the user can enable/disable the floppy disk controller and the IDE interface via the IDE chip select pins (see Table 7). If enabled, these signal pins also select the address assignment. For SWMB and SWAI configuration modes, these signal pins have no effect.

Table 7. FDC and IDE Enable/Disable

DDCFG1 (IDECS1#)	DDCFG0 (IDECS0#)	Floppy Disk Controller	IDE
0	0	Disable	Disable
0	1	Enabled (3F6–3F7h; Primary)	Disable
1	0	Enabled (370–377h; Secondary)	Enabled (170–177h; Secondary)
1	1	Enabled (3F6–3F7h; Primary)	Enabled (1F0–1F7h; Primary)

The 82091AA provides additional hardware configuration options through the SOUTA, SOUTB, RTSA#, RTSB#, DTRA#, and DTRB# signal pins as shown in Table 8. In the case of the Hardware Extended Mode, the 82091AA samples the signal pins at two different times (once for HWEa options and again for HWEb options). The timing for signal sampling is discussed in Section 4.2.3, Hardware Configuration Timing Relationships. The options provide configuration of the serial ports, floppy disk controller, parallel port, IDE interface, 82091AA operating power supply voltage, 82091AA clock frequency, and address assignment for the 82091AA configuration registers. Table 8 provides a matrix of the options available for each hardware configuration mode. The configuration options are selected as shown in Table 8 through Table 14.

Note that for the SWAI and SWMB modes, the selection of the operating frequency (CLKSEL), power supply voltage level (VSEL), and 82091AA configuration register address assignment (CFGADS) are the only hardware configuration options (Table 8). In these modes, software/firmware provides the remainder of the 82091AA configuration by programming the 82091AA configuration registers (see Section 4.1, Configuration Registers). For the SWAI and SWMB modes, the 82091AA modules are placed in the following states after powerup or a hard reset:

- Serial ports disabled
- Parallel port disabled
- FDC enabled for two drives (primary address)
- IDE enabled (primary address)

5

Table 8. Hardware Configuration Mode Option Matrix

Signal Name	Basic Hardware Configuration	Extended Hardware Configuration		Software Add-In Configuration	Software MotherBoard Configuration
	HWB	HWEa	HWEb	SWAI	SWMB
SOUTA	SPCFG0	CLKSEL(3)	SPCFG0	CLKSEL(3)	CLKSEL(3)
SOUTB	SPCFG1	PPMOD0	SPCFG1	—	—
RTSA#	SPCFG2	PPMOD1	SPCFG2	—	—
RTSB#	SPCFG3	FDDQTY	SPCFG3	—	—
DTRA#	PPCFG0	CFGADS	PPCFG0	CFGADS	CDGADS
DTRB#	PPCFG1	VSEL	PPCFG1	VSEL	VSEL

NOTES:
1. HWEa and HWEb reference the switching banks shown in Figure 22.
2. The following mnemonics are used in the table: SPCFGx = serial port configuration, PPCFGx = parallel port configuration, CLKSEL = clock select, PPMODx = parallel port hardware mode, FDDQTY = floppy disk drive quantity, VSEL = power supply voltage select, CFGADS = 82091AA configuration register address assignment select.
3. Always tie this signal low with a 10K resistor.

Table 9. Serial Port Address and Interrupt Assignments

SPCFG3 (RTSB#)	SPCFG2 (RTSA#)	SPCFG1 (SOUTB)	SPCFG0 (SOUTA)	Serial Port B		Serial Port A	
				Address Assignment	Interrupt Assignment	Address Assignment	Interrupt Assignment
0	0	0	0	Disable	—	Disable	—
0	0	0	1	Disable	—	3F8–3FFh	IRQ4
0	0	1	0	Disable	—	2F8–2FFh	IRQ3
0	0	1	1	Disable	—	3E8–3EFh	IRQ4
0	1	0	0	3F8–3FFh	IRQ4	Disable	—
0	1	0	1	3E8–3EFh	IRQ4	Disable	—
0	1	1	0	3F8–3FFh	IRQ4	2F8–2FFh	IRQ3
0	1	1	1	3F8–3FFh	IRQ4(1)	3E8–3EFh	IRQ4(1)
1	0	0	0	2F8–2FFh	IRQ3	Disable	—
1	0	0	1	2F8–2FFh	IRQ3	3F8–3FFh	IRQ4
1	0	1	0	Disable	—	2E8–2EFh	IRQ3
1	0	1	1	2F8–2FFh	IRQ3	3E8–3EFh	IRQ4
1	1	0	0	2E8–2EFh	IRQ3	Disable	—
1	1	0	1	2E8–2EFh	IRQ3	3F8–3FFh	IRQ4
1	1	1	0	2E8–2EFh	IRQ3(1)	2F8–2FFh	IRQ3(1)
1	1	1	1	2E8–2EFh	IRQ3	3E8–3EFh	IRQ4

NOTE:
1. In this configuration, the two serial ports share the same interrupt line. Responding correctly to interrupts generated in this configuration is the exclusive responsibility of software.

Table 10. Parallel Port Address and Interrupt Assignments

PPCFG1 (DTRB#)	PPCFG0 (DTRA#)	Parallel Port Address Assignment	Parallel Port Interrupt Assignment
0	0	Disable	—
0	1	378–37Fh	IRQ7
1	0	278–27Fh	IRQ5
1	1	3BC–3BFh	IRQ7

Table 11. Parallel Port Hardware Mode Select

PPMOD1 (RTSA#)	PPMOD0 (SOUTB)	Mode
0	0	ISA-Compatible
0	1	PS/2-Compatible
1	0	EPP
1	1	Reserved

NOTES:
1. PPMODx hardware configuration is effective in HWE mode only.
2. ECP mode is not selectable via hardware configuration.
3. For EPP mode, address assignment must be either 278h or 378h.

Table 12. AIP Clock Select

CLKSEL (SOUTA)	
0	24 MHz

NOTE:
Always tie this low.

Table 13. AIP Power Supply Voltage

VSEL (DTRB#)	Power Supply Voltage
0	5.0V Operation
1	3.3V Operation

NOTES:
1. VSEL hardware configuration is not available in HWB mode only.
2. To operate the 82091AA and all of the interfaces at 5V or 3.3V, both V_{CC} and V_{CCF} are connected to 5V or 3.3V power supplies, respectively. However, in the mixed mode, hardware configuration (V_{SEL}) is set to 3.3V, V_{CC} is connected to 3.3V, and V_{CCF} connected to 5V.
3. 3.3V operation is available only in the 82091AA.

Table 14. Floppy Drive Quantity Select

FDDQTY (RTSB#)	Number of Supported Floppy Drives
0	2 Floppy Drives
1	4 Floppy Drives

NOTES:
1. FDDQTY hardware configuration is effective in HWE mode only.
2. Four floppy drive support requires external logic to decode.

4.2.3 HARDWARE CONFIGURATION TIMING RELATIONSHIPS

The 82091AA samples all of the hardware configuration signals on the high-to-low transition of RSTDRV. For the HWB, SWMB, and SWAI modes, the 82091AA completes hardware configuration on this sampling (Figure 19). For HWE mode, the 82091AA samples some of the signals twice (Figure 20). The first sampling occurs on the high-to-low transition of RSTDRV. As Figure 22 shows (see Section 4.2.5, Extended Hardware Configuration Mode), the HC367 tri-states its outputs when RSTDRV is negated. This permits the strapping options from the HWEb block to be sampled. A short time after RSTDRV is negated (the time is specified in Section 11.0, Electrical Characteristics), the 82091AA samples the SOUTA, RTSA#, DTRA#, SOUTB, RTSB#, and DTRB# signals.

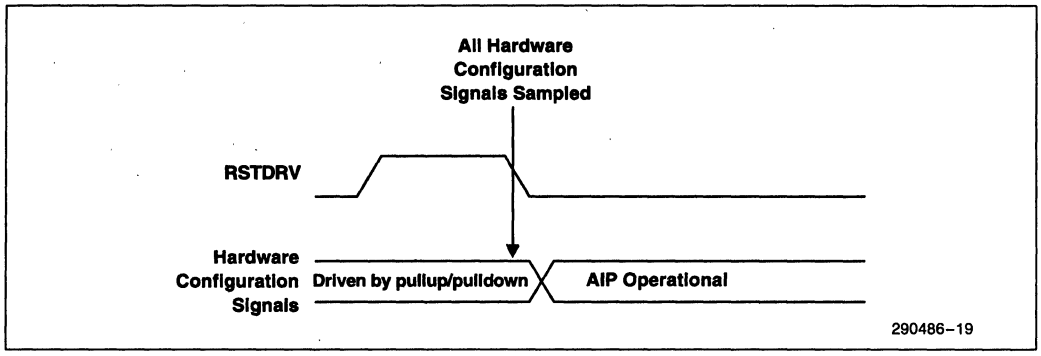


Figure 19. HWB, SWMB, and SWAI Hardware Configuration Mode Timing

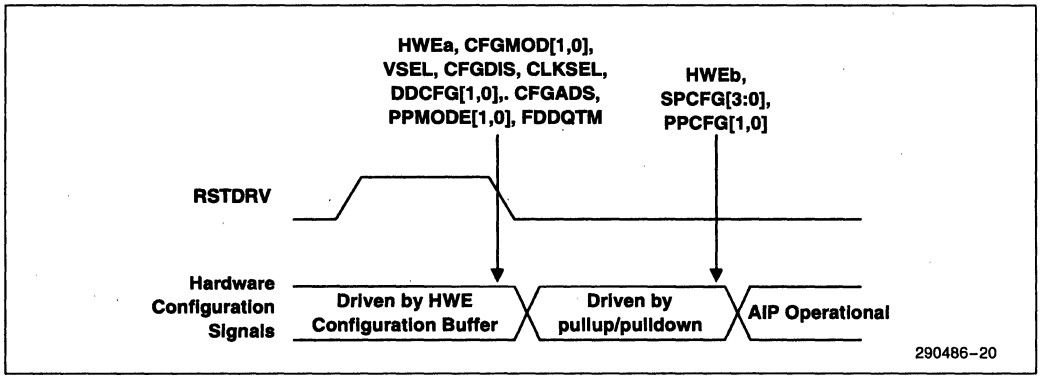


Figure 20. HWE Hardware Configuration Mode Timing

4.2.4 HARDWARE BASIC CONFIGURATION

The Hardware Basic configuration mode permits the user to assign addresses to the serial ports and parallel ports. This is achieved by sampling several of the serial port connections at the end of a hardware reset. The PPDIR/GCS# signal defaults to game port chip select output (GCS#). The 82091AA power supply voltage is not selectable in this mode and is fixed at 5V. The parallel port mode is set to ISA-Compatible. In addition, the FDC floppy drive support is set at two floppy drives. If configuration register access is enabled, the access address is fixed at 398h/399h. To reconfigure the 82091AA using software, the 82091AA configuration mode must be changed to SWAI mode (refer to AIPCFG1 register). Figure 21 shows the implementation of a basic hardware configuration.

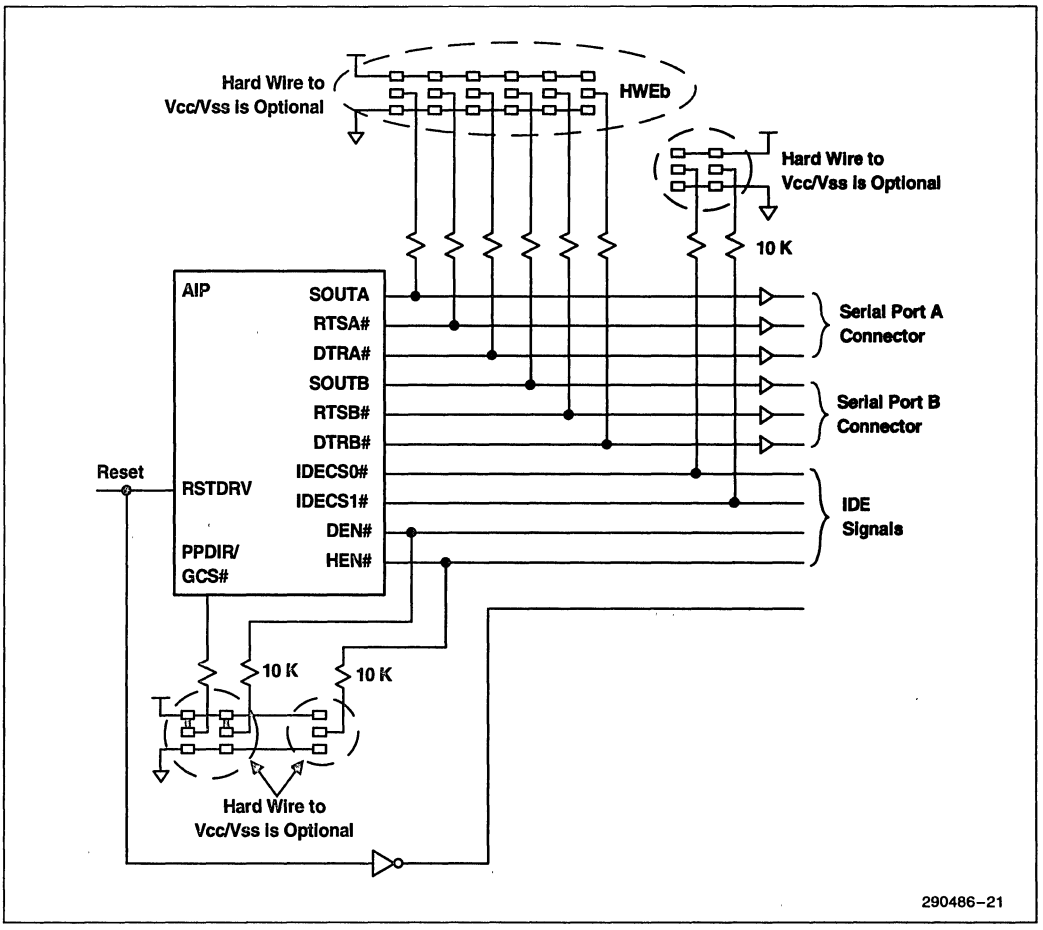


Figure 21. Hardware Basic Configuration

5

4.2.5 HARDWARE EXTENDED CONFIGURATION MODE

The Hardware Extended configuration mode provides all of the features of the Hardware Basic configuration mode. Additional features in Hardware Extended configuration permit the user to select quantity of floppy drives can be selected for either 2 or 4 floppy drive support. The 82091AA operating voltage is selectable between 3.3V* and 5V. In addition, the parallel port can be configured to operate in ISA-Compatible, PS/2-Compatible, or EPP modes. Hardware extended configuration provides these additional hardware configuration options by sampling the pins on the serial ports at two different times. When RSTDRV is asserted, the HC367 drives the values on SOUTA, RTSA#, DTRA#, SOUTB, RTSB#, and DTRB# (Figure 22). When RTSDRV is negated, the HC367 is disabled and these serial port signals are driven by HWEb pullup/down resistors. The PPDIR/GCS# signal defaults to a game port chip select (GCS#). To reconfigure the 82091AA using software, the 82091AA configuration mode must be changed to SWAI mode (refer to AIPCFG1 register).

NOTE:
*3.3V operation is only available in the 82091AA.

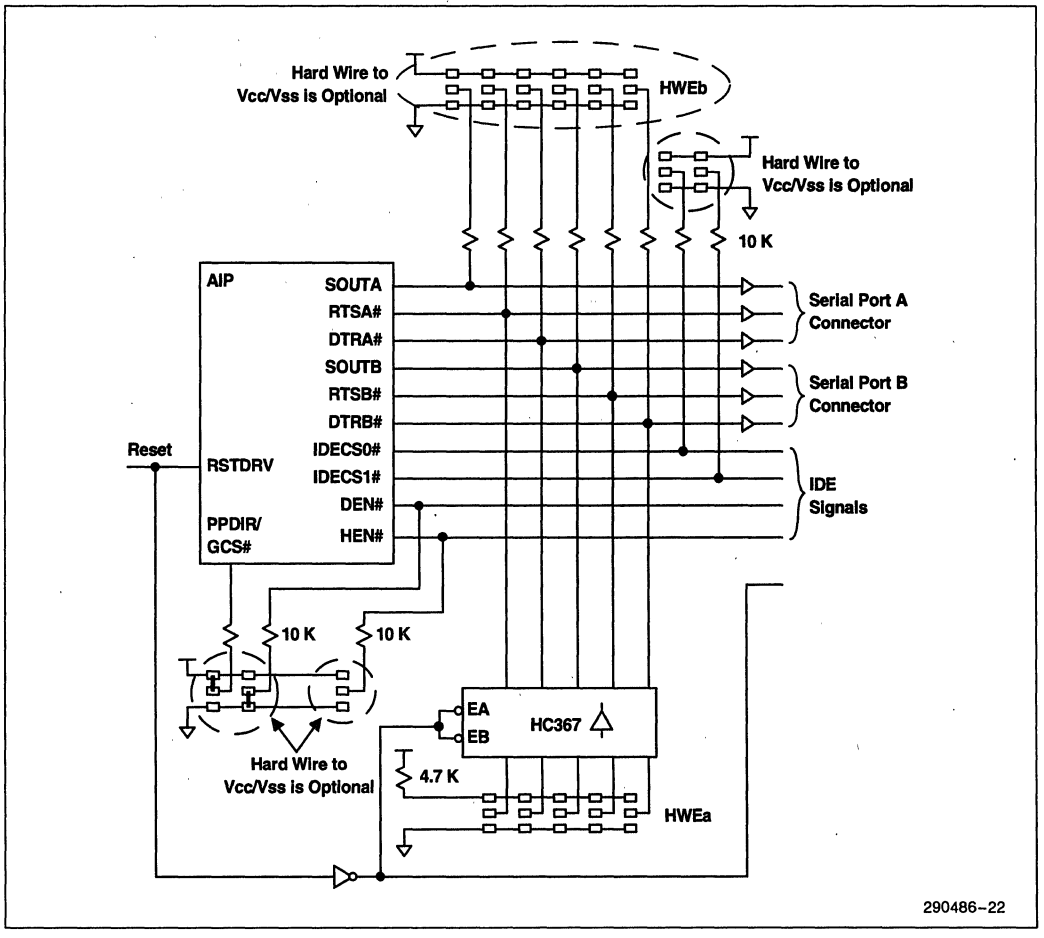


Figure 22. Hardware Extended Configuration

4.2.6 SOFTWARE ADD-IN CONFIGURATION

The Software Add-in configuration mode permits the user to assign the address for the 82091AA configuration registers, and select the power supply voltage for the 82091AA. The 82091AA configuration registers are accessible. The registers are located in the ISA Bus I/O address space and can be selected to be at either 398h/399h or 26Eh/26Fh. The PPDIR/GCS# signal defaults to a game port chip select (GCS#).

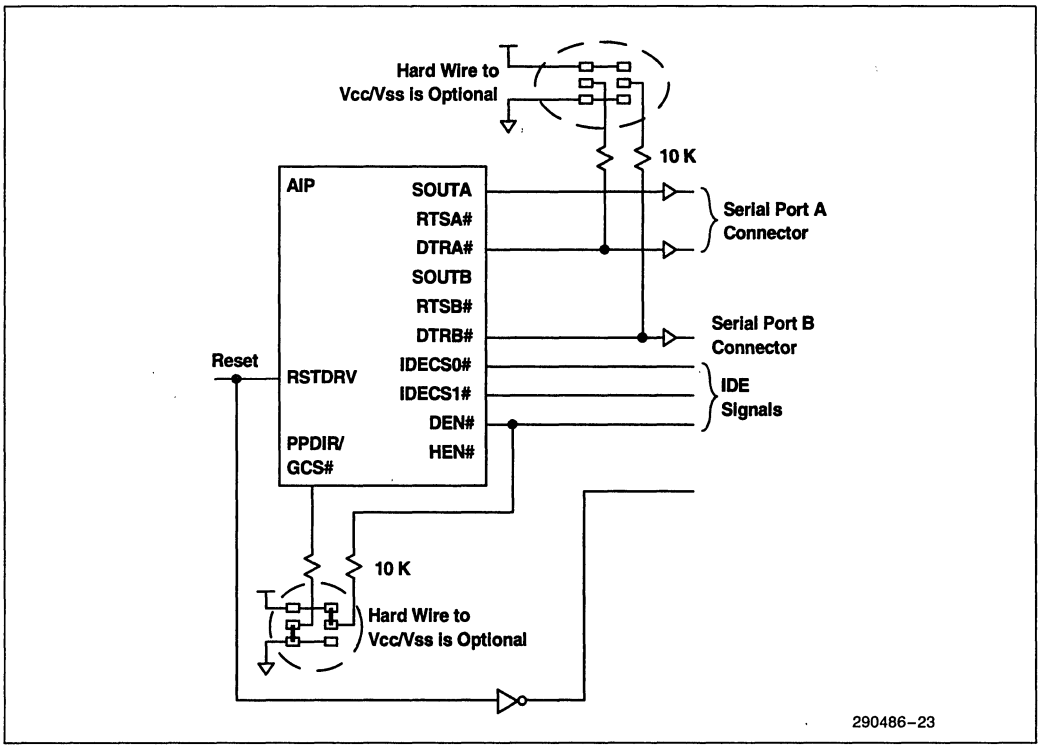


Figure 23. Software Add-In Configuration

5

4.2.7 SOFTWARE MOTHERBOARD CONFIGURATION

The Software Motherboard configuration mode permits the 82091AA to be located on the motherboard. In this mode, the 82091AA configuration registers are accessible via the X-Bus I/O address space and can be selected to be at either 22h/23h or 24h/25h. In addition, the user selects the power supply voltage for the 82091AA. The PPDIR/GCS# signal defaults to a Parallel Port Direction Control Output (PPDIR).

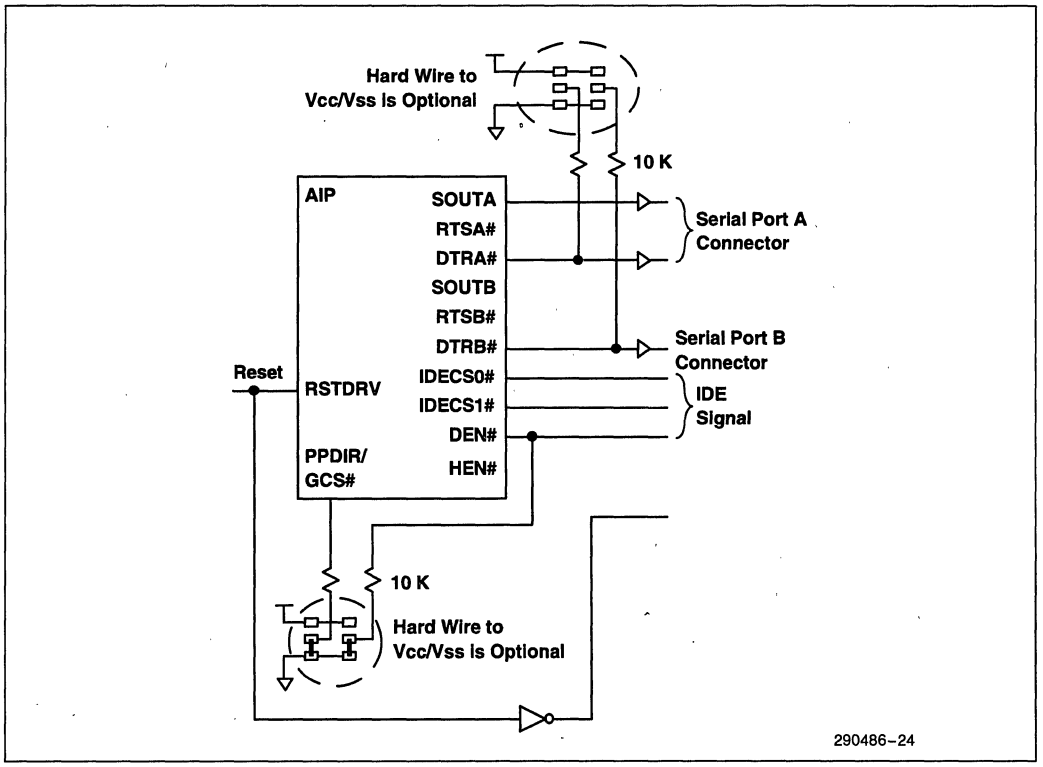


Figure 24. Software Motherboard Hardware Configuration

5.0 HOST INTERFACE

The 82091AA host interface is an 8-bit direct-drive (24 mA) ISA Bus/X-Bus interface that permits the CPU to access its registers through read/write operations in I/O space. These registers may be accessed by programmed I/O and/or DMA bus cycles. With the exception of the IDE Interface, all functions on the 82091AA require only 8-bit data accesses. The 16-bit access required for the IDE Interface is supported through the appropriate chip selects and data buffer enables from the 82091AA. The 82091AA does not participate in 16-bit IDE DMA transfers.

Although the 82091AA has an ISA/X-Bus host interface, there are a few features that differentiate it from conventional ISA/X-Bus peripherals. These features are as follows:

- **Internal, Configurable Chip Select Decode Logic.** SA[9:0] allow full decoding of the ISA I/O address space such that the functional modules contained in the 82091AA can be relocated to the desired I/O address. This feature can be used to resolve potential system configuration conflicts.
- **IOCHRDY for ISA Cycle Extension.** During certain I/O cycles to the parallel port controller in the 82091AA, it is necessary to extend the current bus cycle to match the access time of the device connected to the Parallel Port. The IOCHRDY signal is used by the 82091AA to extend ISA Bus cycles, as needed, according to the ISA protocol. IOCHRDY overrides all other strobes that attempt to shorten the bus cycle.
- **NOWS# for 3 BCLK I/O Cycles.** All programmed I/O accesses to 82091AA registers can be completed in a total of 3 BCLK cycles. This is possible because the 82091AA register access times have been minimized to allow data transfers to occur with shortened read/write control strobes. As a result, the 82091AA is well suited for use in embedded control designs that use an asynchronous microprocessor interface without any particular reference to ISA cycle timings.
- **DMA Transfers:** The 82091AA supports DMA compatible, type A, type B and type F DMA cycles. Some newer system DMA controllers are capable of generating fast DMA cycles (type F) on all DMA channels. If such a controller is used in conjunction with the 82091AA, it will be possible to accomplish a DMA transfer in 2 BCLKs.

The 82091AA ISA data lines (SD[7:0]) can be connected directly to the ISA Bus. If external buffers are used to isolate the SD[7:0] signals from the 240 pF loading of the ISA Bus, the DEN# signal can be used to control the external buffers as shown in Figure 25.

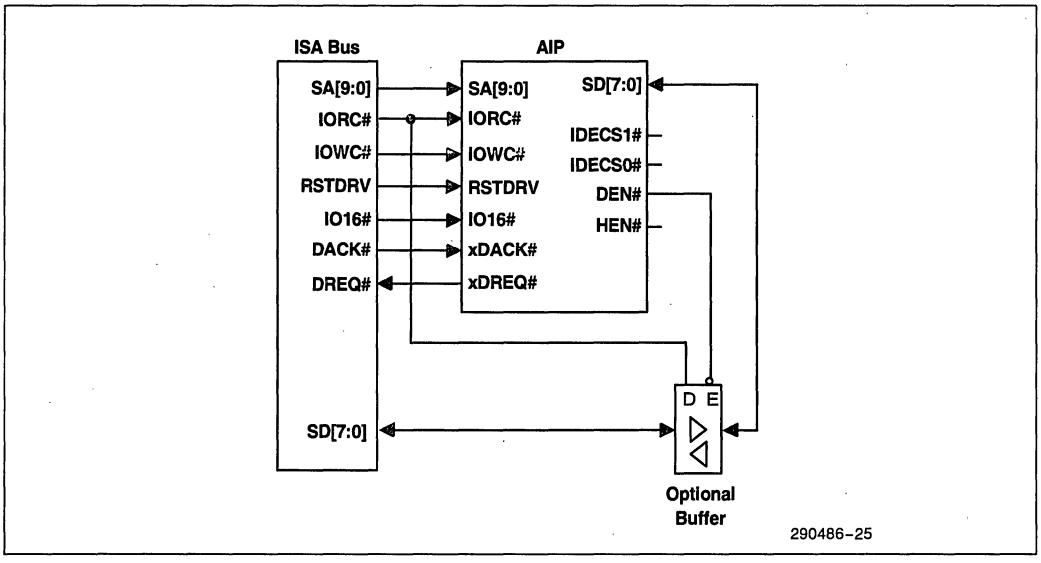


Figure 25. ISA Interface (with Optional Data Buffer)

6.0 PARALLEL PORT

The 82091AA parallel port can be configured for four parallel port modes. These parallel port modes and the associated parallel interface protocols are:

Parallel Port Mode	Parallel Interface Protocol
ISA-Compatible Mode	Compatibility, Nibble
PS/2-Compatible Mode	Byte
EPP Mode	EPP
ECP	ECP

ISA-Compatible, PS/2-Compatible, and EPP modes are selected through 82091AA configuration (see Section 4.0, AIP Configuration). ECP is selected by programming the ECP Extended Control Register (ECR).

In ISA-Compatible mode, the parallel port exactly emulates a standard ISA-style parallel port. The parallel port data bus (PD[7:0]) is uni-directional. The compatibility protocol transfers data to the peripheral device via PD[7:0] (forward direction). Note that the Nibble protocol permits data transfers from the peripheral device (reverse direction) by using four peripheral status signal lines to transfer 4 bits of data at a time.

PS/2-Compatible mode differs from ISA-Compatible mode by providing bi-directional transfers on PD[7:0]. A bit is added to the PCON Register to allow software control of the data transfer direction.

For both the ISA-Compatible and PS/2-Compatible modes, the actual data transfer over the parallel port interface is accomplished by software handshake (i.e., automatic hardware handshake is not used). Software controls data transfer by monitoring handshake signal status from the peripheral device via the PSTAT Register and controlling handshake signals to the peripheral device via the PCON Register.

EPP mode provides bi-directional transfers on PD[7:0]. The 82091AA automatically generates the address and data strobes in hardware.

ECP is a high performance peripheral interface mode. This mode uses an asynchronous automatic handshake to transfer data over the parallel port interface. In addition, the parallel port contains a FIFO for transferring data in ECP mode. The ECP register set contains an Extended Control Register (ECR) that provides a wide range of functions including the ability to operate the parallel port in either ECP, ISA-Compatible, or PS/2-Compatible modes.

NOTE:
In general, this document describes parallel port operations and functions in terms of how the 82091AA parallel port hardware operates. Detailed descriptions of the parallel interface protocols are beyond the scope of this document. Readers should refer to the proposed IEEE Standard 1284 for detailed descriptions of the Compatibility, Nibble, Byte, EPP, and ECP protocols.

Special circuitry on the 82091 prevents it from being powered up or being damaged while a parallel port peripheral is powered on and the 82091 is powered off.

6.1 Parallel Port Registers

This section is organized into three sub-sections—ISA-Compatible and PS/2-Compatible Modes, EPP Mode, and ECP Mode. Since the register sets are similar for ISA-Compatible and PS/2-Compatible modes (differing by a direction control bit in the PCON Register) the register set descriptions are combined. The EPP mode and ECP mode register sets are described separately. Each register set description contains the I/O address assignment and a complete description of the registers and register bits. Note that the PSTAT and PCON Registers are common to all modes and for completeness are repeated in each sub-section. Any difference in bit operations for a particular mode is noted in that particular register description.

The registers provide parallel port control/status information and data paths for transferring data between the parallel port interface and the 8-bit host interface. All registers are accessed as byte quantities. The base address is determined by hardware configuration at powerup (or a hard reset) or via software configuration by programming the 82091AA configuration registers as described in Section 4.0, AIP Configuration. The parallel port can be disabled or configured for a base address of 378h (all modes), 278h (all modes), or 3BCh (all modes except EPP and ECP). This provides the system designer with the option of using additional parallel ports on add-in cards that have fixed address decoding.

Some of the parallel port registers described in this section contain reserved bits. These bits are labeled "R". Software must deal correctly with fields that are reserved. On reads, software must use appropriate masks to extract the defined bits and not rely on reserved bits being any particular value. On writes, software must ensure that the values of reserved bit positions are preserved. That is, the value of reserved bit positions must first be read, merged with the new values for other bit positions, and then written back.

During a hard reset (RSTDRV asserted), the 82091AA registers are set to pre-determined **default** states. The default values are indicated in the individual register descriptions.

The following nomenclature is used for register access attributes:

RO **Read Only**. Note that for registers with read only attributes, writes to the I/O address have no affect on parallel port operations.

R/W **Read/Write**. A register with this attribute can be read and written. Note that individual bits in some read/write registers may be read only.

6.1.1 ISA-COMPATIBLE AND PS/2-COMPATIBLE MODES

This section contains the registers used in ISA-Compatible and PS/2-Compatible modes. The I/O address assignment for this register set is shown in Table 15 and the register descriptions are presented in the order that they appear in the table.

Table 15. Parallel Port Register (ISA-Compatible and PS/2-Compatible)

Parallel Port Register Address Access (AEN = 0) Base+	Abbreviation	Register Name	Access
0h	PDATA	Data Register	R/W
1h	PSTAT	Status Register	RO
2h	PCON	Control Register	R/W

NOTE:
Parallel port base addresses are 278h, 378h and 3BCh.

5

6.1.1.1 PDATA—Parallel Port Data Register (ISA-Compatible and PS/2-Compatible Modes)

I/O Address: Base +00h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

ISA-Compatible Mode

The PDATA Register is a uni-directional data port that transfers 8-bit data from the host to the peripheral device (forward transfer). A write to this register drives the written data onto PD[7:0]. Reads of this register should not be performed in ISA-Compatible mode. For a host read of this address location, the 82091AA completes the handshake on the ISA Bus and the value is the last value stored in the PDATA Register.

PS/2-Compatible Mode

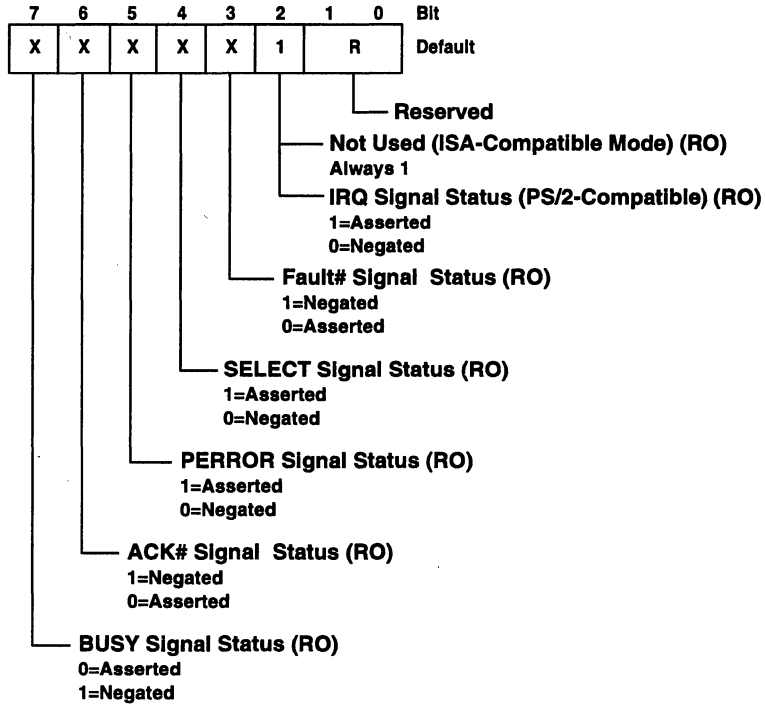
The PDATA Register is a bi-directional data port that transfers 8-bit data between the peripheral device and host. The direction of transfer is determined by the DIR# bit in the PCON Register. If DIR# = 0 (forward direction), and the host writes to this register, the data is stored in the PDATA Register and driven onto PD[7:0]. If DIR# = 1 (reverse direction), a host read of this register returns the data on PD[7:0]. Note that read data is not stored in the PDATA Register.

Bit	Description
7:0	**PARALLEL PORT DATA:** Bits[7:0] correspond to parallel port data lines PD[7:0] and ISA Bus data lines SD[7:0].

6.1.1.2 PSTAT—Status Register (ISA-Compatible and PS/2-Compatible Modes)

I/O Address: Base +01h
Default Value: XXXX X1RR
Attribute: Read Only
Size: 8 bits

The PSTAT Register provides the status of certain parallel port signals and whether a CPU interrupt has been generated by the parallel port. This register indicates the current state of the BUSY, ACK#, PERROR, SELECT, and FAULT# signals.



NOTE:
X = Default value is determined by signal state at reset.

Figure 26. Status Register (ISA-Compatible and PS/2-Compatible Modes)

5

Bit	Description
7	**BUSY STATUS (BUSYS):** This bit indicates the state of the parallel port interface BUSY signal. When BUSY is asserted, BUSYS = 0. When BUSY is negated, BUSYS = 1.This bit is an inverted version of the parallel port BUSY signal.
6	**ACK# STATUS (ACKS):** This bit indicates the state of the parallel port interface ACK# signal. This bit indicates when the peripheral has received a data byte and is ready for another. When ACK# is asserted, ACKS = 0. When ACK# is negated, ACKS = 1. Note that if interrupts are enabled (via bit 4 of the PCON Register), the assertion of the ACK# signal generates an interrupt to the CPU.
5	**PERROR STATUS (PERRS):** This bit indicates the state of the parallel port interface PERROR signal. This bit indicates when an error has occurred in the peripheral paper path (e.g., out of paper). When PERROR is asserted, PERRS = 1, When PERROR is negated, PERRS = 0.
4	**SELECT STATUS (SELS):** This bit indicates the state of the parallel port interface SELECT signal. When the SELECT signal is asserted, SELS = 1, When the SELECT signal is negated, SELS = 0.
3	**FAULT# STATUS (FAULTS):** This bit indicates the state of the parallel port interface FAULT# signal being driven by the peripheral device. When the FAULT# signal is asserted, FAULTS = 0. When the FAULT# signal is negated, FAULTS = 1.
2	**PARALLEL PORT INTERRUPT STATUS (PIRQ):** This bit indicates a CPU interrupt by the parallel port. PIRQ indicates that the printer has accepted the previous character and is ready for another. In ISA-Compatible mode, interrupt status is not reported in this register and this bit is always 1. In PS/2-Compatibile mode, if interrupts are enabled via the PCON Register and the ACK# signal is asserted (low-to-high transition), PIRQ is set to a 0 (and an IRQ generated to the CPU). The 82091AA sets PIRQ to 1 when this register is read or by a hard reset. If interrupts are disabled via the PCON Register, this bit is never set to 0.
1:0	**RESERVED**

6.1.1.3 PCON—Control Register (ISA-Compatible And PS/2-Compatible Mode)

I/O Address: Base + 02h
Default Value: RR00 0000
Attribute: Read/Write
Size: 8 bits

The PCON Register controls certain parallel port interface signals and enables/disables parallel port interrupts. This register permits software to control the STROBE#, AUTOFD#, INIT#, and SELECTIN# signals. For PS/2-Compatible mode, this register also controls the direction of transfer on PD[7:0].

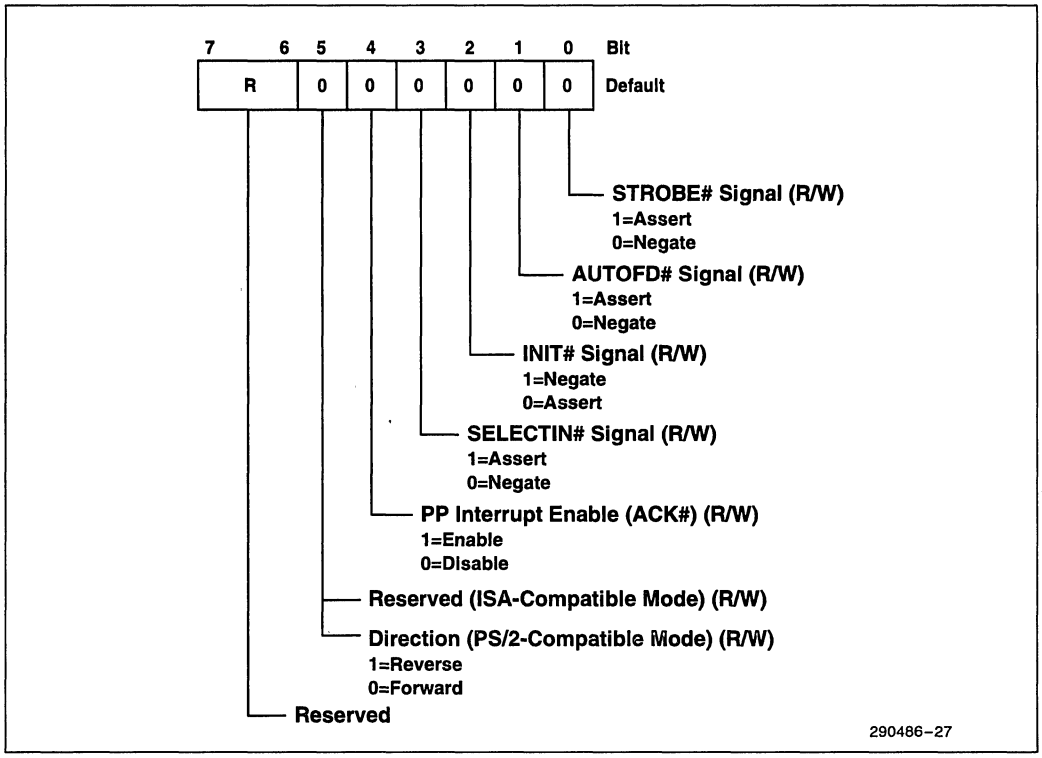


Figure 27. Control Register (ISA-Compatible and PS/2-Compatible Modes)

5

Bit	Description
7:6	**RESERVED**
5	**RESERVED (ISA-COMPATIBLE MODE):** Not used and undefined when read. Writes have no affect on parallel port operations. **DIRECTION (DIR#) (PS/2-COMPATIBLE MODE):** This bit is used to control the direction of data transfer on the parallel port data bus (PD[7:0]). When DIR# = 0, PD[7:0] are outputs. When DIR# = 1, PD[7:0] are inputs.
4	**ACK# INTERRUPT ENABLE (ACKINTEN):** ACKINTEN enables CPU interrupts (via either IRQ5 or IRQ7) to be generated when the ACK# signal on the parallel port interface is asserted. When ACKINTEN = 1, a CPU interrupt is generated when ACK# is asserted. When ACKINTEN = 0, the ACK# interrupt is disabled.
3	**SELECTIN# CONTROL (SELINC):** This bit controls the SELECTIN# signal. SELINC is set to 1 to select the printer. When SELINC = 1, the SELECTIN# signal is asserted, When SELINC = 0, the SELECTIN# signal is negated.
2	**INIT# CONTROL (INITC):** This bit controls the INIT# signal. When INITC = 1, the INIT# signal is negated. When INITC = 0, the INIT# signal is asserted.
1	**AUTOFD# CONTROL (AUTOFDC):** This bit controls the AUTOFD# signal. AUTOFDC is set to 1 to instruct the printer to advance the paper one line each time a carriage return is received. When AUTOFDC = 1, the AUTOFD# signal is asserted. When AUTOFDC = 0, the AUTOFD# signal is negated.
0	**STROBE# CONTROL (STROBEC):** This bit controls the STROBE# signal. The STROBE# signal is set active to instruct the printer to accept the character being presented on the data lines. When STROBEC = 1, the STROBE# signal is asserted. When STROBEC = 0, the STROBE# signal is negated.

6.1.2 EPP MODE

This section contains the registers used in EPP mode. The I/O address assigment for this register set is shown in Table 16 and the register descriptions are presented in the order that they appear in the table.

Table 16. Parallel Port Registers (EPP Mode)

Parallel Port Register Address Access (AEN = 0) Base +	Abbreviation	Register Name	Access
0h	PDATA	Data Register	R/W
1h	PSTAT	Status Register	RO
2h	PCON	Control Register	R/W
3h	ADDSTR	Address Strobe Register	R/W
4h–7h	DATASTR	Data Strobe Registers	R/W

NOTE:
Parallel port base addresses are 278h (LPT2) and 378h (LPT1). Base address 3BCh is not available in EPP or ECP modes.

6.1.2.1 PDATA—Parallel Port Data Register (EPP Mode)

I/O Address: Base + 00h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

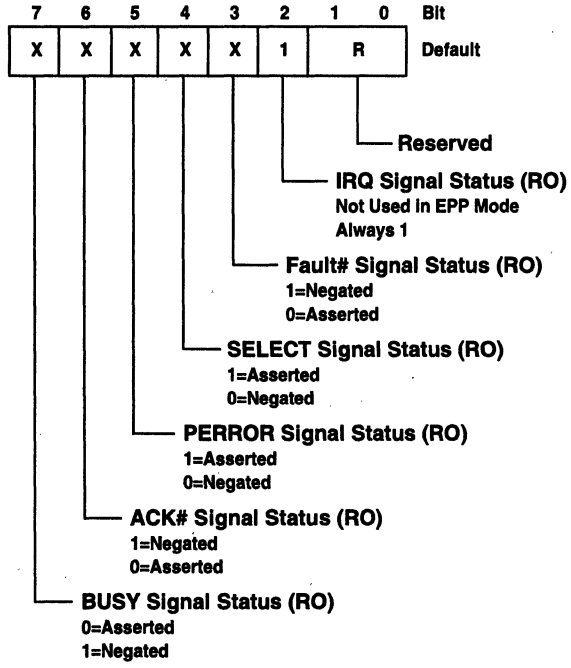
The PDATA Register is a bi-directional data port that transfers 8-bit data between the peripheral device and host. The direction of transfer is determined by the DIR# bit in the PCON Register. If DIR# = 0 (forward direction) and the host writes to this register, the data is stored in the PDATA Register and driven onto PD[7:0]. If DIR# = 1 (reverse direction), a host read of this register returns the data on PD[7:0]. However, read data is not stored in the PDATA Register.

Bit	Description
7:0	**PARALLEL PORT DATA:** Bits[7:0] correspond to parallel port data lines PD[7:0] and ISA Bus data lines.

6.1.2.2 PSTAT—Status Register (EPP Mode)

I/O Address: Base +01h
Default Value: XXXX X1RR
Attribute: Read Only
Size: 8 bits

The PSTAT Register provides the status of certain parallel port signals. It also indicates whether a CPU interrupt has been generated by the parallel port. This register indicates the current state of the BUSY, ACK#, PERROR, SELECT, and FAULT# signals.



NOTE:
X=Default value is determined by signal state at reset.

Figure 28. Status Register (EPP Mode)

Bit	Description
7	**BUSY STATUS (BUSYS):** This bit indicates the state of the parallel port interface BUSY signal. When BUSY is asserted, BUSYS = 0. When BUSY is negated, BUSYS = 1. This bit is an inverted version of the parallel port BUSY signal.
6	**ACK# STATUS (ACKS):** This bit indicates the state of the parallel port interface ACK# signal. This bit indicates when the peripheral has received a data byte and is ready for another. When ACK# is asserted, ACKS = 0. When ACK# is negated, ACKS = 1. Note that if interrupts are enabled (via bit 4 of the PCON Register), the assertion of the ACK# signal generates an interrupt to the CPU.
5	**PERROR STATUS (PERRS):** This bit indicates the state of the parallel port interface PERROR signal. This bit indicates when an error has occurred in the peripheral paper path (e.g., out of paper). When PERROR is asserted, PERRS = 1. When PERROR is negated, PERRS = 0.
4	**SELECT STATUS (SELS):** This bit indicates the state of the parallel port interface SELECT signal. When the SELECT signal is asserted, SELS = 1. When the SELECT signal is negated, SELS = 0.
3	**FAULT# STATUS (FAULTS):** This bit indicates the state of the parallel port interface FAULT# signal being driven by the peripheral device. When the FAULT# signal is asserted, FAULTS = 0. When the FAULT# signal is negated, FAULTS = 1.
2	**PARALLEL PORT INTERRUPT (PIRQ):** In EPP mode interrupt status is not reported in this register and this bit is always 1.
1:0	**RESERVED**

6.1.2.3 PCON—Control Register (EPP Mode)

I/O Address: Base + 02h
Default Value: RR00 0000
Attribute: Read/Write
Size: 8 bits

The PCON Register controls certain parallel port interface signals, enables/disables parallel port interrupts, and selects the direction of data transfer on PD[7:0]. This register permits software to control the INIT# signal. Note that in the EPP parallel interface protocol, the STROBE#, AUTOFD#, and SELECTIN# signals are automatically generated by the parallel port and are not controlled by software.

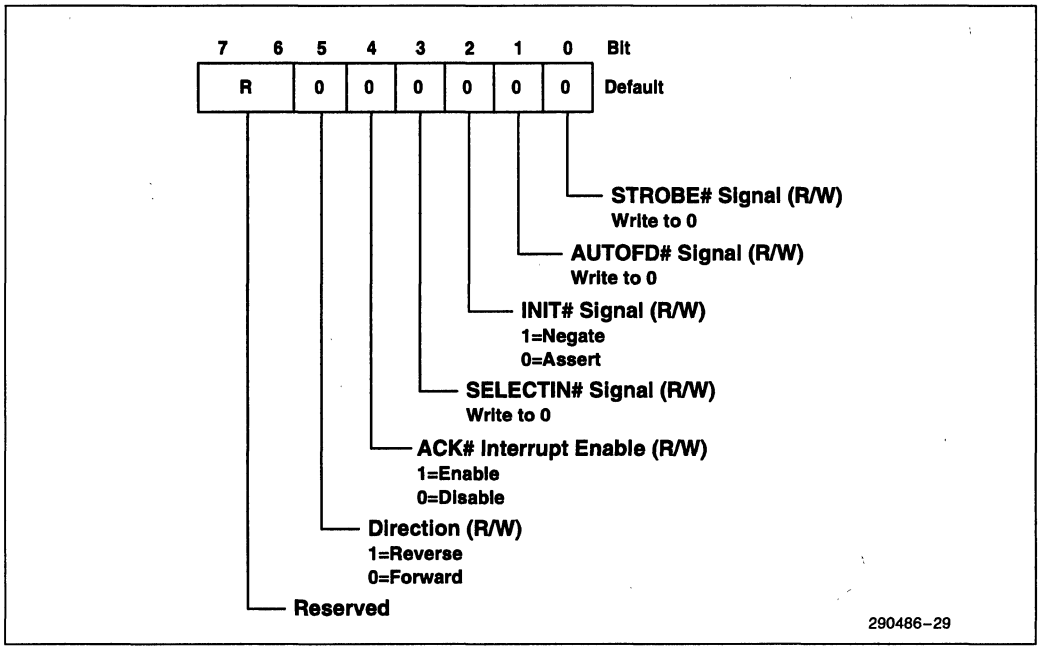


Figure 29. Control Register (EPP Mode)

Bit	Description
7:6	**RESERVED**
5	**DIRECTION (DIR#):** This bit is used to control the direction of data transfer on the parallel port data bus (PD[7:0]). When DIR# = 0 (forward direction), PD[7:0] are outputs. When DIR# = 1 (reverse direction), PD[7:0] are inputs.
4	**ACK# INTERRUPT ENABLE (ACKINTEN):** ACKINTEN enables CPU interrupts (via IRQ5 or IRQ7) to be generated when the ACK# signal on the parallel port interface is asserted. When ACKINTEN = 1, a CPU interrupt is generated when ACK# is asserted. When ACKINTEN = 0, the ACK# interrupt is disabled.
3	**SELECTIN# CONTROL (SELINC):** Write to 0 when programming this register. This bit must be 0 for the parallel port handshake to operate properly.
2	**INIT# CONTROL (INITC):** This bit controls the INIT# signal. When INITC = 1, the INIT# signal is negated. When INITC = 0, the INIT# signal is asserted.
1	**AUTOFD# CONTROL (AUTOFDC):** Write to 0 when programming this register.
0	**STROBE# CONTROL (STROBEC):** Write to 0 when programming this register. This bit must be 0 for the parallel port handshake to operate properly.

6.1.2.4 ADDSTR—EPP Auto Address Strobe Register (EPP Mode)

I/O Address: Base + 03h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

The ADDSTR Register provides a peripheral address to the peripheral (via PD[7:0]) during a host address write operation and to the host (via PD[7:0]) during a host address read operation. An automatic address strobe is generated on the parallel port interface when data is read from or written to this register.

Bit	Description
7:0	**EPP ADDRESS:** Bits[7:0] correspond to SD[7:0] and PD[7:0].

6.1.2.5 DATASTR—Auto Data Strobe Register (EPP Mode)

I/O Address: Base +04h, 05h, 06h, 07h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

The DATASTR Register provides data from the host to the peripheral device (via PD[7:0]) during host write operations and from the peripheral device to the host (via PD[7:0]) during a host read operation. An automatic data strobe is generated on the parallel port interface when data is read from or written to this register. To maintain compatibility with Intel's 82360SL I/O device that has a 32-bit Host Bus interface, four consecutive byte address locations are provided for transferring data.

Bit	Description
7:0	**EPP DATA:** Bits[7:0] correspond to SD[7:0] and PD[7:0].

6.1.3 ECP MODE

This section contains the registers used in ECP mode. The I/O address assignment for this register set is shown in Table 17 and the register descriptions are presented in the order that they appear in the table. The Extended Control Register (ECR) permits various modes of operation. Note that ECR[7:5] = 000 selects ISA-Compatible mode and ECR[7:5] = 001 selects PS/2-Compatibile mode. These modes are discussed in Section 6.1.1, ISA-Compatible and PS/2 Compatible modes. The other modes selected by ECR[7:5] are discussed in this section.

Table 17. Parallel Port Registers (ECP Mode)

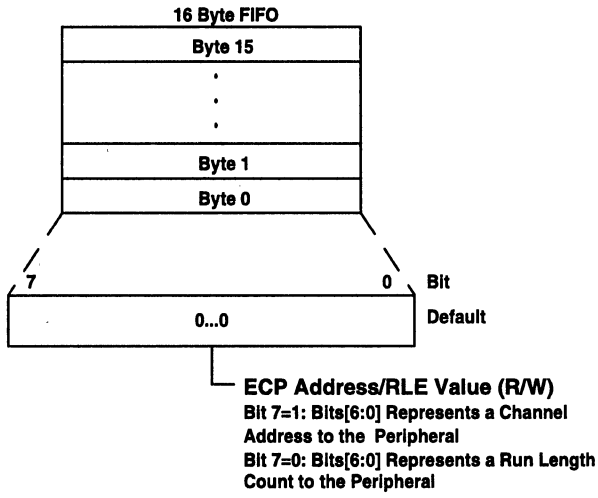
Parallel Port Register Address Access (AEN = 0) Base +	Abbreviation	Register Name	Access	
			ECR[7:5]	Read/Write Attribute
0h	ECPAFIFO	ECP Address/RLE FIFO	011	R/W
1h	PSTAT	Status Register	All	RO
2h	PCON	Control Register	All	R/W
400h	SDFIFO	Standard Parallel Port Data FIFO	010	R/W
400h	ECPDFIFO	ECP Data FIFO	011	R/W
400h	TFIFO	Test FIFO	110	R/W
400h	ECPCFGA	ECP Configuration A	111	R/W
401h	ECPCFGB	ECP Configuration B	111	R/W
402h	ECR	Extended Control Register	All	R/W

NOTES:
1. Parallel port base addresses are 278h, 378h, and 3BCh.
2. A register is accessible when the ECR[7:5] field contains the value specified in the ECR[7:5] column. The register is not accessible if the ECR[7:5] field does not match the value specified in this column. The term "All" means that the register is accessible in all modes selected by ECR[7:5].

6.1.3.1 ECPAFIFO—ECP Address/RLE FIFO Register (ECP Mode)

I/O Address: Base +00h
Default Value: UUUU UUUU (Undefined)
Attribute: Read/Write
Size: 8 bits

The ECPAFIFO Register provides a channel address or a Run Length Count (RLE) to the peripheral, depending on the state of bit 7. This I/O address location is only used in ECP mode (ECR bits[7:5] = 011). In this mode, bytes written to this register are placed in the parallel port FIFO and transmitted over PD[7:0] using ECP protocol.



NOTE:
U = Undefined

Figure 30. ECP Address/RLE FIFO Register (ECP Mode)

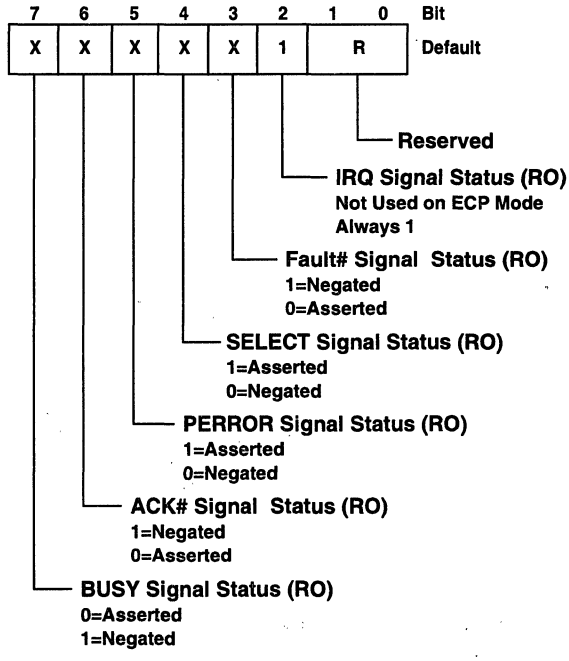
Bit	Description
7:0	**ECP ADDRESS/RLE VALUE:** Bits[7:0] correspond to parallel port data lines PD[7:0] and ISA Bus data lines SD[7:0]. The peripheral device should interpret bits[6:0] as a channel address when bit 7 = 1 and as a run length count when bit 7 = 0. Note that this interpretation is performed by the peripheral device and the value of bit 7 has no affect on 82091AA operations. Note that the 82091AA asserts AUTOFD# to indicate that the information on PD[7:0] represents an ECP address/RLE count. The 82091AA negates AUTOFD# (drives high) when PD[7:0] is transferring data.

5

6.1.3.2 PSTAT—Status Register (ECP Mode)

I/O Address: Base +01h
Default Value: XXXX X1RR
Attribute: Read Only
Size: 8 bits

The PSTAT Register provides the status of certain parallel port signals and whether a CPU interrupt has been generated by the parallel port. This register indicates the current state of the BUSY, ACK#, PERROR, SELECT, and FAULT# signals.



NOTE:
X = Default value is determined by the state of the corresponding signal pin at reset.

Figure 31. Status Register (ECP Mode)

Bit	Description
7	**BUSY STATUS (BUSYS):** This bit indicates the state of the parallel port interface BUSY signal. When BUSY is asserted, BUSYS = 0. When BUSY is negated, BUSYS = 1. This is an inverted version of the parallel port BUSY signal. Refer to Section 6.2.3 ECP Mode for more detail.
6	**ACK# STATUS (ACKS):** This bit indicates the state of the parallel port interface ACK# signal. This bit indicates when the peripheral has received a data byte and is ready for another. When ACK# is asserted, ACKS = 0. When ACK# is negated, ACKS = 1. Note that if interrupts are enabled (via bit 4 of the PCON Register), the assertion of the ACK# signal generates an interrupt to the CPU. Refer to Section 6.2.3 ECP Mode for more detail.
5	**PERROR STATUS (PERRS):** This bit indicates the state of the parallel port interface PERROR signal. This bit indicates when an error has occurred in the peripheral paper path (e.g., out of paper). When PERROR is asserted, PERRS = 1, When PERROR is negated, PERRS = 0.
4	**SELECT STATUS (SELS):** This bit is used in all parallel port modes and indicates the state of the parallel port interface SELECT signal. When the SELECT signal is asserted, SELS = 1. When the SELECT signal is negated, SELS = 0.
3	**FAULT# STATUS (FAULTS):** This bit is used in all parallel port modes and indicates the state of the parallel port interface FAULT# signal being driven by the peripheral device. When the FAULT# signal is asserted, FAULTS = 0. When the FAULT# signal is negated, FAULTS = 1.
2	**PARALLEL PORT INTERRUPT (PIRQ):** In ECP mode, interrupt status is not reported in this register and this bit is always 1.
1:0	**RESERVED**

6.1.3.3 PCON—Control Register (ECP Mode)

I/O Address: Base + 02h
Default Value: RR00 0000
Attribute: Read/Write
Size: 8 bits

The PCON Register controls certain parallel port interface signals, enables/disables parallel port interrupts, and selects the direction of data transfer on PD[7:0]. Note that the function of some bits depends on the programming of the ECR.

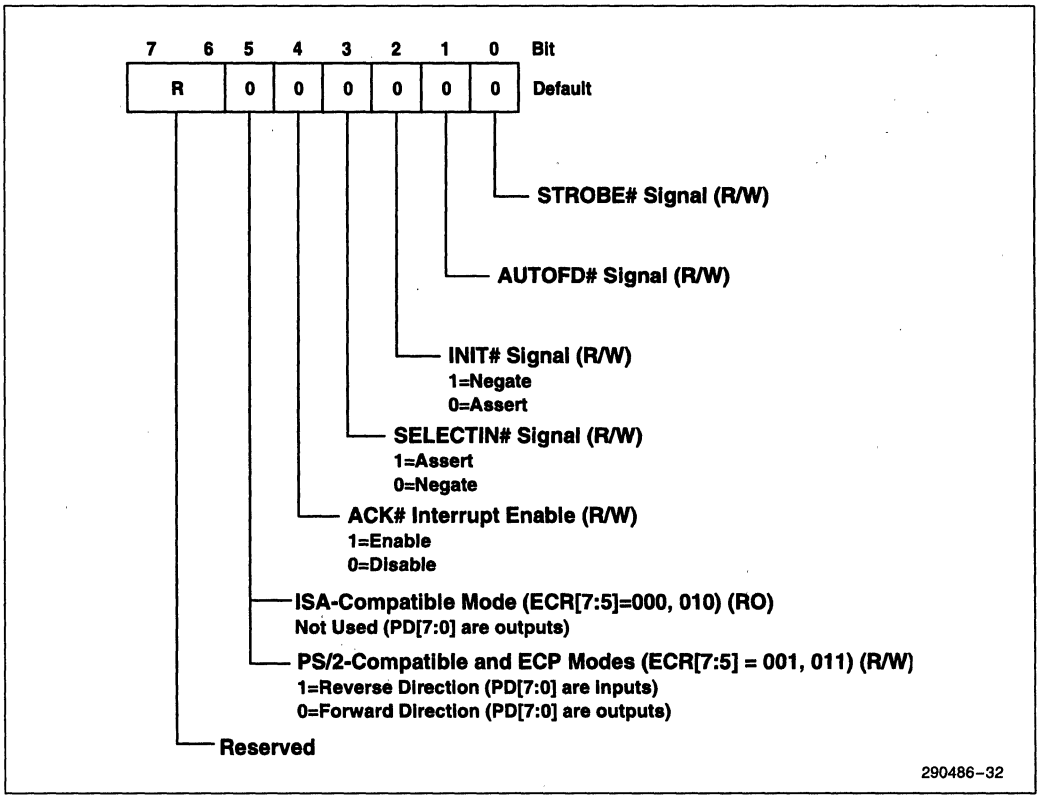


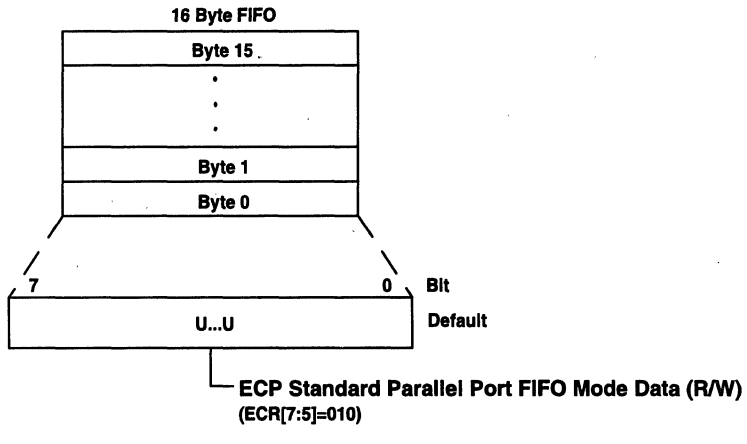
Figure 32. Control Register (ECP Mode)

Bit	Description
7:6	**RESERVED**
5	**DIRECTION (DIR#):** This bit is used to control the direction of data transfer on the parallel port data bus (PD[7:0]). When DIR# = 0 (forward direction), PD[7:0] are outputs. When DIR# = 1 (reverse direction), PD[7:0] are inputs.
4	**INTERRUPT ENABLE (ACK#) (IRQEN):** IRQEN enables interrupts to the CPU to be generated when the ACK# signal on the parallel port interface is asserted and is used in all parallel port interface modes. When IRQEN = 1, a CPU interrupt is generated when ACK# is asserted. When IRQEN = 0, parallel port interrupts are disabled.
3	**SELECTIN# CONTROL (SELINC):** This bit controls the SELECTIN# signal. SELINC is set to 1 to select the printer. When SELINC = 1, the SELECTIN# signal is asserted, When SELINC = 0, the SELECTIN# signal is negated.
2	**INIT# CONTROL (INITC):** This bit controls the INIT# signal When INITC = 1, the INIT# signal is negated. When INITC = 0, the INIT# signal is asserted.
1	**AUTOFD# CONTROL (AUTOFDC):** In ECP mode or ISA-Compatible FIFO mode (ECR[7:5] = 011, 010), this bit has no effect. Refer to Section 6.2.3 ECP Mode for more details.
0	**STROBE# CONTROL (STROBEC):** In ECP mode or ISA-Compatible FIFO mode (ECR[7:5] = 011, 010), this bit has no effect. Refer to Section 6.2.3 ECP Mode for more details.

6.1.3.4 SDFIFO—Standard Parallel Port Data FIFO

I/O Address: Base + 400h and (ECR[7:5] = 010)
Default Value: UUUU UUUU (undefined)
Attribute: Read/Write
Size: 8 bits

SDFIFO is used to transfer data from the host to the peripheral when the ECR Register is set for ISA-Compatible FIFO mode (bits[7:5] = 010). Data bytes written or DMAed from the system to this FIFO are transmitted by a hardware handshake to the peripheral using the standard ISA-Compatible protocol. Note that bit 5 in the PCON Register must be set to 0 for a forward transfer direction.



NOTE:
U = Undefined

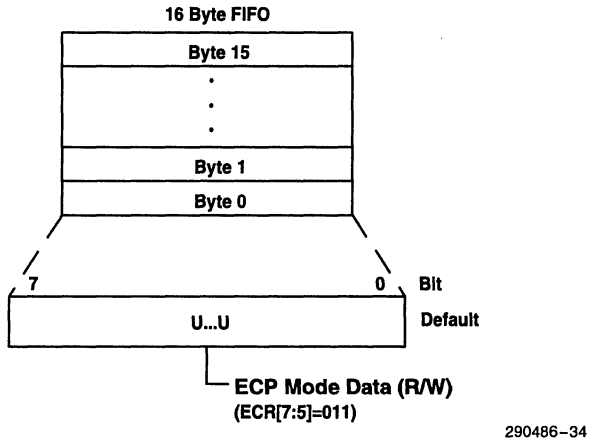
Figure 33. ECP ISA-Compatible Data FIFO

Bit	Description
7:0	**ECP STANDARD PARALLEL PORT DATA:** Bits[7:0] correspond to SD[7:0] and PD[7:0].

6.1.3.5 DFIFO—Data FIFO (ECP Mode)

I/O Address: Base +400h and (ECR[7:5]=011)
Default Value: UUUU UUUU (undefined)
Attribute: Read/Write
Size: 8 bits

This I/O address location transfers data between the host and peripheral device when the parallel port is in ECP mode (ECR Bits[7:5]=011). Transfers use the parallel port FIFO. Data is transferred on PD[7:0] via hardware handshakes on the parallel port interface using ECP parallel port interface handshake protocol.



NOTE:
U=Undefined

Figure 34. ECP Data FIFO (ECP Mode)

Bit	Description
7:0	**ECP MODE DATA:** Data bytes written or DMAed from the system to this FIFO in the forward direction (PCON bit 5=0) are transmitted to the peripheral by an ECP mode protocol hardware handshake. In the reverse direction (PCON bit 5=1) data bytes from the peripheral are transferred to the FIFO using the ECP mode protocol hardware handshake. Reads or DMAs from the FIFO return bytes of ECP data to the system. Bits[7:0] correspond to SD[7:0] and PD[7:0].

5

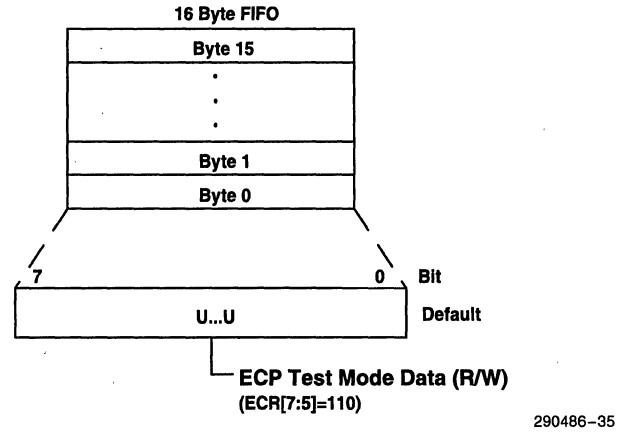
6.1.3.6 TFIFO—ECP Test FIFO Register (ECP Mode)

I/O Address: Base +400 and (ECR[7:5] = 110)
Default Value: UUUU UUUU (undefined)
Attribute: Read/Write
Size: 8 bits

The TFIFO Register provides a test mechanism for the ECP mode FIFO. Test mode is enabled via the ECR Register. In test mode (ECR[7:5] = 110), data can be read, written or DMAed to/from the FIFO by accessing this register I/O address.

Data bytes may be read, written or DMAed to or from the system to this FIFO in any direction. The parallel port interface signals are not affected by TFIFO accesses and TFIFO data is not transmitted to PD[7:0]. The test FIFO does not stall when overwritten or underrun. Data is simply re-written or over-run. The full and the empty bits in the ECR always keep track of the correct FIFO state.

The test FIFO transfers data at the maximum ISA rate so that software can generate performance metrics. The FIFO write threshold can be determined by starting with a full TFIFO and emptying it a byte at a time until a service interrupt is set to 1 in the ECR. The FIFO read threshold can be determined by setting the direction bit in the PCON Register to 1, and filling the FIFO a byte at a time until the service interrupt is set to 1 in the ECR.



NOTE:
U = Undefined

Figure 35. ECP Test FIFO Register (ECP Mode)

Bit	Description
7:0	**ECP TEST FIFO** Data: Bits[7:0] correspond to SD[7:0].

6.1.3.7 ECPCFGA—ECP Configuration A Register (ECP Mode)

I/O Address: Base +400h and (ECR[7:5] = 111)
Default Value: 1001 RRRR
Attribute: Read/Write
Size: 8 bits

The ECPCFGA Register provides information about the ECP mode implementation. Access to this register is enabled by programming the ECR Register (ECR[7:5] = 111).

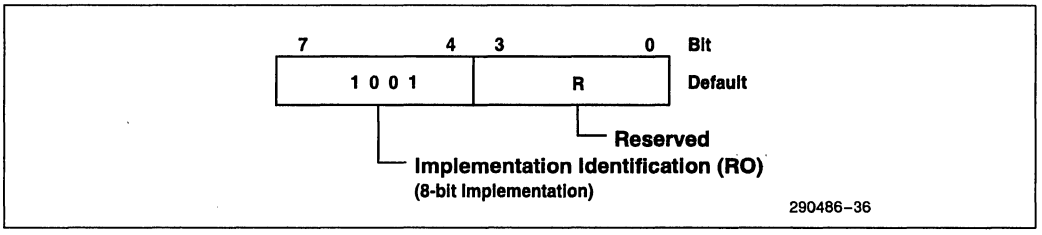


Figure 36. ECP Configuration A Register (ECP Mode)

Bit	Description
7:4	**IMPLEMENTATION IDENTIFICATION (IMPID):** This field is hardwired to 1001 to indicate an 8-bit implementation (bit 4) and an ISA-style interrupt (bit 7). This field is read only and writes have no affect.
3:0	**RESERVED**

5

6.1.3.8 ECPCFGB—ECP Configuration B Register (ECP Mode)

I/O Address: Base +401h and (ECR[7:5] = 111)
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

The ECPCFGB Register is part of the ECP specification and is implemented in the 82091AA as a scratchpad register. Software can use the fields in this register to maintain system information. Programming these bits does not affect parallel port operations. Access to the ECPCFGB Register is enabled by programming the ECR Register (ECR[7:5] = 111).

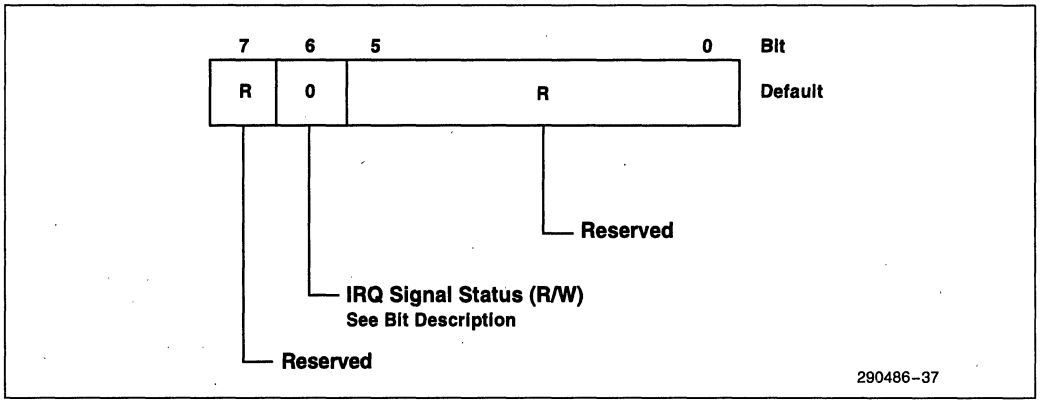


Figure 37. ECP Extended Control Register (ECP Mode)

Bit	Description
7	**RESERVED:** This bit always reads back as 0.
6	**INTRVALUE (INTRV):** This bit returns the value on the ISA IRQ line (IRQ5/IRQ7) to determine possible conflicts. The value of either IRQ5 or IRQ7 is read back based on the parallel port interrupt selection in the 82091AA configuration registers. IRQ5/IRQ7 are tri-stated in ECP configuration mode (ECR[7:5] = 111] to allow the state of the selected parallel port interrupt line to be read back. Note that the ACKINTEN bit in the PCON register must be written to 0 before the interrupt status can be read on this bit.
5:0	**RESERVED:** These bits always read back as 0.

6.1.3.9 ECR ECP—Extended Control Register (ECP Mode)

I/O Address: Base +402h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

This register selects the ECP mode, enables service and error interrupts and provides interrupt status. The ECR also enables/disables DMA operations and provides FIFO empty and FIFO full status. The FIFO empty and FIFO full status bits are also used to report FIFO overrun and underrun conditions.

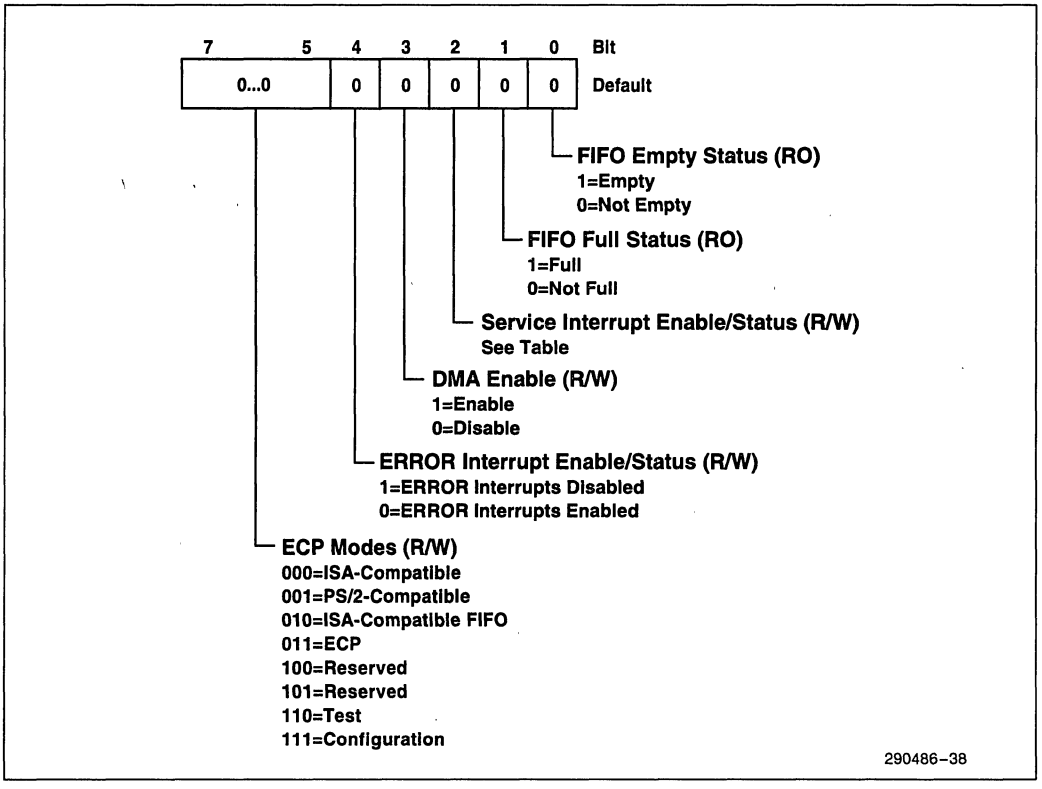


Figure 38. ECP Extended Control Register (ECP Mode)

5

Bit	Description
7:5	**ECP MODE SELECT:** This field selects one of the following ECP Modes: **Mode** **Operation** **0 0 0** **ISA-Compatible Mode.** In this mode the parallel port operates in ISA-Compatible mode. The FIFO is reset and common collector drivers are used on the control lines (STROBE#, AUTOFD#, INIT# and SELECTIN#). Setting the direction bit to 1 in the PCON Register does not affect the parallel port interface. For register descriptions in this mode, See Section 6.1.1, ISA-Compatible and PS/2-Compatible Modes. **0 0 1** **PS/2-Compatible Mode.** In this mode the parallel port operates in PS/2-Compatible mode. The FIFO is reset and common collector drivers are used on the control lines (STROBE#, AUTOFD#, INIT# and SELECTIN#). Unlike mode 000 above, setting the direction bit to 1 in the PCON Register tri-states the data lines and reading the data register returns the value on the PD[7:0]. For register descriptions in this mode, see Section 6.1.1, ISA-Compatible and PS/2-Compatible Modes. **0 1 0** **ISA-Compatible FIFO Mode.** This mode is the same as mode 000 above, except that data is written or DMAed to the FIFO. FIFO data is automatically transmitted using the ISA-style protocol. For this mode, the direction control bit in the PCON register must be 0. **0 1 1** **ECP Mode.** In the forward direction, bytes written to the ECP DFIFO location and bytes written to the ECP AFIFO location are placed in the ECP FIFO and transmitted automatically to the peripheral using ECP protocol. In reverse direction bytes are transferred from PD[7:0] to the ECP FIFO. **1 0 0** **Reserved** **1 0 1** **Reserved** **1 1 0** **Test Mode.** In this mode, the FIFO may be written and read, but the data will not be transmitted on PD[7:0]. **1 1 1** **Configuration Mode.** In this mode, the ECP Configuration A and B Registers are accessible. ECP Mode Switching Guidelines Software will execute P1284 negotiations and all operation prior to a data transfer phase under programmed I/O (using mode 000 or 001). Hardware provides an automatic control line handshake, moving data between the FIFO and the ECP port only in the data transfer phase (using modes 011 or 010). Setting the mode to 011 or 010 causes the hardware to initiate the data transfer. If the parallel port is in mode 000 or 001, the port can be switched to any other mode. If the parallel port is not in mode 000 or 001, the port can only be switched into mode 000 or 001. The direction and the FIFO threshold can only be changed in modes 000 or 001. Note that the FIFO, FIFO Error, and TC conditions are also reset when the mode is switched to 000 or 001. Once in an extended forward mode, the software should wait for the FIFO to be empty before switching back to mode 000 or 001. In this case, all control signals are negated before the mode switch. In an ECP reverse mode the software waits for all the data to be read from the FIFO before changing to mode 000 or 001.

Bit	Description
4	**ERROR INTERRUPT DISABLE (ERRINTREN):** This bit enables error interrupts to the host. In ECP Mode, When ERRINTREN = 1, interrupts are disabled. When ERRINTREN = 0, interrupts are enabled. When enabled and a high-to-low transition occurs on the FAULT# signal (FAULT# asserted), an interrupt is generated to the host. Note that if this bit is written from a 1 to a 0 while FAULT# is asserted, an interrupt is generated to the host.
3	**DMA ENABLE (DMAEN):** This bit enables/disables DMA. When DMAEN = 1, DMA is enabled and the host uses PPDREQ, PPDACK, and TC to transfer data. When DMAEN = 0, DMA is disabled and the PPDREQ output is tri-stated. In this case, programmed I/O is used to transfer data between the host and the 82091AA FIFO. The Service Interrupt (bit 2) needs to be set to 0 to allow generation of a TC interrupt. This bit must be written to 0 to reset the TC interrupt.
2	**SERVICE INTERRUPT (SERVICEINTR):** This bit enables FIFO and TC service interrupts. When the CPU writes SERVICEINTR = 1, FIFO request interrupts, FIFO error interrupts, and TC interrupts are disabled. Setting this bit to a 0 enables interrupts for one of the four cases listed below. When enabled (set to 0) and one of the four conditions below occurs, the 82091AA sets SERVICEINTR to a 1 and generates an interrupt to the host. 1. During DMA operations (DMAEN = 1), when terminal count is reached (TC asserted). To clear the TC interrupt, switch to ISA-Compatible or PS/2-Compatible mode (write ECR[7:5] to 000, 001) or set DMAEN to 0. 2. In the forward direction and DMAEN = 0, when there is a threshold number of bytes in the FIFO to be written. 3. In the reverse direction and DMAEN = 0, when there is a threshold number of bytes in the FIFO to be read. 4. In either DMA or programmed I/O mode when there is a FIFO overrun or underrun. Reading the SERVICEINTR bit indicates the presence of an active interrupt when this bit has been written to a 0 prior to reading it back. To disable interrupts, the SERVICEINTR bit must be explicitly written to a 1. **NOTE:** The ACK# and FAULT# interrupts can be generated independent of the value of the SERVICEINTR bit. ACK# and FAULT# interrupts are enabled via the ACKINTREN bit in the PCON Register and the ERRINTREN bit in the ECR Registers, respectively. The parallel port IRQ output (IRQ5/IRQ7) is enabled when ACKINTREN = 1 in the PCON Register or when ECR[7:5] = 010, 011, or 110. Otherwise, the IRQ output is tri-stated.
1	**FIFO FULL STATUS (FIFOFS):** This bit indicates when the FIFO is full. When FIFOFS = 1 (and FIFOES = 0), the FIFO is full and cannot accept another byte of data. When FIFOFS = 0, at least one byte location is free in the FIFO. This bit is read only and writes have no affect. When a FIFO overrun or underrun occurs, the 82091AA sets both FIFOES and FIFOFS to 1. To clear the FIFO error condition interrupt, swiitch the parallel port mode from ECP (011) to either ISA-Compatible or PS/2-Compatible modes (000 or 001).
0	**FIFO EMPTY STATUS (FIFOES):** This bit indicates when the FIFO is empty. When FIFOES = 1 (and FIFOFS = 0), the FIFO is empty. When FIFOES = 0, the FIFO contains at least one byte. This bit is read only and writes have no affect. When a FIFO overrun or underrun occurs, the 82091AA sets both FIFOES and FIFOFS to 1. To clear the FIFO error condition interrupt, swiitch the parallel port mode from ECP (011) to either ISA-Compatible or PS/2-Compatible modes (000 or 001).

6.2 Parallel Port Operations

The parallel port can be placed in ISA-Compatible, PS/2-Compatible, or EPP mode by hardware configuration or by writing to the 82091AA's configuration registers (PCFG1 Register). If access to the configuration registers is not disabled by hardware configuration, a hardware configured parallel port mode can be changed by programming the PCFG1 Register.

ECP mode is entered by programming the ECP Extended Control Register (ECR). Writing to this register changes any previously selected parallel port mode (via hardware configuration or writing the PCFG1 Register) to one of the ECP ECR Register modes selected via ECR[7:5]. Note that ECP mode cannot be entered by hardware configuration or programming the 82091AA configuration registers.

6.2.1 ISA-COMPATIBLE AND PS/2-COMPATIBLE MODES

The ISA-Compatible mode is used for standard ISA-type parallel port interfaces. Figure 39 shows the parallel port interface for ISA-Compatible mode. STROBE#, AUTOFD#, INIT#, and SELECTIN# are controlled by software via the PCON Register and the status of SELECT, PERROR, FAULT, ACK#, and BUSY are reported in the PSTAT Register. PD[7:0] are outputs only. Note that for a reverse data transfer using the Nibble protocol, the peripheral device supplies data, 4 bits at a time, using the BUSY, SELECT, PERROR, and FAULT# signals.

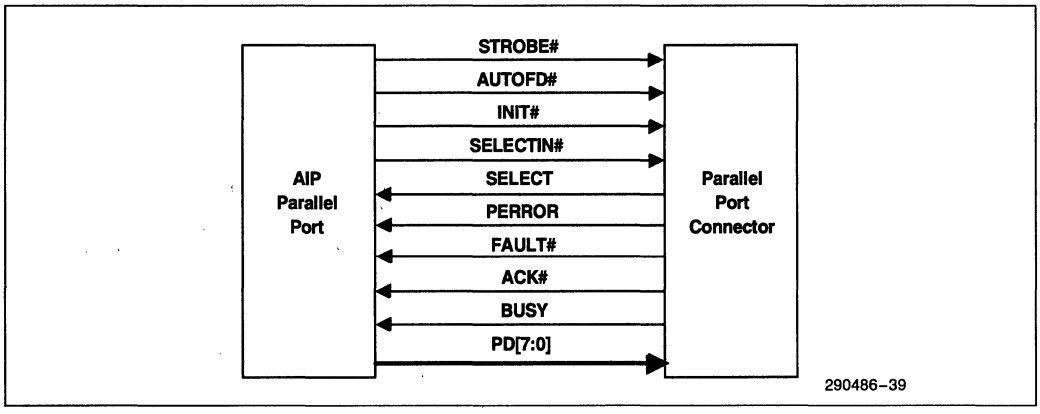


Figure 39. ISA-Compatible Mode

The following general protocol is used when communicating with a printer or other parallel port device.

Software selects the peripheral device by asserting the SELECTIN# signal. The peripheral device, in turn, acknowledges that it is selected by asserting the SELECT signal. The INIT# signal is used to initialize the peripheral device. If an error is encountered during initialization or normal operations, the peripheral device asserts FAULT#. If a printer (or plotter) encounters problems in the paper path, the device asserts PERROR. Other peripheral devices may not use the PERROR signal.

During normal operation, the peripheral device asserts BUSY when it is not ready to receive data. When it has finished processing the previous data, the peripheral device asserts ACK# and negates BUSY. If interrupts are enabled, a low-to-high transition on ACK# when the signal is negated generates an interrupt. If interrupts are not enabled, software must poll the PSTAT Register to determine when ACK# is pulsed.

The parallel port driver software sends data to the peripheral device by writing to the PDATA Register and asserting the STROBE# signal after an appropriate data stabilization interval. After a sufficient setup time has elapsed, software then negates STROBE#. Valid data is read by the peripheral device.

In the PS/2-Compatible mode, data can be written to or read from the parallel port. Figure 40 shows the parallel port interface for PS/2-Compatible mode. The Byte protocol signal names are shown in parenthesis. Before reading or writing the PDATA Register, the direction control bit in the PCON Register must be set to the proper transfer direction on PD[7:0]. During a write to the PDATA Register (with DIR# = 0), data is latched by the PDATA Register and driven onto PD[7:0]. During a parallel port read of the PDATA Register (with DIR# = 1), the data on PD[7:0] is driven onto SD[7:0]. The data is not latched by the PDATA Register during the read cycle.

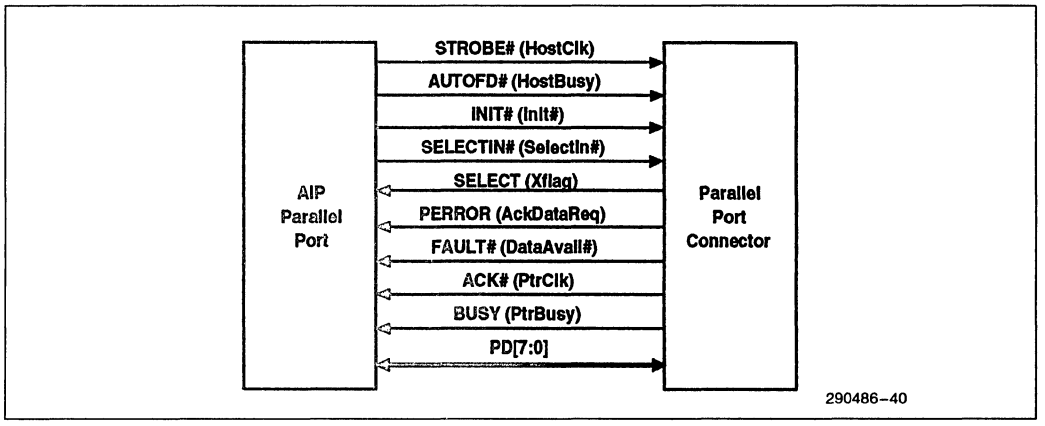


Figure 40. PS/2-Compatible Mode

6.2.2 EPP MODE

The 82091AA is EPP 1.7 compliant. This means EPP cycles will begin with WAIT (Busy) inactive, however, WAIT will still prolong the cycle when active. Figure 41 shows the parallel port interface for EPP mode. The EPP parallel port interface protocol signal names are shown in parenthesis. In EPP mode, the initialization, printer selection, and error signals (PERROR and FAULT#) work the same way as in the ISA-Compatible mode. However, in EPP mode, SELECTIN# and AUTOFD# are automatically generated and become Address Strobe (AStrb#) and Data Strobe (DStrb#), respectively, enabling direct access to parallel port devices. STROBE (Write#) is used to indicate a read or write cycle. Note that BUSY (Wait) is an active low signal in EPP mode rather than an active high signal as in ISA-Compatible mode. In addition, BUSY, in combination with IOCHRDY on the ISA Bus extends clock cycles to enable the completion of a read or write without additional wait states. EPP write and read cycles are shown in Figure 42 and Figure 43.

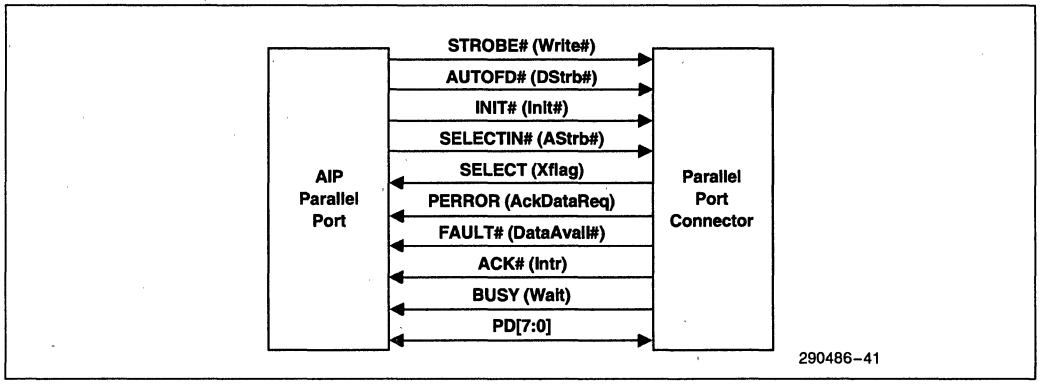


Figure 41. EPP Mode

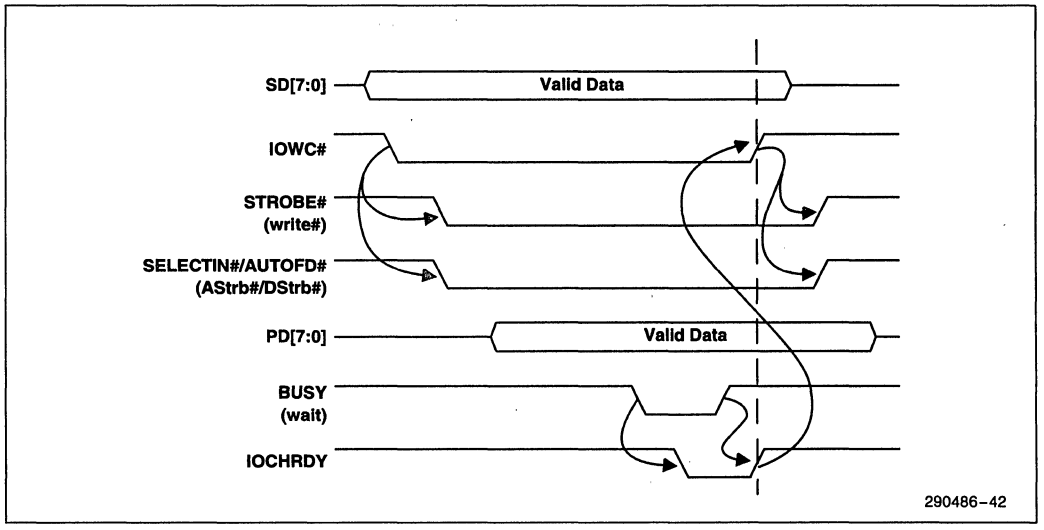


Figure 42. EPP Mode Write Cycle

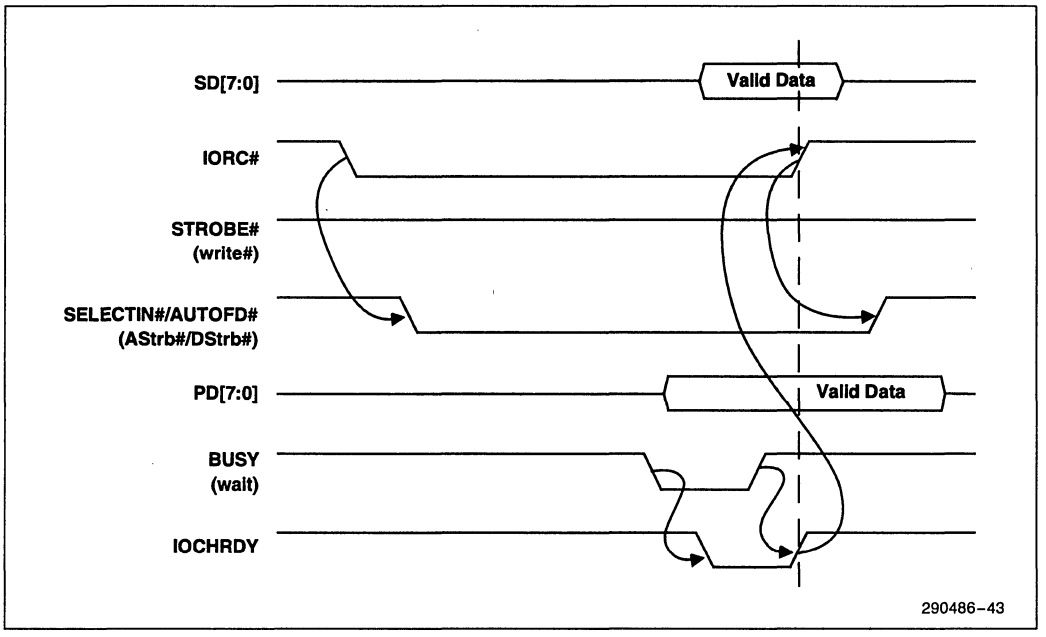


Figure 43. EPP Mode Read Cycle

5

6.2.3 ECP MODE

Figure 44 shows the parallel port interface for ECP mode with the ECP protocol signal names in parenthesis. The ECP modes are selected by programming the Extended Control Register (ECR bits[7:5]). Two of the modes (Test and Configuration) provide information about the 82091AA's parallel port and are not used for interfacing with a peripheral device. Four peripheral interface modes are selectable via the ECR—ISA-Compatible mode, ISA-Compatible FIFO mode, PS/2-Compatible mode, and ECP mode.

ISA-Compatible and PS/2-Compatible Modes (ECR[7:5] = 000,001)

The ISA-Compatible and PS/2-Compatible mode selections in the ECR are equivalent to selecting these modes via hardware configuration or programming the PCFG1 Register. For these modes the operation is the same as described in Section 6.2.1, ISA-Compatible and PS/2-Compatible Modes.

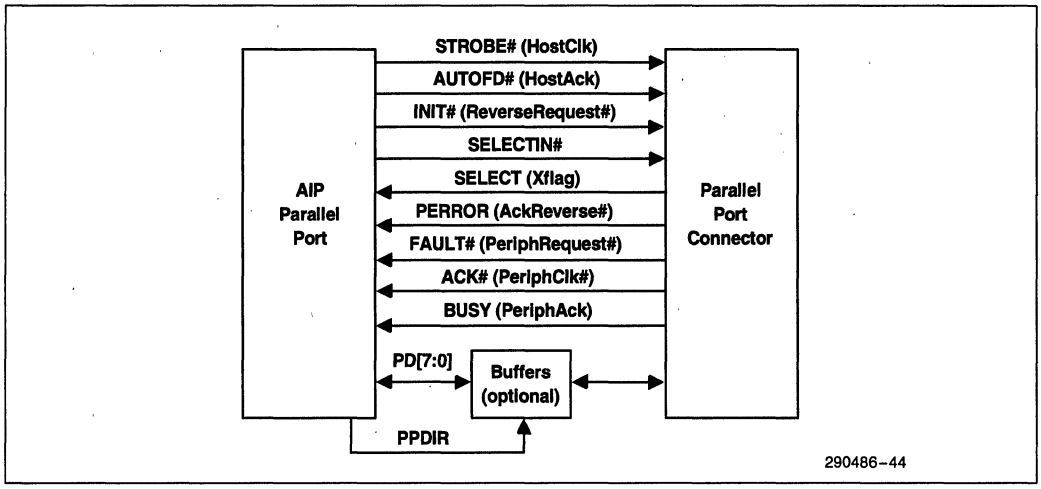


Figure 44. ECP Mode

ISA-Compatible FIFO Mode (ECR[7:5] = 010)

The ISA-Compatible FIFO mode uses the same signaling protocol on the parallel port interface as the ISA-Compatible mode. However, there are two major operational differences. First, data is written to a 16-byte FIFO (via the SDFIFO address location). The FIFO empty and FIFO full bits in the ECR provide FIFO status. In addition, DMA can be used to transfer data to the FIFO by enabling this feature in the ECR.

Second, the data is transferred to the peripheral using an automatic hardware handshake. This handshake emulates the standard ISA-Compatible style software generated handshake (Figure 45). For ISA-Compatible FIFO mode, the 82091AA does not monitor the ACK# signal. Service interrupts are enabled and reported via the ECR. The generation of service interrupts is based on the state of the FIFO and not individual transfers (using ACK#) as is the case in standard ISA-Compatible mode.

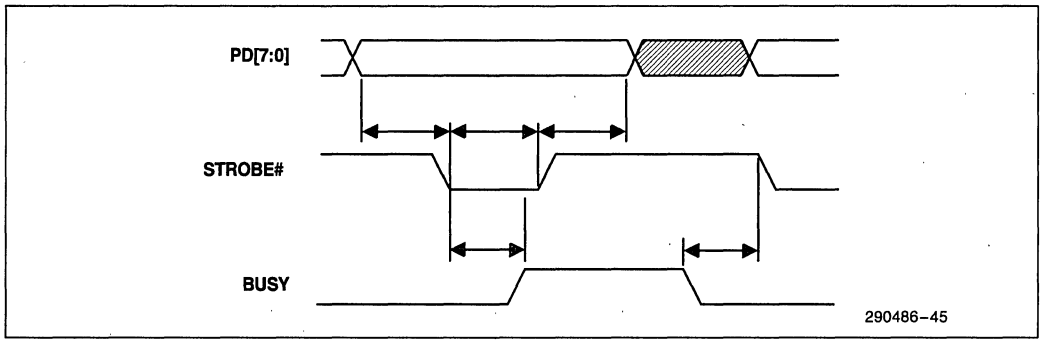


Figure 45. ISA-Compatible Timing

5

ECP Mode (ECR[7:5] = 011)

When ECR[7:5] = 011, the parallel port operates in ECP mode. In ECP mode, both data and commands (addresses and RLE) are transferred using the parallel port 16-byte FIFO. This information can be either written to or read from the FIFO using DMA or non-DMA ISA Bus transfers. The parallel port interface transfers use an automatic handshake generated by the 82091AA. The host controls the transfer direction by programming the DIR# bit in the PCON Register.

When the host is writing to the peripheral device (forward direction), STROBE#, and BUSY provide the automatic handshake for transfer on the parallel port interface (Figure 46). The peripheral device negates BUSY when it is ready to receive data or commands. AUTOFD# indicates whether PD[7:0] contain data (AUTOFD# is high) or a command (AUTOFD# is low). For commands (address or RLE), the host writes to the ECPAFIFO Register I/O address and for data, the host writes to the DFIFO Register I/O address. The addresses and data are placed in the same 16-byte FIFO. When the FIFO is full and cannot accept more data/addresses, the FIFO Full status bit is set in the ECR.

Data/addresses written to the FIFO are transferred to the peripheral device via PD[7:0]. To begin a transfer on the peripheral interface, the 82091AA checks BUSY to make sure the peripheral is in the ready state. If BUSY is negated, the 82091AA drives PD[7:0] and AUTOFD#, and asserts STROBE# to indicate that the data/command is on PD[7:0]. The peripheral device asserts BUSY to indicate that it is receiving the data/command. BUSY asserted causes the 82091AA to negate STROBE#.

When the host is reading from the peripheral device (reverse direction), AUTOFD# and ACK# provide the automatic handshake for transfer on the parallel port interface (Figure 47). Data/commands from the peripheral device are placed in the parallel port FIFO using this handshake. In this case, BUSY indicates whether PD[7:0] contain data (BUSY is high) or a command (BUSY is low).

The peripheral device asserts ACK# to indicate that a data/command is on PD[7:0]. The 82091AA negates AUTOFD# when it is ready for a peripheral transfer and asserts AUTOFD# to indicate that it is receiving the data/command. AUTOFD# asserted causes the peripheral device to negate ACK#. The peripheral transfers are to the parallel port 16-byte FIFO.

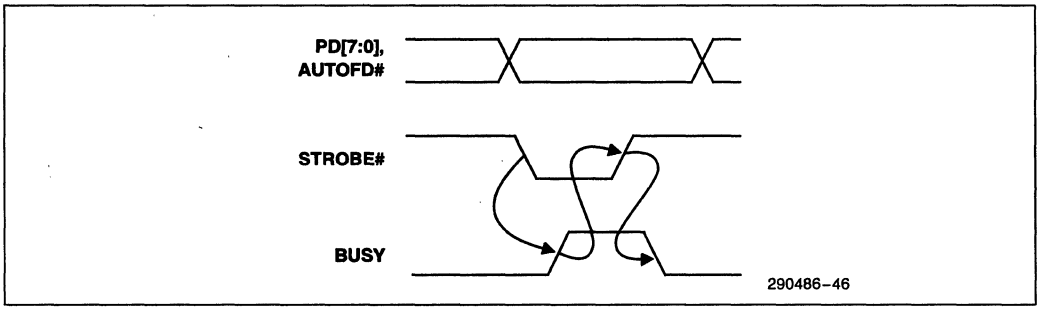


Figure 46. ECP Mode Handshake (Forward Direction)

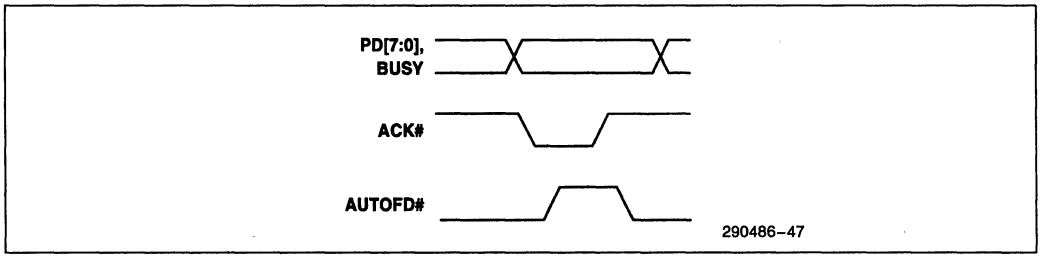


Figure 47. ECP Mode Handshake (Reverse Direction)

Test Mode (ECR[7:5] = 110) and Configuration Mode (ECR7:5] = 111)

The test mode can be used to check the FIFO read and write interrupt thresholds as described in Section 6.1.3.7, TFIFO—ECP Test FIFO Register. Note that for the 82091AA parallel port, the read and write FIFO interrupt thresholds are the same. The FIFO threshold is set by programming the PCFG1 Register in the 82091AA configuration space. The configuration mode is used to access the ECPCFGA and ECPCFGB Registers. This mode must first be set before the ECPCFGA and ECPCFGB Registers can be accessed.

6.2.3.1 FIFO Operations

The parallel port FIFO is used for ECP transfers (ECR[7:5] = 011), ISA-Compatible FIFO transfers (ECR[7:5] = 010), and Test mode (ECR[7:5] = 110). Either DMA or programmed I/O can be used for transfers between the host and the parallel port.

The FIFO threshold value is selected via the 82091AA configuration registers (PCFG1 Register). The threshold is set to either 1 (forward)/15 (reverse) or 8 in both directions. A threshold setting of 1 (forward)/15 (reverse) results in longer periods of time between service request, but requires faster servicing of both read and write requests. A threshold setting of 8 results in more service requests, but tolerates slower servicing of the requests.

In modes 010 and 011, an internal temporary holding register is used in conjunction with the 16-byte FIFO to provide 17 bytes of storage for both forward and reverse transfers. Thus, in the forward direction if the peripheral asserts the BUSY signal during the filling of the FIFO, the host needs to write 17 bytes before the FIFO full flag in the ECR is set to 1. In Test mode (110) only the 16-byte FIFO is used and the temporary holding register is not used.

The FIFO is reset by a hard reset (RSTDRV asserted) or whenever the parallel port is placed in ISA-Compatible or PS/2-Compatible modes. Note that the FIFO threshold can only be changed when the parallel port is in ISA-Compatible or PS/2-Compatible mode.

6.2.3.2 DMA Transfers

The 82091AA contains parallel port DMA request (PPDREQ) and acknowledge (PPDACK#) signals to communicate with a standard PC DMA controller. Before initiating a DMA transfer the direction bit in the PCON Register must be set to the proper direction. To initiate DMA transfers, software sets the DMAEN bit to 1 and the SERVICEINTR bit to 0 in the ECR. The PPDREQ and PPDACK# signals will then be used to fill (forward direction) or empty (reverse direction) the FIFO. When the DMA controller reaches terminal count and asserts the TC signal, an interrupt is generated and the SERVICEINTR bit is set to 1. To reset the TC interrupt, software can either switch the mode to 000 or 001 or write the DMAEN bit to 0.

In DMA mode, if 32 consecutive reads or writes are performed to the FIFO and PPDREQ is still asserted, the 82091AA negates PPDREQ for the length of the last PPDACK#/command pulse to force an arbitration cycle on the ISA Bus.

6.2.3.3 Reset FIFO and DMA Terminal Count Interrupt

The following operations are used to reset the parallel port FIFO and TC interrupt

Function	Reset Operations
FIFO	-Changing to modes 000 or 001 -Hard reset
FIFO Error	-Changing to modes 000 or 001 -Hard reset
TC Interrupt	-Changing to modes 000 or 001 -Setting DMAEN to 0 in ECR -Hard reset

6.2.3.4 Programmed I/O Transfers

Programmed I/O (non-DMA) can also be used for transfers between the host and the parallel port FIFO. Software can determine the read/write FIFO thresholds and the FIFO depth by accessing the FIFO in Test mode. To use programmed I/O transfers software sets the direction bit in the PCON Register to the desired direction and sets the DMAEN bit to 0 and the SERVICEINTR bit to 0 in the ECR. The parallel port requests programmed I/O transfers from the host by asserting IRQ5/IRQ7.

In the reverse direction an interrupt occurs when SERVICEINTR = 0 either 8 or 15 bytes (depending on threshold setting) are in the FIFO. IRQ5/IRQ7 can be used in an interrupt-driven system. The host must respond to the interrupt request by reading data from the FIFO.

In the forward direction an interrupt occurs when SERVICEINTR=0 and there are either 8 or 1 byte locations available in the FIFO (depending on threshold setting). IRQ5/IRQ7 can be used in an interrupt-driven system. The host must respond to the interrupt request by writing data to the FIFO.

6.2.3.5 Data Compression

The 82091AA supports Run Length Encoded (RLE) decompression in hardware and can transfer compressed data to the peripheral. To transfer compressed data to the peripheral (forward direction), the compression count is written to the ECPAFIFO location and the data is written to the ECPDFIFO location. The most significant bit (bit 7) in the byte written to the ECPAFIFO Register informs the peripheral whether the value is a channel address (bit 7=1) or a run length count (bit 7=0). The RLE count in the ECPAFIFO (bits[6:0]) informs the peripheral of how many times the data in the ECPDFIFO is to be repeated. An RLE count of 0 indicates that only one byte of the data is present and a count of 127 indicates to the peripheral that the next byte should be expanded 128 times. An RLE count of 1 should be avoided as it will cause unnecessary expansions. Note that the 82091AA asserts AUTOFD# to indicate that PD[7:0] contains address/RLE instead of data.

In the reverse direction, the peripheral negates the BUSY signal to indicate that PD[7:0] contains address/RLE. During an address/RLE cycle, the 82091AA checks bit 7 to see if the next byte received needs to be decompressed. If bit 7 is 0, the 82091AA decompresses (replicates) the next data received by the RLE count received on bits[6:0].

6.2.4 PARALLEL PORT EXTERNAL BUFFER CONTROL

A multi-function signal (GCS#/PPDIR) is provided for controlling optional external parallel port data buffers. The PPDIR function is only available when the 82091AA configuration is in software motherboard (SWMB) mode. In this mode, this signal operates as a parallel port direction control signal (PPDIR). Note that, if any other configuration is used (SWAI, HWB, or HWE configuration modes), this multi-function signal operates as a game port chip select (GCS#). In SWMB, PPDIR is low when PD[7:0] are outputs and high when PD[7:0] are inputs. Figure 44 shows an example of external buffers being used when the parallel port is in ECP mode.

External buffering affects the ability of the port to read software security devices. Typically these software secutiry devices are designed to hold the data pins of the parallel port connector at either high or low logic levels when the pins are not being driven by the parallel port. The bit pattern read from the parallel port by the security software may not be correctly transferred through the external buffer.

6.2.5 PARALLEL PORT SUMMARY

Table 18 summarizes the parallel port interrupt, DMA, and parallel port signal drive type for the various modes of operation.

Table 18. Parallel Port Summary

Parallel Port Mode	ECR[7:5]	PD[7:0] Direction	Parallel Port Control Signals Controlled By PCON	IRQ Enable	DMA Enable
ISA-Compatible	000	Output	Open Drain	ACKINTEN	
PS/2-Compatible	001	Bi-directional	Open Drain	ACKINTEN	
EPP	N/A	Bi-directional	Push Pull	ACKINTEN	
ISA-Compatible FIFO	010	Output	Push Pull	Always Enabled	DMAEN
ECP	011	Bi-directional	Push Pull	Always Enabled	DMAEN
ECP Test	110	Output	Push Pull	Always Enabled	DMAEN
ECP Configuration	111	Bi-directional	Push Pull	ACKINTEN	DMAEN

NOTES:
1. The selected IRQ pin (IRQ5/IRQ7) is enabled if ACKINTEN is enabled in the PCON Register. Otherwise, the IRQ pin is tri-stated.
2. PPDREQ is enabled whenever the DMAEN bit is enabled in the ECR, independent of the parallel port mode.

7.0 SERIAL PORT

The two 82091AA serial ports are identical. This section describes the serial port registers and FIFO operations.

7.1 Register Description

The register descriptions in this section apply to both serial port A and serial port B and provide a complete operational description of the serial ports. Table 19 shows the I/O address assignments for the serial port registers. The individual register descriptions follow in the order that they appear in the table. Note that serial port interrupt assignments (IRQ3 or IRQ4) and the base address assignments are made by 82091AA configuration as described in Section 4.0, AIP Configuration.

All registers are accessed as byte quantities. The base address is determined by hardware configuration at powerup (or a hard reset) or via software configuration by programming the 82091AA configuration registers as described in Section 4.0, AIP Configuration. Note that access to certain serial port registers requires prior programming of the DLAB bit in the Line Control Register (LCR).

During a hard reset (RSTDRV asserted), the 82091AA registers are set to pre-determined **default** states. The default values are indicated in the individual register descriptions. Reserved bits in the 82091AA's serial port registers must be programmed to 0 when writing the register and these bits are 0 when read. The following bit notation is used for default settings:

X Default bit position value is determined by conditions on an 82091AA signal pin.

The following nomenclature is used for serial port register access attributes:

RO **Read Only**. Note that for all registers with read only attributes, writes to the I/O address access a different register.

WO **Write Only**. Note that for all registers with write only attributes, reads to the I/O address access a different register.

R/W **Read/Write**. A register with this attribute can be read and written. Note that some read/write registers contain bits that are read only.

Table 19. Serial Port Registers

Register Address Access (AEN = 0)		Abbreviation	Register Name	Access
Base +	DLAB			
0h	0	THR	Transmit Holding Register	WO
0h	0	RBR	Receiver Buffer Register	RO
0h	1	DLL	Divisor Latch LSB	R/W
1h	1	DLM	Divisor Latch MSB	R/W
1h	0	IER	Interrupt Enable Register	R/W
2h	—	IIR	Interrupt Identification Register	RO
2h	—	FCR	FIFO Control Register	WO
3h	—	LCR	Line Control Register	R/W
4h	—	MCR	Modem Control Register	R/W
5h	—	LSR	Line Status Register	R/W
6h	—	MSR	Modem Status Register	R/W
7h		SCR	Scratch Pad Register	R/W

Table 20. Serial Port Register Summary

Bit #	Receiver Buffer Register	Transmitter Holding Register	Interrupt Enable Register	Interrupt Identification Register	FIFO Control Register	Line Control Register
0	Data Bit 0	Data Bit 0	Enable Received Data Available Interrupt	0 if Interrupt Pending	FIFO Enable	Word Length Select Bit 0
1	Data Bit 1	Data Bit 1	Enable XMTR Holding Register Empty Interrupt	Interrupt ID Bit	RCVR FIFO Reset	Word Length Select Bit 1
2	Data Bit 2	Data Bit 2	Enable RCVR Line Status Interrupt	Interrupt ID Bit	XMIT FIFO Reset	Number of Stop Bits
3	Data Bit 3	Data Bit 3	Enable Modem Status Interrupt	Interrupt ID Bit (Non-FIFO = 0)	DMA Mode Select	Parity Enable
4	Data Bit 4	Data Bit 4	0	0	Reserved	Event Parity Select
5	Data Bit 5	Data Bit 5	0	0	Reserved	Stick Parity
6	Data Bit 6	Data Bit 6	0	FIFOs Enabled (Non-FIFO = 0)	RCVR Trigger (LSB)	Set Break
7	Data Bit 7	Data Bit 7	0	FIFOs Enabled (Non-FIFO = 0)	RCVR Trigger (MSB)	Divisor Latch Access Bit (DLAB)

Table 20. Serial Port Register Summary (Continued)

Bit #	Modem Control Register	Line Status Register	Modem Status Register	ScratchPad Register	Divisor Latch - MSB	Divisor Latch - LSB
0	Data Terminal Ready (DTR)	Data Ready (DR)	Delta Clear to Send	Bit 0	Bit 0	Bit 8
1	Request to Send (RTS)	Overrun Error (OE)	Delta Data Set Ready	Bit 1	Bit 1	Bit 9
2	Out 1 Bit	Parity Error (PE)	Trailing Edge Ring Indicator	Bit 2	Bit 2	Bit 10
3	IRQ Enable	Framing Error (FE)	Delta Data Carrier Detect	Bit 3	Bit 3	Bit 11
4	Loop	Break Interrupt (BI)	Clear to Send (CTS)	Bit 4	Bit 4	Bit 12
5	0	Transmitter Holding Register (THRE)	Data Set Ready (DSR)	Bit 5	Bit 5	Bit 13
6	0	Transmitter Empty (TEMT)	Ring Indicator (RI)	Bit 6	Bit 6	Bit 14
7	0	Error in RCVR FIFO	Data Carrier Detect (DCD)	Bit 7	Bit 7	Bit 15

7.1.1 THR(A,B)—TRANSMITTER HOLDING REGISTER

I/O Address: Base +0h (DLAB=0)
Default Value: 00h
Attribute: Write Only
Size: 8 bits

The THR contains data to be transmitted out on the SOUT[A,B] signal line. Bit 0 is the least significant bit and is the first bit serially transmitted. If the serial word length is less than 8 bits (as selected in the LCR), the data word must be written to this register right-justified. Bit positions above the number of bits selected for the word size are discarded (not transmitted).

Bit	Description
7:0	**Transmit Data:** Bits[7:0] correspond to SD[7:0].

7.1.2 RBR(A,B)—RECEIVER BUFFER REGISTER

I/O Address: Base +0h (DLAB=0)
Default Value: 00h
Attribute: Read Only
Size: 8 bits

The RRB contains data received from the SIN[A,B] signal line. Bit 0 is the least significant bit and is the first bit serially received. If the serial word length is less than 8 bits (as selected in the LCR), the data word in this register is right-justified. Bit positions above the number of bits selected for the word size are 0.

Bit	Description
7:0	**Receiver Data:** Bits[7:0] correspond to SD[7:0].

7.1.3 DLL(A,B), DLM(A,B)—DIVISOR LATCHES (LSB AND MSB) REGISTERS

I/O Address: Base +0h,1h (DLAB=1)
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

The 82091AA contains two independently programmable baud rate generators. The 24 MHz crystal oscillator frequency input is divided by 13, resulting in a frequency of 1.8462 MHz. This frequency is the input to each baud rate generator and is divided by the divisor of the associated serial port. The output frequency of the baud rate generator (BOUT[A,B]) is 16 x the baud rate.

$$\text{divisor } \# = (\text{frequency input})/(\text{baud rate} \times 16)$$

The output of each baud rate generator drives the transmitter and receiver sections of the associated serial port. Two 8-bit latches per serial port store the divisor in a 16-bit binary format. These divisor latches must be loaded during initialization to ensure proper operation of the baud rate generator. Upon loading either of the divisor latches, a 16-bit baud counter is loaded. Table 21 provides decimal divisors to use with crystal frequencies of 24 MHz. Using a divisor of zero is not recommended.

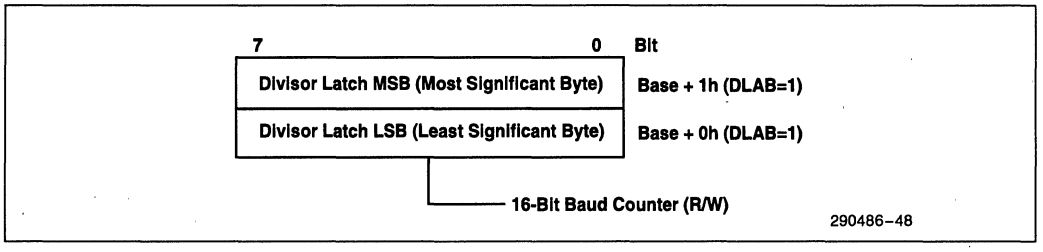


Figure 48. Divisor Latches (LSB and MSB) Registers

Bit	Description
7:0	**Divisor Latch Data:** Bits[7:0] correspond to SD[7:0].

Table 21. AIP Serial Port A and B Divisors, Baud Rates, and Clock Frequencies

	24 MHz Input Divided to 1.8461 MHz	
Baud Rate	**Decimal Divisor for 16x Clock**	**Percent Error**
50	2304	0.1
75	1536	
110	1047	
134.5	857	0.4
150	768	—
300	384	—
600	192	—
1200	96	—
1800	64	—
2000	58	0.5
2400	48	—
3600	32	—
4800	24	—
7200	16	—
9600	12	—
19200	6	—
38400	3	—
56000	2	3.0
115200	1	—

7.1.4 IER(A,B)—INTERRUPT ENABLE REGISTER

I/O Address: Base + 1h (DLAB = 0)
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

This register enables/disables interrupts for five types of serial port conditions. If a particular condition occurs whose interrupt is disabled in this register, the corresponding interrupt status bit in the IIR will not be set and an interrupt request (IRQ3 or IRQ4) will not be generated.

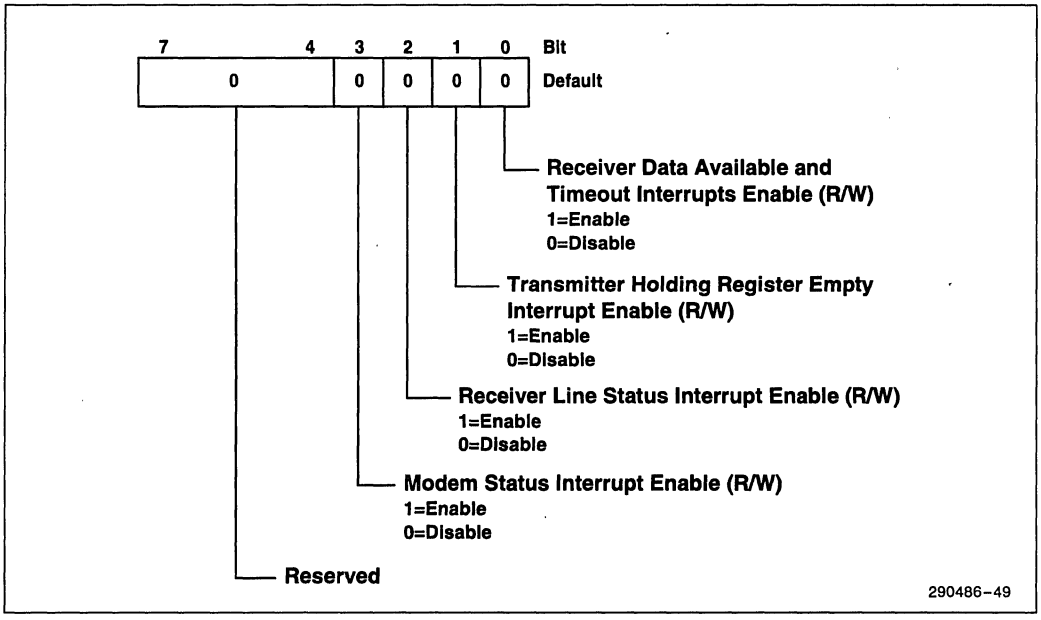


Figure 49. Interrupt Enable Register

Bit	Description
7:4	**RESERVED**
3	**MODEM INTERRUPT ENABLE (MIE):** When MIE = 1, the Modem Status Interrupt is enabled. When MIE = 0, the Modem Status Interrupt is disabled.
2	**RECEIVER INTERRUPT ENABLE (RIE):** When RIE = 1, the Receiver Line Status interrupt is enabled. When RIE = 0, the receiver line status interrupt is disabled.
1	**TRANSMITTER HOLDING REGISTER EMPTY INTERRUPT ENABLE (THEIE):** When THREIE = 1, the Transmitter Holding Register Empty Interrupt is enabled. When THREIE = 0, the Transmitter Holding Register Empty Interrupt is disabled.
0	**RECEIVER DATA AVAILABLE INTERRUPT ENABLE AND TIMEOUT INTERRUPT ENABLE IN FIFO MODE (RAVIE):** When RAVIE = 1, the Received Data Available Interrupt is Enabled. When RAVIE = 0, the Received Data Available Interrupt is disabled. In addition, in the FIFO Mode, this bit enables the Timeout Interrupt when set to 1 and disables the Timeout Interrupt when set to 0.

7.1.5 IIR(A,B)—INTERRUPT IDENTIFICATION REGISTER

I/O Address: Base +2h
Default Value: 01h
Attribute: Read Only
Size: 8 bits

This register provides interrupt status and indicates whether the serial port receive/transmit FIFOs are enabled (FIFO mode) or disabled (non-FIFO mode). In order to provide minimum software overhead during data character transfers, the serial port prioritizes interrupts into four levels and records these in the Interrupt Identification Register. The four levels of interrupt conditions in order of priority are Receiver Line Status; Received Data Ready; Transmitter Holding Register Empty; and Modem Status. When the CPU accesses the IIR, the serial port freezes all interrupts and indicates the highest priority pending interrupt to the CPU. While this CPU access is occurring, the serial port records new interrupts, but does not change its current indication until the current access is complete.

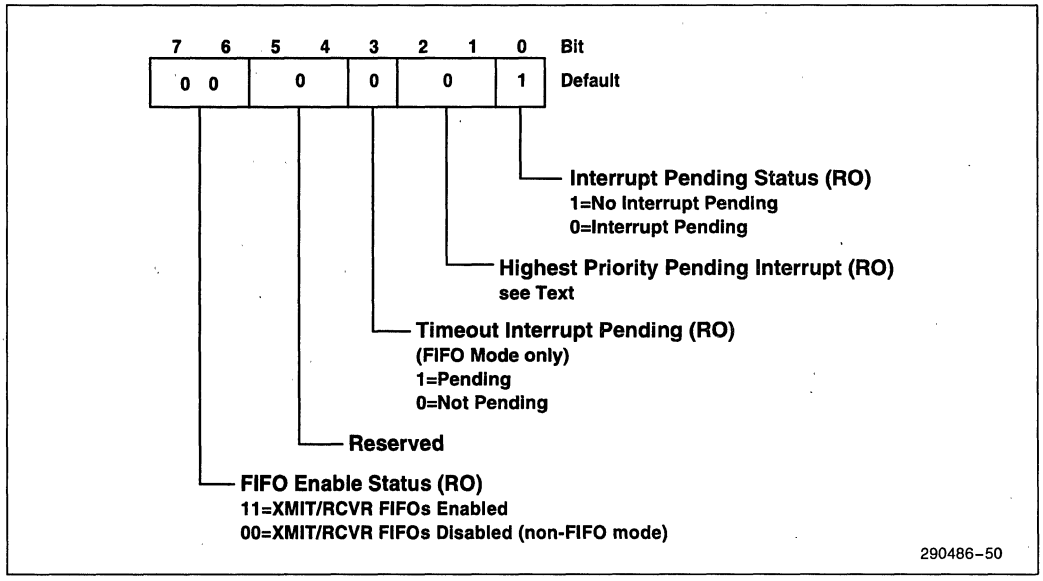


Figure 50. Interrupt Identification Register

Bit	Description
7:6	**FIFO MODE ENABLE STATUS (FIFOES):** This status field indicates whether the serial port is in FIFO mode or non-FIFO mode (FIFO/non-FIFO mode is selected via the FCR). When FIFOES = 11, the serial port is in FIFO mode (FIFOs enabled). When FIFOES = 00, the serial port is in non-FIFO mode (FIFOs disabled). The 82091AA never sets this field to either = 01 or 10.
5:4	**RESERVED**
3	**TIMEOUT INTERRUPT PENDING (TOUTIP)—FIFO MODE ONLY:** In the non-FIFO mode, this bit is 0. In FIFO mode TOUTIP is set to 1 when no characters have been removed from or input to the receive FIFO during the last 4 character times and there is at least 1 character in the FIFO during this time. When a timeout interrupt is pending, the 82091AA sets this bit along with bit 2 of this register.
2:1	**HIGHEST PRIORITY INTERRUPT INDICATOR:** This field identifies the highest priority interrupt pending as indicated in Table 22.
0	**INTERRUPT PENDING STATUS (IPS):** This bit can be used in an interrupt environment to indicate whether an interrupt condition is pending. When IPS = 0, an interrupt is pending and the IIR contents may be used as a pointer to the appropriate interrupt service routine. When IPS = 1, no interrupt is pending.

Table 22. Interrupt Priority

FIFO Mode Only	Interrupt Identification Register			Interrupt Set and Reset Functions			
Bit 3	Bit 2	Bit 1	Bit 0	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Control
0	0	0	1	—	None	None	—
0	1	1	0	Highest	Receiver Line Status	Overrun Error, Parity Error, Framing Error, or Break Interrupt	Reading the Line Status Register
0	1	0	0	Second	Received Data Available	Receiver Data Available	Read Receiver Buffer
1	1	0	0	Second	Character Timeout Indication	No Characters Have Been Removed from or Input to the RCVR FIFO during the Last 4 Char. Times and there is at least 1 Char. in it during this time	Reading the Receiver Buffer Register
0	0	1	0	Third	Transmitter Holding Register Empty	Transmitter Holding Register Empty	Reading the IIR Register (if Source or Interrupt) or Writing the Transmitter Holding Register
0	0	0	0	Fourth	Modem Status	Clear to Send or Data Set Ready or Ring Indicator or Data Carrier Detect.	Reading the Modem Status Register

7.1.6 FCR(A,B)—FIFO CONTROL REGISTER

I/O Address: Base + 2h
Default Value: 00h
Attribute: Write Only
Size: 8 bits

FCR is a write only register that is located at the same address as the IIR (the IIR is a read only register). FCR enables/disables the transmit/receive FIFOs, clears the transmit/receive FIFOs, and sets the receive FIFO trigger level.

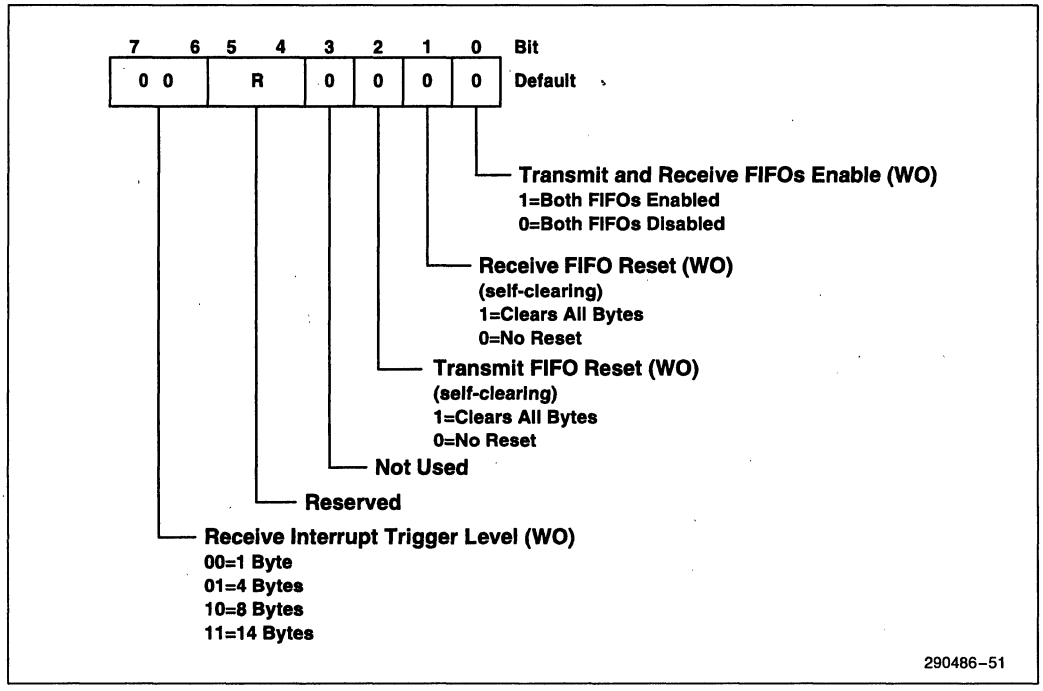


Figure 51. FIFO Control Register

Bit	Description
7:6	**INTERRUPT TRIGGER LEVEL (ITL):** The ITL field indicates the interrupt trigger level. When the number of bytes in the receive FIFO equals the interrupt trigger level programmed into this field and the Received Data Available Interrupt enabled (via the IER), an interrupt will be generated and the appropriate bits set in the IIR. **Bits [7:6] — Trigger Level (Bytes)** 0 0 — 01 (default) 0 1 — 04 1 0 — 08 1 1 — 14
5:4	**RESERVED**
3	**NOT USED:** Writing to this bit causes no change in serial port operations. The serial port does not support DMA operations. Note that the TXRDY# and RXRDY# pins are not available in the 82091AA.
2	**RESET TRANSMITTER FIFO (RESETTF):** When RESETTF is set to a 1, the FIFO counter is set to 0. The shift register is not cleared. When the FIFO is cleared, the 82091AA sets this bit to 0.
1	**RESET RECEIVER FIFO (RESETRF):** When RESETRF is set to a 1, the FIFO counter is set to 0. The shift register is not cleared. When the FIFO is cleared, the 82091AA sets this bit to 0.
0	**TRANSMIT AND RECEIVE FIFO ENABLE (TRFIFOE):** TRFIFOE enables/disables the transmit and receive FIFOs. When TRFIFOE = 1, both FIFOs are enabled (FIFO Mode). When TRFIFOE = 0, the FIFOs are both disabled (non-FIFO MODE). Writing a 0 to this bit clears all bytes in both FIFOs. When changing from FIFO mode to non-FIFO mode and vice versa, data is automatically cleared from the FIFOs. This bit must be written with a 1 when other bits in this register are written or the other bits will not be programmed.

7.1.7 LCR(A,B)—LINE CONTROL REGISTER

I/O Address: Base +3h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

This register specifies the format of the asynchronous data communications exchange. LCR also enables/disables access to the Baud Rate Generator Divisor latches or the Transmitter Data Holding Register, Receiver Buffer Register, and Interrupt Enable Register.

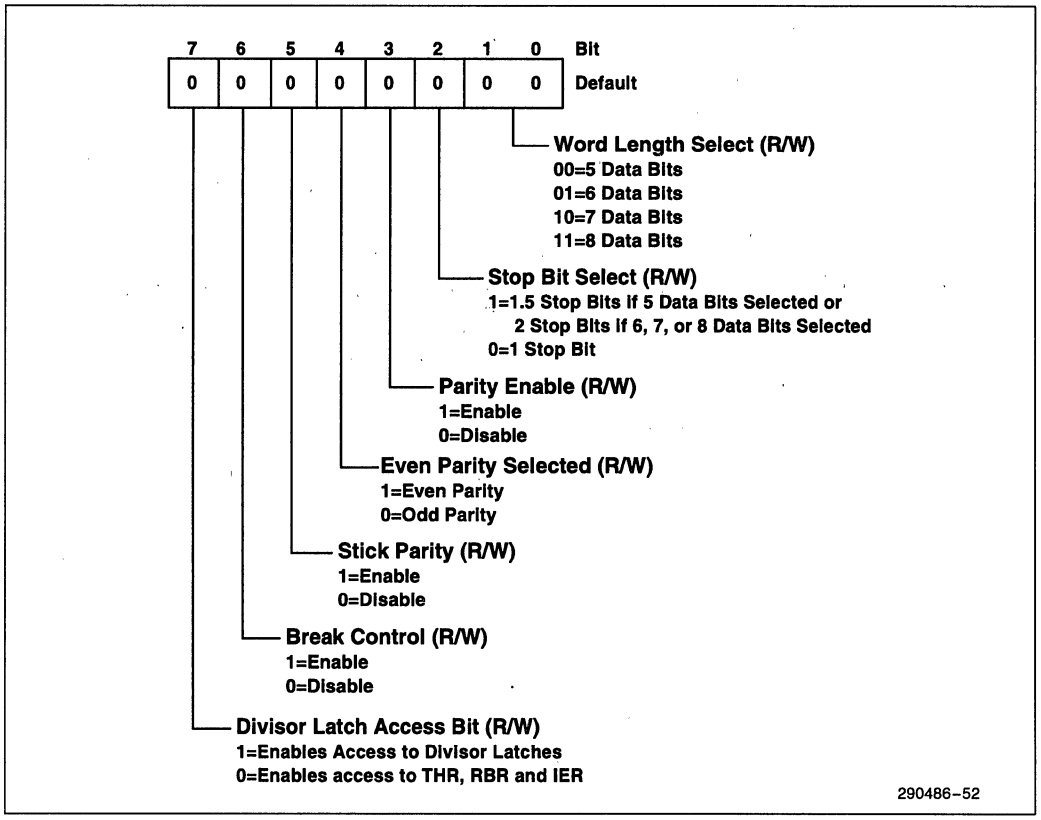


Figure 52. Line Control Register

Bit	Description
7	**DIVISOR LATCH ACCESS BIT (DLAB):** DLAB controls access to the Baud Rate Generator Divisor Latches (and to the Transmit Holding Register, Receiver Buffer Register and Interrupt Enable Register which are located at the same I/O addresses). When DLAB = 1, access to the two Divisor Latches is selected and access to the THR, RBR, and IER is disabled. When DLAB = 0, access to the two Divisor Latches is disabled and access to the THR, RBR, and IER is selected. During test mode operations, DLAB must be set to 1 for the BOUT signal to appear on the SOUT pin.
6	**BREAK CONTROL (BRCON):** When BRCON = 1, a break condition is transmitted from the 82091AA serial port to the receiving device. When BRCON = 1, the serial output (SOUT) is forced to the 'spacing' state (logical 0). BRCON only affects the SOUT signal and has no effect on the transmitter logic. Note that this feature permits the CPU to alert a terminal. If the following sequence is used, no erroneous characters will be transmitted because of the break. 1. Wait for the transmitter to be idle (TEMT = 1). 2. Set break (BRCON = 1) for the appropriate amount of time. If the transmitter will be used to time the break duration, then check that TEMT = 1 before clearing the BRCON. 3. Clear break (BRCON = 0) when normal transmission has to be restored. During the break, the transmitter can be used as a character timer to accurately establish the break duration by sending characters and monitoring THRE and TEMT.
5	**STICKY PARITY (STICPAR):** STICPAR is the Stick Parity bit. When parity is enabled (PAREN = 1) this bit is used in conjunction with EVENPAR to select "Mark" or "Space" Parity. When bits PAREN, EVENPAR and STICPAR are 1, the parity bit is transmitted and checked as a 0 (Space Parity). If bits PAREN and STICPAR are 1 and EVENPAR is 0, the parity bit is transmitted and checked as a 1 (Mark Parity). When STICPAR = 0, stick parity is disabled.
4	**EVEN PARITY SELECT (EVENPAR):** EVENPAR selects between even and odd parity. When parity is enabled (PAREN = 1) and EVENPAR = 0, an odd number of 1s is transmitted or checked in the data word bits and parity bit. When parity is enabled and EVENPAR = 1, an even number of 1s is transmitted or checked.
3	**PARITY ENABLE (PAREN):** This bit enables/disables parity generation and checking. When PAREN = 1, a parity bit is generated (transmit data) or checked (receive data) between the last data bit and stop bit of the serial data. (The Parity bit is used to produce an even or odd number of 1s when the data bits and the Parity bit are summed.) When PAREN = 0, parity generation and checking is disabled.
2	**STOP BITS (STOPB):** This bit specifies the number of stop bits transmitted with each serial character. When STOPB = 0, one stop bit is generated in the transmitted data. When STOPB = 1 and a 5-bit data length is selected, one and a half stop bits are generated. When STOPB = 1 and either a 6-, 7-, or 8-bit data length is selected, two stop bits are generated. The receiver checks the first Stop bit only, regardless of the number of Stop bits selected.
1:0	**SERIAL DATA BITS (SERIALDB):** This field specifies the number of data bits in each transmitted or received serial character as follows: **Bits[1:0] — Data Length** 0 0 — 5 Bits - Default 0 1 — 6 Bits 1 0 — 7 Bits 1 1 — 8 Bits

5

7.1.8 MCR(A,B)—MODEM CONTROL REGISTER

I/O Address: Base +4h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

This register controls the interface with the modem or data set (or a peripheral device emulating a modem).

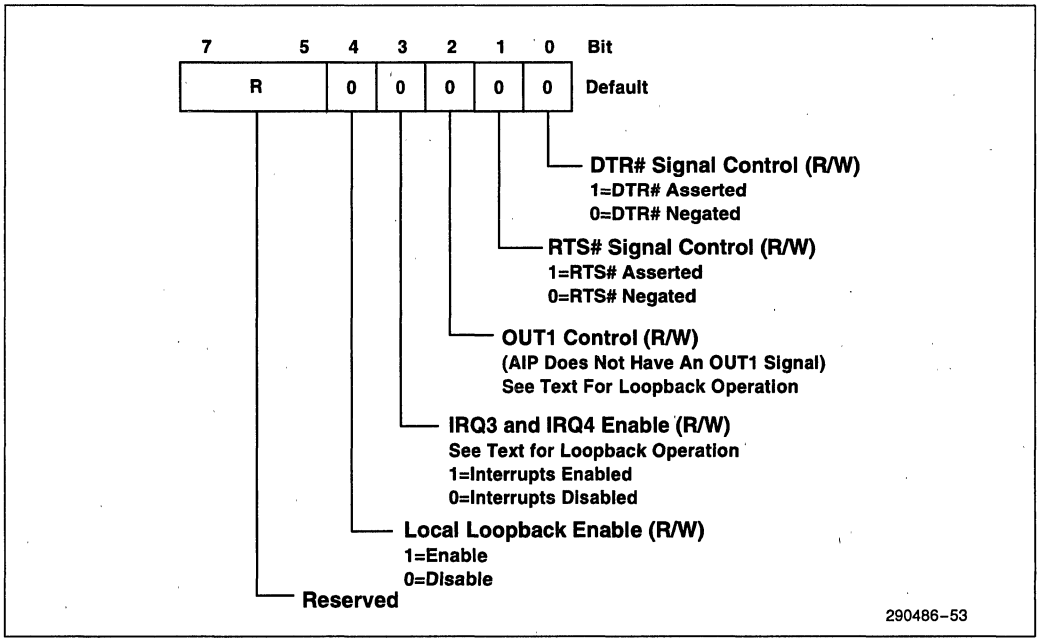


Figure 53. Modem Control Register

Bit	Description
7:5	**RESERVED**
4	**LOOPBACK MODE ENABLE (LME):** LME provides a local loopback feature for diagnostic testing of the serial port module. When LME = 1, the following occurs: 1. The transmitter Serial Output (SOUT) is set to the Marking (logic 1) state. 2. The receiver Serial Input (SIN) is disconnected. 3. The output of the Transmitter Shift Register is "looped back"(connected) to the Receiver Shift Register. 4. The four modem control inputs (DSR#, CTS#, RI and DCD#) are disconnected. 5. The DTRC, RTSC, OUT1C, IE bits in the MCR are internally connected to DSRS, CTSS, RIS, and DCDS in MSR, respectively. 6. The modem control output pins are forced to their high (inactive) state. 7. Data that is transmitted is immediately received. This feature allows the CPU to verify the transmit and received data paths of the serial port. In the loopback mode, the receiver and transmitter interrupts are fully operational. The modem status interrupts are fully operational. The modem status interrupts are also operational, but the interrupt sources are the lower four bits of MCR instead of the four modem control inputs. Writing a 1 to any of these 4 MCR bits (bits[3:0]) causes an interrupt. In Loopback Mode the interrupts are still controlled by the Interrupt Enable Register. The IRQ3 and IRQ4 signal pins are tri-stated in the loopback mode.
3	**INTERRUPT ENABLE (IE):** When IE = 1, the associated interrupt is enabled (either IRQ3 or IRQ4 as selected via the associated serial port configuration register - A or B). In Local Loopback Mode, this bit controls bit 7 of the Modem Status Register.
2	**OUT1 BIT CONTROL (OUT1C):** This bit is the OUT1 bit. It does not have an output pin associated with it. It can be written to and read by the CPU. In Local Loopback Mode, this bit controls bit 6 of the Modem Status Register.
1	**REQUEST TO SEND CONTROL (RTS):** This bit controls the Request to Send (RTS#) output. When RTSC = 1, the RTS# output is asserted. When RTSC = 0, the RTS# output is negated. In Local Loopback Mode, this bit controls bit 4 of the Modem Status Register.
0	**DATA TERMINAL READY CONTROL (DTRC):** This bit controls the Data Terminal Ready (DTR#) output. When DTRC = 1, the DTR# output is asserted. When DTRC = 0, the DTR# output is negated. In Local Loopback Mode, this bit controls bit 5 of the Modem Status Register. **NOTE:** The DTR# and RTS# outputs of the serial port may be applied to an EIA inverting line driver (such as the DS1488) to obtain the proper polarity input at the modem or data set.

7.1.9 LSR(A,B)—LINE STATUS REGISTER

I/O Address: Base + 5h
Default Value: 60h
Attribute: Read/Write
Size: 8 bits

This 8-bit register provides data transfer status information to the CPU. Note that the Line Status Register is intended for read operations only. Writing to this register is not recommended and could result in unintended operations. For this reason, the figure shows these bits as RO (read only).

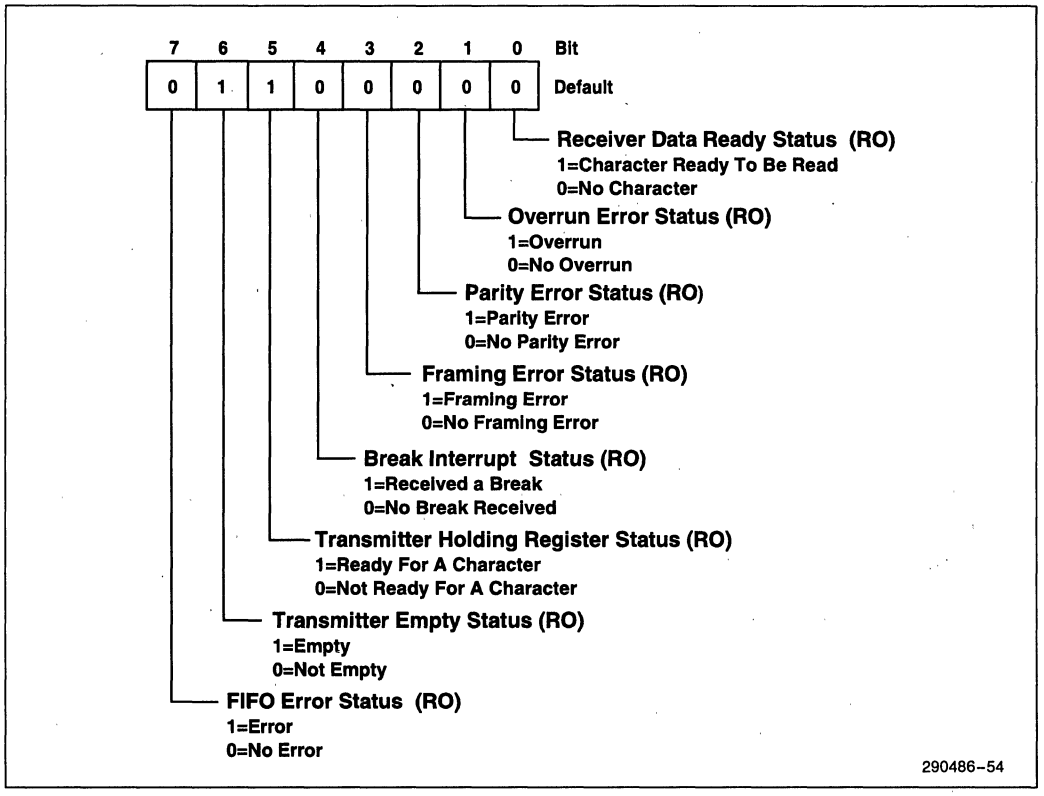


Figure 54. Line Status Register

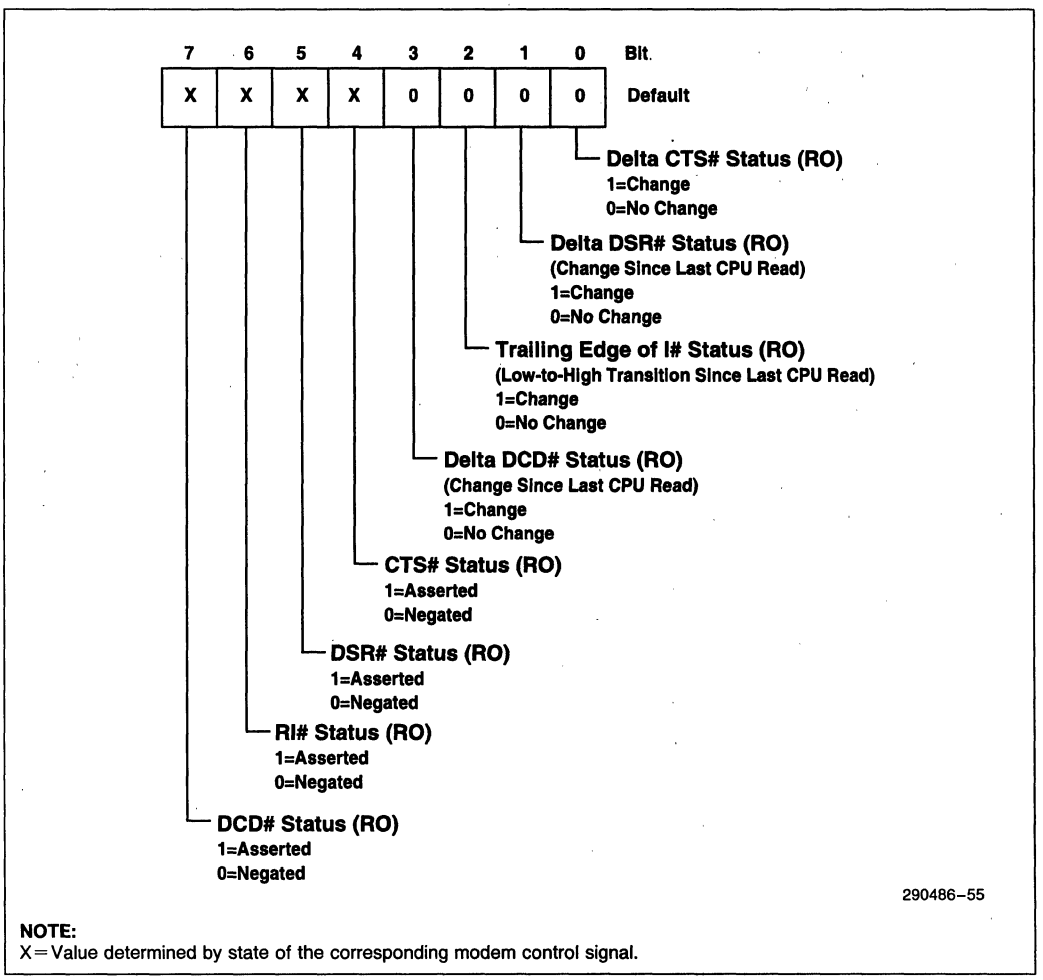
Bit	Description
7	**FIFO ERROR STATUS (FIFOE):** In the non-FIFO Mode this is a 0. In the FIFO Mode, FIFOE is set to 1 when there is at least one parity error, framing error, or break indication in the FIFO. FIFOE is set to 0 when the CPU reads the LSR, if there are no subsequent errors in the FIFO.
6	**TRANSMITTER EMPTY STATUS (TEMT):** This bit is the Transmitter Empty (TEMT) indicator. When the Transmitter Holding Register (THR) and the Transmitter Shift Register (TSR) are both empty, the 82091AA sets TEMT to a 1. When either the THR or TSR contains a data character, TEMT is set to a 0. The default is 0. In FIFO mode, this bit is set to 1 when the transmitter FIFO and the shift register are both empty.

Bit	Description
5	**TRANSMITTER HOLDING REGISTER STATUS (THRE):** This bit is the Transmitter Holding Register Empty (THRE) indicator. THRE indicates that the serial port module is ready to accept a new character for transmission. In addition, this bit causes the serial port module to issue an interrupt to the CPU when the Transmit Holding Register Empty Interrupt enable is set to a 1. THRE is set to 1 when a character is transferred from the Transmitter Holding Register into the Transmitter Shift Register. THRE is set to 0 when the CPU loads the Transmitter Holding Register. In the FIFO mode, this bit is set to a 1 when the transmit FIFO is empty, and is set to 0 when at least 1 byte is written to the transmit FIFO.
4	**BREAK INTERRUPT STATUS (BI):** This bit is the Break Interrupt (BI) indicator. BI is set to a 1 when the received data input is held in the Spacing state (logic 0) for longer than a full word transmission time (that is, the total time of Start bit + data bits + Parity + Stop bits). When the CPU reads the contents of the Line Status Register, BI is set to 0. In FIFO mode, this error is associated with the particular character in the FIFO associated with the Break. BI is indicated to the CPU when its associated character is at the top of the FIFO. When break occurs only one character is loaded into the FIFO. Restarting after a break is received requires the SIN pin to be a logical 1 for at least ½ bit times. **NOTE:** Bits[3:0] are the error conditions that produce a Receiver Line Status interrupt whenever any of the corresponding conditions are detected and that interrupt is enabled.
3	**FRAMING ERROR STATUS (FE):** This bit is the Framing Error (FE) indicator. FE indicates that the received character did not have a valid stop bit. FE is set to a 1 when the stop bit following the last data bit or parity bit is 0 (spacing level). FE is set to 0 when the CPU reads the contents of the Line Status Register. In FIFO mode, this error is associated with the particular character in the FIFO that it applies to. This error is revealed to the CPU when its associated character is at the top of the FIFO. When a framing error is due to the next start bit, the serial port attempts to resynchronize. In this case, the serial port module samples this start bit twice and, if no FE exists, then the module takes in the rest of the bits.
2	**PARITY ERROR STATUS (PE):** This bit is the Parity Error (PE) indicator. PE indicates that the received data character does not have the correct even or odd parity, as selected by the EVENPAR bit in the Line Status Register. When a parity error is detected, PE is set to 1. PE is set to 0 when the CPU reads the contents of the Line Status Register. In the FIFO mode, this error is associated with the particular character in the FIFO that it applies to. This error is indicated to the CPU when its associated character is at the top of the FIFO.
1	**OVERRUN ERROR STATUS (OE):** OE indicates that data in the Receiver Buffer Register was not read by the CPU before the next character was transferred into the Receiver Buffer Register. In this case, the previous character is overwritten. When an overrun is detected, OE is set to 1. when the CPU reads the Line Status Register, OE is set to 0. This bit is read only. If the FIFO mode data continues to fill the FIFO beyond the trigger level, an overrun error will occur only after the FIFO is completely full and the next character has been received in the shift register. OE is indicated to the CPU as soon as it happens. The character in the shift register is overwritten, but it is not transferred to the FIFO.
0	**RECEIVER DATA READY STATUS (DR):** DR is set to 1 when a complete incoming character has been received and transferred into the Receiver Buffer Register or the FIFO. When the data in the Receiver Buffer Register or FIFO is read, DR is set to 0. This bit is read only.

7.1.10 MSR(A,B)—MODEM STATUS REGISTER

I/O Address: Base +6h
Default Value: XXXX 0000
Attribute: Read/Write
Size: 8 bits

The MSR provides the current state of the control lines from the Modem (or peripheral device) to the CPU. Bits[7:4] provide the status of the DCD#, RI, DSR#, and CTS# Modem signals. In addition to the current-state information of the Modem signals, bits[3:0] provide change information for these signals. Bits[3:0] are set to a 1 when the corresponding input signal changes state. Bits[3:0] are set to a 0 when the CPU reads the Modem Status Register.



NOTE:
X = Value determined by state of the corresponding modem control signal.

Figure 55. Modem Status Register

Bit	Description
7	**DATA CARRIER DETECT STATUS:** This bit is the compliment of the Data Carrier Detect (DCD#) input. If bit 4 of the MCR is set to a 1, this bit is equivalent to IRQ ENABLE in the MCR.
6	**RING INDICATOR STATUS:** This bit is the compliment of the Ring Indicator (RI) input. If bit 4 of the MCR is set to a 1, this bit is equivalent to OUT1 in the MCR.
5	**DATA SET READY STATUS:** This bit is the compliment of the Data Set Ready (DSR#) input. If bit 4 of the MCR is set to a 1, this bit is equivalent to DTR in the MCR.
4	**CLEAR TO SEND STATUS:** This bit is the compliment of the Clear to Send (CTS#) input. If bit 4 of the MCR is set to a 1, this bit is equivalent to RTS in the MCR.
3	**DELTA DATA CARRIER DETECT STATUS:** This bit is the Delta Data Carrier Detect (DDCD) indicator. Bit 3 indicates that the DCD# input to the chip has changed state. **NOTE:** Whenever bit 0, 1, 2, or 3 is set to logic 1, a Modem Status Interrupt is generated.
2	**TRAILING EDGE OF RING INDICATOR STATUS:** This bit is the Trailing Edge of Ring Indicator (TERI) detector. Bit 2 indicates that the RI# input to the chip has changed from a low to a high state.
1	**DELTA DATA SET READY STATUS:** This bit is the Delta Data Set Ready (DDSR) indictor. Bit 1 indicates that the DSR# input to the chip has changed state since the last time it was read by the CPU.
0	**DELTA CLEAR TO SEND STATUS:** This bit is the Delta Clear to Send (DCTS) indicator. Bit 0 indicates that the CTS# input to the chip has changed state since the last time it was read by the CPU.

7.1.11 SCR(A,B)—SCRATCHPAD REGISTER

I/O Address: Base +7h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

This 8-bit read/write register does not control the serial port module in any way. It is intended as a scratchpad register to be used by the programmer to hold data temporarily.

Bit	Description
7:0	**SCRATCHPAD DATA:** Bits[7:0] of this register correspond to SD[7:0].

7.2 FIFO Operations

This section describes the FIFO operations for interrupt and polled modes.

7.2.1 FIFO INTERRUPT MODE OPERATION

When the Receive FIFO and receiver interrupts are enabled (FCR0 = 1 and IER0 = 1), receiver interrupts occur as follows:

1. The receive data available interrupt is invoked when the FIFO has reached its programmed trigger level. The interrupt is cleared when the FIFO drops below the programmed trigger level.
2. The IIR receive data available indication also occurs when the FIFO trigger level is reached, and like the interrupt, the bits are cleared when the FIFO drops below the trigger level.
3. The receiver line status interrupt (IIR-06h), as before, has higher priority than the received data available (IIR = 04h) interrupt.
4. The data ready bit (LSR0) is set as soon as a character is transferred from the shift register to the receive FIFO. This bit is set to 0 when the FIFO is empty.

When receiver FIFO and receiver interrupts are enabled, receiver FIFO timeout interrupts occur as follows:

1. A FIFO timeout interrupt occurs, if the following conditions exist:
   a. At least one character is in the FIFO.
   b. The most recent serial character received was longer than 4 continous character times ago (if 2 stop bits are programmed, the second one is included in this time delay).
   c. The most recent CPU read of the FIFO was longer than 4 continous character times ago.

   The maximum time between a received character and a timeout interrupt is 160 ms at 300 baud with a 12-bit receive character (i.e., 1 start, 8 data, 1 parity, and 2 stop bits).
2. Character times are calculated by using the RCLK input for a clock signal (this makes the delay proportional to the baud rate).
3. When a timeout interrupt occurs, it is cleared and the timer reset when the CPU reads one character from the receiver FIFO.
4. When a timeout interrupt does not occur, the timeout timer is reset after a new character is received or after the CPU reads the receiver FIFO.

When the transmit FIFO and transmitter interrupts are enabled (FCR0 = 1, IER1 = 1), transmit interrupts occur as follows:

1. The transmitter holding register interrupt occurs when the transmit FIFO is empty. The interrupt is cleared as soon as the transmitter holding register is written (1 to 16 characters may be written to the transmit FIFO while servicing the interrupt) or the IIR is read.

Character timeout and receiver FIFO trigger level interrupts have the same priority as the current received data available interrupt. Transmit FIFO empty has the same priority as the current transmitter holding register empty interrupt.

7.2.2 FIFO POLLED MODE OPERATION

With FIFO = 1, setting IER[3:0] to all 0s puts the serial port in the FIFO polled mode of operation. Since the receiver and transmitter are controlled separately, either one or both can be in the polled mode of operation.

In this mode, software checks receiver and transmitter status via the LSR. As stated in the register description:

- LSR0 is set as long as there is one byte in the receiver FIFO.
- LSR1 and LSR4 specify which error(s) has occurred. Character error status is handled the same way as interrupt mode. The IIR is not affected since IER2 = 0.
- LSR5 indicates when the transmitter FIFO is empty.
- LSR6 indicates that both the transmitter FIFO and shift register are empty.
- LSR7 indicates whether there are any errors in the receiver FIFO.

8.0 FLOPPY DISK CONTROLLER

The 82091AA's Floppy Disk Controller (FDC) is functionally compatible with 82078/82077SL/82077AA/8272A floppy disk controllers. During 82091AA configuration, the FDC can be configured for either two drive support or four drive support via the FCFG1 Register. This section provides a complete description of the FDC when it is configured for two drive support. Additional information on four drive support is provided in Appendix A, FDC Four Drive Support.

NOTE:
For FDC compatibility and programming guidelines, refer to the 82078 Floppy Disk Controller Data sheet.

8.1 Floppy Disk Controller Registers

The FDC contains seven status, control, and data registers. Table 23 shows the I/O address assignments for the FDC registers and the individual register descriptions follow in the order that they appear in the table. The registers provide control/status information and data paths for transfering data between the floppy disk controller interface and the 8-bit host interface. In some cases, two different registers occupy the same I/O address. In these cases, one register is read only and the other is write only (i.e., a read to the I/O address accesses one register and a write accesses the other register).

All registers are accessed as byte quantities. The base address is determined by hardware configuration at powerup (or a hard reset) or via software configuration by programming the 82091AA configuration registers as described in Section 4.0, AIP Configuration.

During a hard reset (RSTDRV asserted), the 82091AA registers are set to pre-determined **default** states. The default values are indicated in the individual register descriptions. Reserved bits in the FDC registers must be programmed to 0 when writing the register and these bits are 0 when read. The following bit notation is used for default settings:

X Default bit position value is determined by conditions on an 82091AA signal pin.

The following nomenclature is used for register access attributes:

RO **Read Only.** Note that for registers with read only attributes, writes to the I/O address have no affect on floppy disk operations.

WO **Write Only.** Note that for all FDC registers with write only attributes, reads of the I/O address access a different register.

R/W **Read/Write.** A register with this attribute can be read and written. Note that individual bits in some read/write registers may be read only.

Table 23 lists the register accesses that bring the FDC out of a powerdown state. All other registers accesses are possible without waking the part from a powerdown state and reads from these registers reflects the true status as shown in the register description. For writes that do not affect the powerdown state, the FDC retains the data and will subsequently reflect it when the FDC awakens. Note that for accesses that do not affect powerdown, the access may cause a temporary increase in FDC power consumption. The FDC reverts back to low power mode when the access has been completed. None of the extended registers effect the behavior of the powerdown mode.

Table 23. Floppy Disk Controller Registers(1)

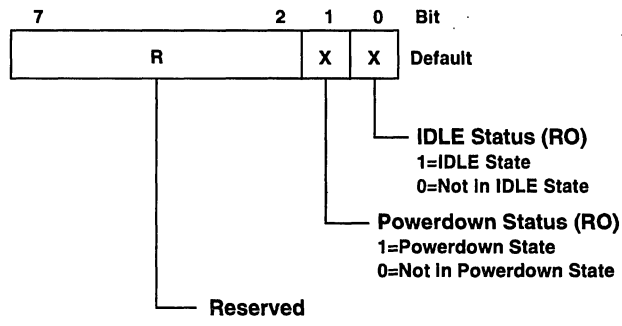
FDC Register Address Access Base +	Abbreviation	Register Name	Access Wakes Up FDC	Access
0h	—	Reserved	—	—
1h	SRB	Status Register B	No	RO
2h	DOR	Digital Output Register	No(2)	R/W
3h	TDR	Tape Drive Register	No	R/W
4h	MSR	Main Status Register	Yes	RO
4h	DSR	Datarate Select Register	No(2)	WO
5h	FIFO	Data FIFO	Yes	R/W
6h	—	Reserved	—	—
7h	DIR#	Digital Input Register	No	RO
7h	CCR	Configuration Control Register		WO

NOTES:
1. The base address is 3F0h (primary address) or 370 (secondary address).
2. While writing to the DOR or DSR does not wake up the FDC, writing any of the motor enable bits in the DOR or invoking a software reset (either via DOR or DSR reset bits) will wake up the FDC.

8.1.1 SRB—STATUS REGISTER B (EREG EN = 1)

I/O Address: Base + 1h
Default Value: RRRR RRXX
Attribute: Read/Write
Size: 8 bits

SRB provides status and control information when auto powerdown is enabled. In the AT/EISA mode the SRB is made available whenever the EREG EN bit in the POWERDOWN MODE Command is set to 1. When EREG EN bit is set to 0, this register is not accessible. In this case, writes have no affect and reads return indeterminate values.



NOTE:
X = Value is determined by the state of the corresponding signal pin.

Figure 56. Status Register B

Bit	Description
7:2	**RESERVED**
1	**POWERDOWN STATUS (PD):** This bit reflects the powerdown state of the FDC module. The 82091AA sets PD to 1 when the FDC is in the powerdown state. When PD = 0, the FDC is not in the powerdown state.
0	**IDLE STATUS (IDLE):** This bit reflects the idle state of the FDC module. The 82091AA sets IDLE to 1 when the FDC is in the idle state. When IDLE = 0, the FDC is not in the idle state.

8.1.2 DOR—DIGITAL OUTPUT REGISTER

I/O Address: Base +2h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

The Digital Output Register enables/disables the floppy disk drive motors, selects the disk drives, enables/disables DMA, and provides a FDC module reset. The DOR reset bit and the motor enable bits have to be inactive when the FDC is in powerdown. The DMAGATE# and drive select bits are unchanged. During powerdown, writing to the DOR does not wake up the FDC, except for activating any of the motor enable bits. Setting the motor enable bits to 1 wakes up the FDC.

NOTES:

1. The descriptions in this section for DOR only apply when two-drive support is selected in the FCFG1 Register (FDDQTY = 0). For four-drive support (FDDQTY = 1), refer to Appendix A, FDC Four Drive Support.
2. The drive motor can be enabled separately without selecting the drive. This permits the motor to come up to speed before selecting the drive. Note also that only one drive can be selected at a time. However, the drive should not be selected without enabling the appropriate drive motor via bits[5:4] of this register.

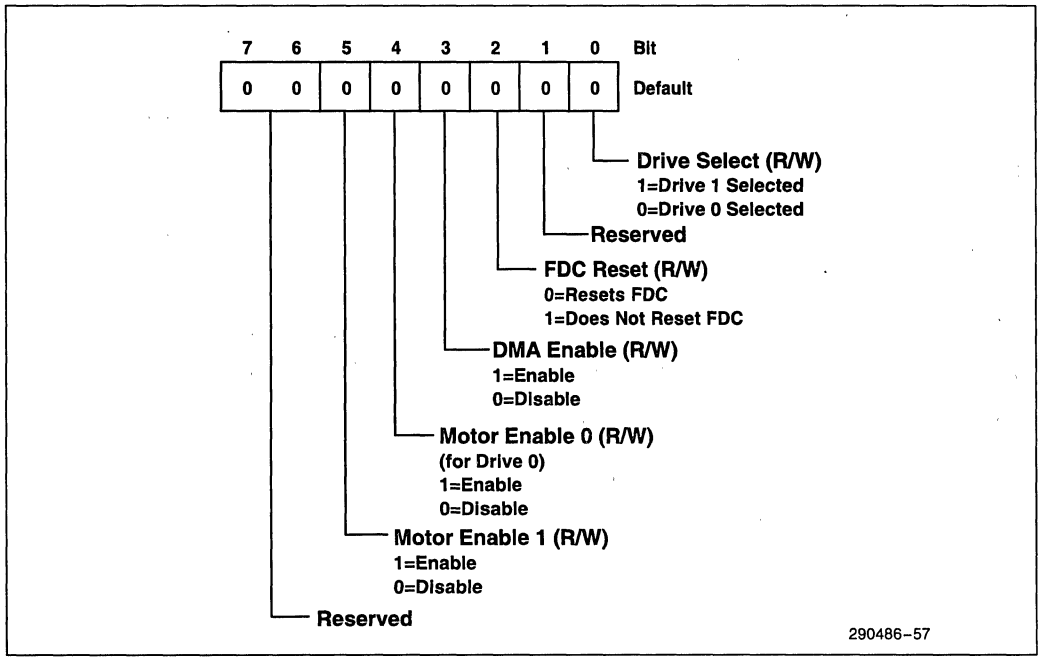


Figure 57. Digital Output Register

Bit	Description
7:6	**RESERVED:** For a two-drive system, these bits are not used and have no affect on FDC operation. For a four drive system, see Appendix A, FDC Four Drive Support.
5	**MOTOR ENABLE 1 (ME1):** This bit controls a motor drive enable signal. ME1 directly controls either the FDME1# signal or FDME0# signal, depending on the state of the BOOTSEL bit in the TDR. When ME1 = 1, the selected motor enable signal (FDME1# or FDME0#) is asserted and when ME1 = 0, the selected motor enable signal is negated.
4	**MOTOR ENABLE 0 (ME0):** This bit controls a motor drive enable signal. ME1 directly controls either the FDME0# signal or FDME1# signal, depending on the state of the BOOTSEL bit in the TDR. When ME0 = 1, the selected motor enable signal (FDME0# or FDME1#) is asserted and when ME0 = 0, the selected motor enable signal is negated.
3	**DMA GATE (DMAGATE):** This bit enables/disables DMA for the FDC. When DMAGATE = 1, DMA for the FDC is enabled. In this mode, FDDREQ, TC, IRQ6, and FDDACK# are enabled. When DMAGATE = 0, DMA for the FDC is disabled. In this mode the IRQ6, and DRQ outputs are tri-stated and the DACK# and TC inputs are disabled to the FDC. Note that the TC input is only disabled to the FDC module. Other functional units in the 82091AA (e.g., parallel port or IDE interface) can still use the TC input signal for DMA activities.
2	**FDC RESET (DORRST):** DORRST is a software reset for the FDC module. When DORRST is set to 0, the basic core of the FDC and the FIFO circuits are cleared conditioned by the LOCK bit in the CONFIGURE Command. This bit is set to 0 by software or a hard reset (RSTDRV asserted). The FDC remains in a reset state until software sets this bit to 1. This bit does not affect the DSR, CCR and other bits of the DOR. DORRST must be held active for at least 0.5 μs at 250 Kbps. This is less than a typical ISA I/O cycle time. Thus, in most systems consecutive writes to this register to toggle this bit allows sufficient time to reset the FDC.
1	**RESERVED:** For a two-drive system, this bit is not used and must be programmed to 0. For a four drive system, see Appendix A, FDC Four Drive Support.
0	**DRIVE SELECT (DS):** This selects the floppy drive by controlling the FDS0# and FDS1# output signals. DS directly controls FDS1 and FDS0 as follows: **Bit 0** — **Output Pin Status** 0 — FDS0# asserted (FDS1 asserted if BOOTSEL = 1) 1 — FDS1# asserted (FDS1 asserted if BOOTSEL = 1)

5

8.1.3 TDR—ENHANCED TAPE DRIVE REGISTER

I/O Address: Base + 3h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

This register allows the user to assign tape support to a particular drive during initialization. Any future references to that drive number automatically invokes tape support. A hardware reset sets all bits in this register to 0 making drive 0 not available for tape support. A software reset via bit 2 of the DOR does not affect this register. Drive 0 is reserved for the floppy boot drive. Bits[7:2] are only available when EREG EN = 1; otherwise the bits are tri-stated. EREG EN is a bit in the POWERDOWN Command.

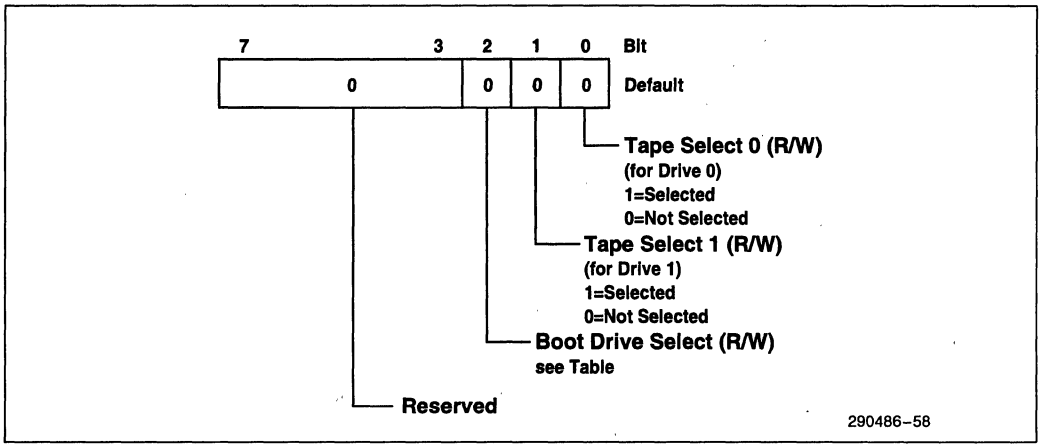


Figure 58. Enhanced Tape Drive Register

Bit	Description
7:3	**RESERVED**
2	**BOOT DRIVE SELECT (BOOTSEL):** The BOOTSEL bit is used to remap the drive selects and motor enables. The functionality is as described below: **BOOTSEL** — **Mapping** 0 — DS0 → FDS0, ME0 → FDME0 (default) DS1 → DS1, ME1 → FDME1 1 — DS0 → DS1, ME0 → FDME1 DS1 → FDS0, ME1 → FDME0 Note that this mapping also applies to a four drive system (FDDQTY = 1 in the FCFG1 Register). In a four drive system, only drive 0 or drive 1 can be selected as the boot drive.
1	**RESERVED:** For a two-drive system, this bit is not used and must be programmed to 0. For a four drive system, see Appendix A, FDC Four Drive Support.
0	**TAPE SELECT (TAPESEL):** This bit is used by software to assign logical drive number 1 to be a tape drive. Other than adjusting precompensation delays for tape support, this bit does not affect the FDC hardware. The bit can be written and read by software as an indication of the tape drive assignment. Drive 0 is not available as a tape drive and is reserved as the floppy disk boot drive. The tape drive assignment is as follows: **Bit 0** — **Drive Selected** 0 — None (all are floppy disk drives) 1 — Drive 1 is a tape drive.

8.1.4 MSR—MAIN STATUS REGISTER

I/O Address: Base +4h
Default Value: 00h
Attribute: Read Only
Size: 8 bits

This read only register provides FDC status information. This information is used by software to control the flow of data to and from the FIFO (accessed via the FDCFIFO Register). The MSR indicates when the FDC is ready to send or receive data through the FIFO. During non-DMA transfers, this register should be read before each byte is transferred to or from the FIFO.

After a hard or soft reset or recovery from a powerdown state, the MSR is available to be read by the host. The register value is 00h until the oscillator circuit has stabilized and the internal registers have been initialized. When the FDC is ready to receive a new command, MSR[7:0] = 80h. The worst case time allowed for the MSR to report 80h (i.e., RQM is set to 1) is 2.5 μs after a hard or soft reset.

Main Status Register is used for controlling command input and result output for all commands. Some example values of the MSR are:

- MSR = 80H; The controller is ready to receive a command.
- MSR = 90H; executing a command or waiting for the host to read status bytes (assume DMA mode).
- MSR = D0H; waiting for the host to write status bytes.

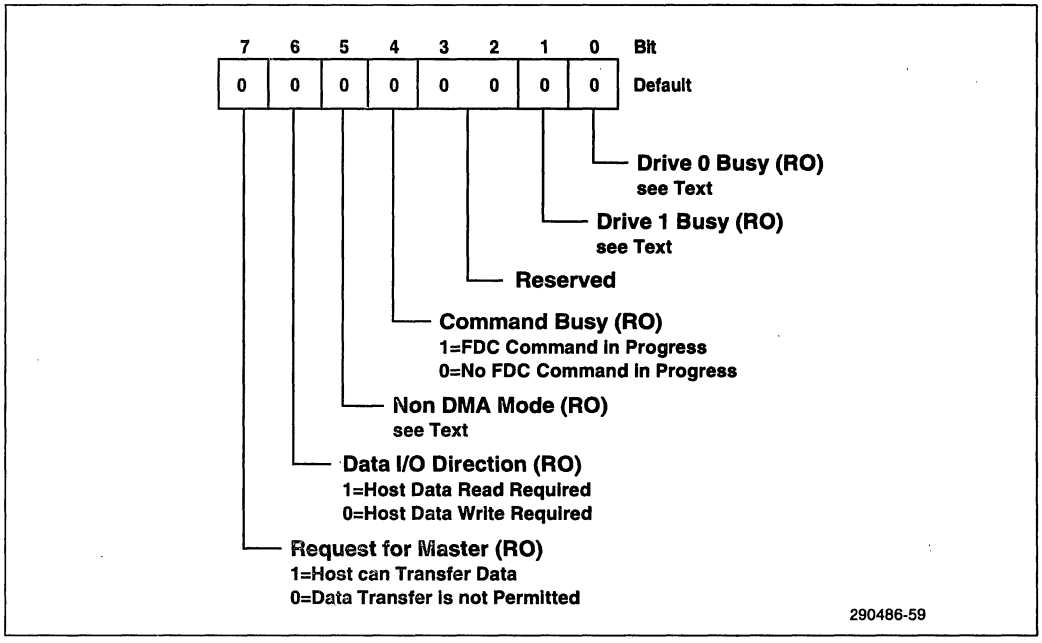


Figure 59. Main Status Register

5

Bit	Description
7	**REQUEST FOR MASTER (RQM):** When RQM = 1, the FDC is ready to send/receive data through the FIFO (FDCFIFO Register). The FDC sets this bit to 0 after a byte transfer and then sets the bit to 1 when it is ready for the next byte. During non-DMA execution phase, RQM indicates the status of IRQ6.
6	**DIRECTION I/O (DIO):** When RQM = 1, DIO indicates the direction of a data transfer. When DIO = 1, the FDC is requesting a read of the FDCFIFO. When DIO = 0, the FDC is requesting a write to the FDCFIFO.
5	**NON-DMA (NONDMA):** Non-DMA mode is selected via the SPECIFY Command. In this mode, the FDC sets this bit to a 1 during the execution phase of a command. This bit is for polled data transfers and helps differentiate between the data transfer phase and the reading of result bytes.
4	**COMMAND BUSY (CMDBUSY):** CMDBUSY indicates when a command is in progress. When the first byte of the command phase is written, the FDC sets this bit to 1. CMDBUSY is set to 0 after the last byte of the result phase is read. If there is no result phase (e.g., SEEK or RECALIBRATE Commands), CMDBUSY is set to 0 after the last command byte is written.
3:2	**RESERVED:** For a two-drive system, these bits are not used and must be programmed to 0. For a four drive system, see Appendix A, FDC Four Drive Support.
1	**DRIVE 1 BUSY (DRV1BUSY):** The FDC module sets this bit to 1 after the last byte of the command phase of a SEEK or RECALIBRATE Command is issued for drive 1. This bit is set to 0 after the host reads the first byte in the result phase of the SENSE INTERRUPT Command for this drive.
0	**DRIVE 0 BUSY (DRV0BUSY):** The FDC module sets this bit to 1 after the last byte of the command phase of a SEEK or RECALIBRATE Command is issued for drive 0. This bit is set to 0 after the host reads the first byte in the result phase of the SENSE INTERRUPT Command for this drive.

8.1.5 DSR—DATA RATE SELECT REGISTER

I/O Address: Base +4h
Default Value: 02h
Attribute: Write Only
Size: 8 bits

The DSR selects the data rate, amount of write precompenstion, invokes direct powerdown, and invokes a FDC software reset. This write only register ensures backward compatibility with the Intel series of floppy disk controllers. Changing the data rate changes the timings of the drive control signals. To ensure that drive timings are not violated when changing data rates, choose a drive timing such that the fastest data rate will not violate the timing.

In the default state, the PDOSC bit is low and the oscillator is powered up. When this bit is programmed to a 1, the oscillator is shut off. Hardware reset sets this bit to a 0. Neither of the software resets (via DOR or DSR) have any effect on this bit. Note that PDOSC should only be set to a 1 when the FDC module is in the powerdown state. Otherwise, the FDC will not function correctly and must be hardware reset once the oscillator has turned back on and stabilized. Setting the PDOSC bit has no effect on the clock input to the FDC (the X1 pin). The clock input is separately disabled when the part is powered down. The Save Command checks the status of PDOSC. However the Restore Command will not restore this bit to a 1.

Software resets do not affect the DRATE or PRECOMP bits.

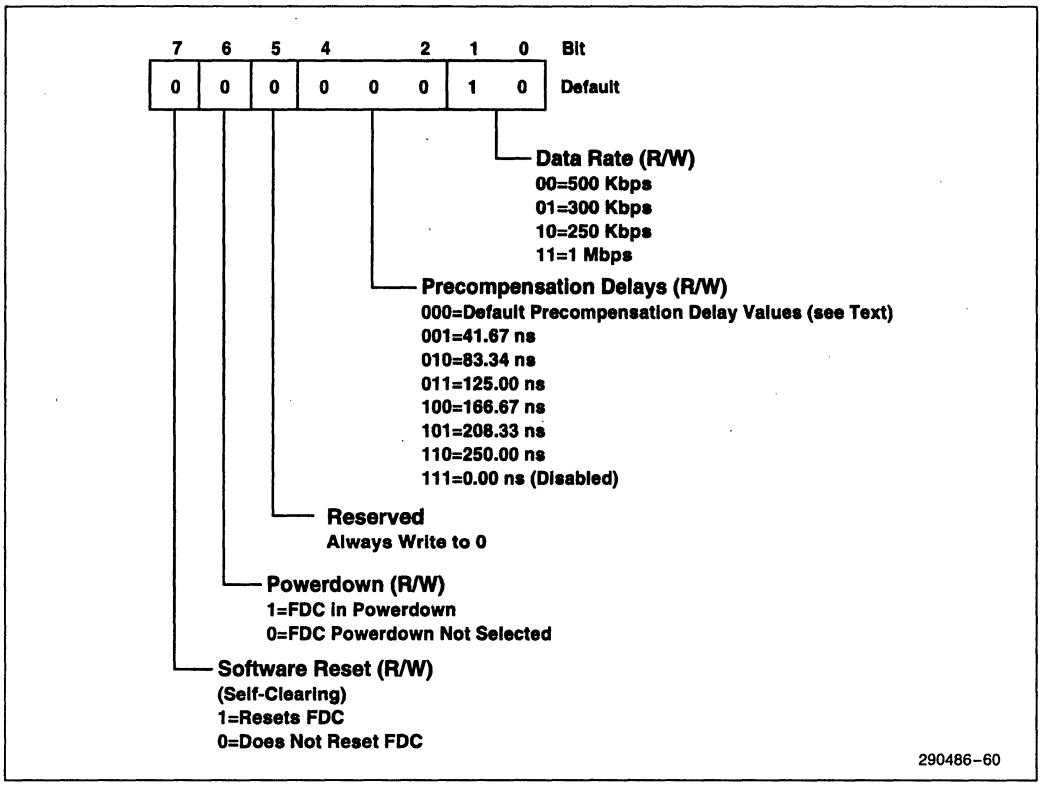


Figure 60. Data Rate Select Register

5

Bit	Description
7	**SOFTWARE RESET (DSRRST):** DSRRST operates the same as the DORRST bit in the DOR, except that this bit is self clearing.
6	**POWERDOWN (FPD):** FPD provides direct powerdown for the FDC module. When FPD = 1, the FDC module enters the powerdown state, regardless of the state of the module. The FDC module is internally reset and then put into powerdown. No status is saved and any operation in progress is aborted. A hardware or software reset causes the 82091AA to exit the FDC module powerdown state.
5	**RESERVED**
4:2	**PRECOMPENSATION (PRECOMP):** Bits[4:2] adjusts the WRDATA output to the disk to compensate for magnetic media phenomena known as bit shifting. The data patterns that are susceptible to bit shifting are well understood and the FDC compensates the data pattern as it is written to the disk. The amount of precompensation depends on the drive and media but in most cases the default value is acceptable. The FDC module starts pre-compensating the data pattern starting on Track 0. The CONFIGURE Command can change the track where pre-compensating originates. **Bits[4:2] — Precompensation Delays (ns)** 0 0 0 — Default mode 0 0 1 — 41.67 0 1 0 — 83.34 0 1 1 — 125.00 1 0 0 — 166.67 1 0 1 — 208.33 1 1 0 — 250 1 1 1 — 0.00 (disabled) The default precompensation delay mode provides the following delays: **Data Rate — Default Precompensation Delays (ns)** 1 Mbps — 41.67 0.5 Mbps — 125.00 0.3 Mbps — 125.00 0.25 Mbps — 125.00
1:0	**DATA RATE SELECT (DRATESEL):** DRATESEL[1:0] select one of the four data rates as listed below. The default value is 250 Kbps. **Bits[1:0] — Date Rate** 1 1 — 1 Mbps 0 0 — 500 Kbps 0 1 — 300 Kbps 1 0 — 250 Kbps - default

8.1.6 FDCFIFO—FDC FIFO (DATA)

I/O Address: Base +5h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

All command parameter information and disk data transfers go through the 16-byte FIFO. The FIFO has programmable threshold values. Data transfers are governed by the RQM and DIO bits in the MSR. At the start of a command, the FIFO action is always disabled and command parameters must be sent based upon the RQM and DIO bit settings. At the start of the command execution phase, the FDC clears the FIFO of any data to ensure that invalid data is not transferred. An overrun or underrun will terminate the current command and the transfer of data. Disk writes complete the current sector by generating a 00 pattern and valid CRC.

The FIFO defaults to an 8272A compatible mode after a hardware reset (via RSTDRV pin). Software resets (via DOR or DSR) can also place the FDC into 8272A compatible mode, if the LOCK bit is set to 0 (see the definition of the LOCK bit) maintaining PC-AT hardware compatibility. The default values can be changed through the CONFIGURE Command (enable full FIFO operation with threshold control). The FIFO provides the system a larger DMA latency without causing a disk error. The following table gives several examples of the delays with a FIFO. The data is based upon the formula: Threshold# × 1/DATA RATE × 8 − 1.5 μs = DELAY.

FIFO Threshold	Maximum Service Delay (1 Mbps Data Rate)	Maximum Delay to Servicing at 500 Kbps Data Rate
1 byte	1 × 8 μs − 1.5 μs = 6.5 μs	1 × 16 μs − 1.5 μs = 14.5 μs
2 bytes	2 × 8 μs − 1.5 μs = 14.5 μs	2 × 16 μs − 1.5 μs = 30.5 μs
8 bytes	8 × 8 μs − 1.5 μs = 62.5 μs	8 × 16 μs − 1.5 μs = 126.5 μs
15 bytes	15 × 8 μs − 1.5 μs = 118.5 μs	15 × 16 μs − 1.5 μs = 238.5 μs

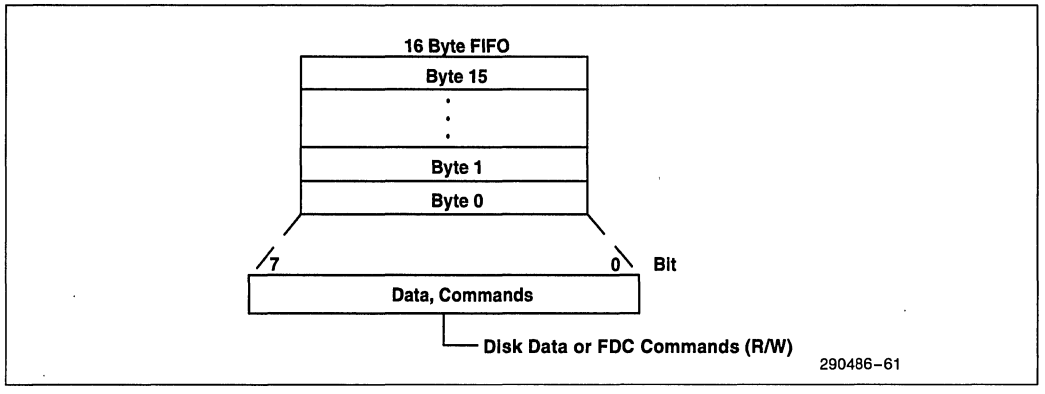


Figure 61. FDC FIFO

Bit	Description
7:0	**FIFO DATA:** Bits[7:0] correspond to SD[7:0].

8.1.7 DIR—DIGITAL INPUT REGISTER

I/O Address: Base +7h
Default Value: 00h
Attribute: Read Only
Size: 8 bits

This register is read only in all modes. In PC-AT mode only bit 7 is driven and all other bits remain tri-stated.

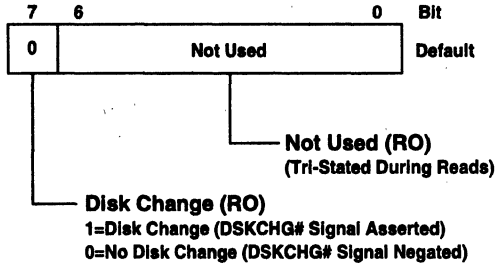


Figure 62. Digital Input Register

Bit	Description
7	**DISK CHANGE (DSKCHG):** This bit monitors a disk change in the floppy disk drive. DSKCHG is set to a 1 when the DSKCHG# signal on the floppy interface is asserted. DSKCHG is set to a 0 when the DSKCHG# signal on the floppy interface is negated. During powerdown, this bit is invalid.
6:0	**NOT USED:** These bits are tri-stated during a read.

8.1.8 CCR—CONFIGURATION CONTROL REGISTER

I/O Address: Base +7h
Default Value: 02h
Attribute: Write Only
Size: 8 bits

This register sets the data rate.

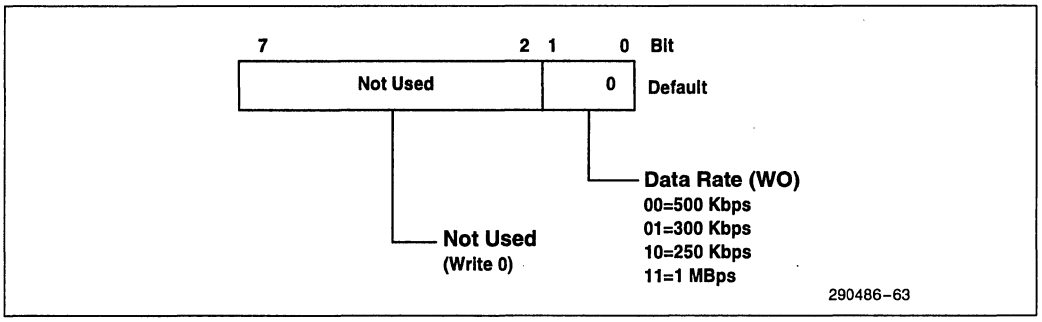


Figure 63. Configuration Control Register

5

8.2 Reset

There are four sources of FDC reset—a hard reset via the RSTDRV signal and three software resets (via the FCFG2, DOR, and DSR Registers). At the end of the reset, the FDC comes out of the powerdown state. Note that the DOR reset condition remains in effect until software programs the DORRST bit to 1 in the DOR. All operations are terminated and the FDC enters an idle state. Invoking a reset while a disk write activity is in progress will corrupt the data and CRC. On exiting the reset state, various internal registers are cleared, and the FDC waits for a new command. Drive polling will start unless disabled by a new CONFIGURE Command.

8.2.1 HARD RESET AND CONFIGURATION REGISTER RESET

A hard reset (asserting RSTDRV) and a software reset through the FCFG2 Registers have the same affect on the FDC. These resets clear all FDC registers, except those programmed by the SPECIFY command. The DOR reset bit is enabled and must be set to 0 by the host to exit the reset state.

8.2.2 DOR RESET vs DSR RESET

The DOR and DSR resets are functionally the same. The DSR reset is included to maintain 82072 compatibility. Both reset the 8272 core, which affects drive status information. The FIFO circuits are also reset if the LOCK bit is a 0 (see definition of the LOCK bit). The DSR reset is self-clearing (exits the reset state automatically) while the DOR reset remains in the reset state until software writes the DOR reset bit to 0. DOR reset has precedence over the DSR reset. The DOR reset is set automatically when a hard reset or configuration reset occurs. Software must set the DOR reset bit to 0 to exit the reset state.

The AC Specifications gives the minimum amount of time that the DOR reset must be held active. This amount of time that the DOR reset must be held active is dependent upon the data rate. FDC requires that the DOR reset bit must be held active for at least 0.5 μs at 250 Kbps. This is less than a typical ISA I/O cycle time.

8.3 DMA Transfers

DMA transfers are enabled with the SPECIFY Command. When enabled, The FDC initiates DMA transfers by asserting the FDDREQ signal during a data transfer command. The FIFO is enabled directly by asserting FDDACK# and addresses need not be valid.

8.4 Controller Phases

The FDC handles commands in three phases—*command*, *execution* and *result*. Each phase is described in the following sections. When not processing a command, the FDC can be in the *idle*, *drive polling* or *powerdown state*. This section describes the command, execute and result phases.

8.4.1 COMMAND PHASE

After a reset, the FDC enters the command phase and is ready to accept a command from the host. For each of the commands, a defined set of command code bytes and parameter bytes must be written to the FDC (as described in Section 8.8, Command Set Description) before the command phase is complete. These bytes of data must be transferred in the order described.

Before writing to the FDC, the host must examine the RQM and DIO bits of the Main Status Register. RQM must be 1 and DIO must be 0, before command bytes may be written. The FDC sets RQM to 0 after each write cycle and keeps the bit at 0 until the received byte is processed. After processing the byte, the FDC sets RQM to 1 again to request the next parameter byte of the command, unless an illegal command condition is detected. After the last parameter byte is received, RQM remains 0, and the FDC automatically enters the next phase (execution or result phase) as defined by the command definition.

The FIFO is disabled during the command phase to retain compatibility with the 8272A, and to provide for the proper handling of the Invalid Command condition.

8.4.2 EXECUTION PHASE

The following paragraphs detail the operation of the FIFO flow control. In these descriptions, threshold is defined as the number of bytes available to the FDC when service is requested from the host, and ranges from 1 to 16. The FIFOTHR parameter, which the user programs, is one less and ranges from 0 to 15.

A low threshold value (e.g., 2) results in longer periods of time between service requests but requires faster servicing of the request for both read and write cases. The host reads (writes) from (to) the FIFO until empty (full), then the transfer request goes inactive. The host must be very responsive to the service request. This is the desired case for use with a "fast" system.

A high value of threshold (e.g., 12) is used with a "sluggish" system by affording a long latency period after a service request, but results in more frequent service requests.

8.4.2.1 Non-DMA Mode Transfers from the FIFO to the Host

The IRQ6 pin and RQM bits in the Main Status Register are activated when the FIFO contains 16 (or set threshold) bytes, or the last bytes of a full sector transfer have been placed in the FIFO. The IRQ6 pin can be used for interrupt driven systems and RQM can be used for polled sytems. The host must respond to the request by reading data from the FIFO. This process is repeated until the last byte is transferred out of the FIFO, then FDC negates the IRQ6 pin and RQM bit.

8.4.2.2 Non-DMA Mode Transfers from the Host to the FIFO

The IRQ6 pin and RQM bit in the Main Status Register are activated upon entering the execution phase of data transfer commands. The host must respond to the request by writing data into the FIFO. The IRQ6 pin and RQM bit remain true until the FIFO becomes full. They are set true again when the FIFO has (threshold) bytes remaining in the FIFO. The IRQ6 pin is also negated if TC and DACK# both go inactive. The FDC enters the result phase after the last byte is taken by the FDC from the FIFO (i.e. FIFO empty condition).

8.4.2.3 DMA Mode Transfers from the FIFO to the Host

The FDC asserts the FDDREQ signal when the FIFO contains 16 (or set threshold) bytes or the last byte of a full sector transfer has been placed in the FIFO. The DMA controller must respond to the request by reading data from the FIFO. The FDC negates FDDREQ when the FIFO is empty. FDDREQ is negated after FDDACK# is asserted for the last byte of a data transfer (or on the active edge of RD#, on the last byte, if no edge is present on FDDACK#).

NOTE:
FDDACK# and TC must overlap for at least 50 ns for proper functionality. A data underrun may occur if FDDREQ is not removed in time to prevent an unwanted cycle.

8.4.2.4 DMA Mode Transfers from the Host to the FIFO

The FDC asserts FDDREQ when entering the execution phase of data transfer commands. The DMA controller must respond by asserting FDDACK# and WR# signals and placing data in the FIFO. FDDREQ remains asserted until the FIFO becomes full. FDDREQ is again asserted when the FIFO has (threshold) bytes remaining in the FIFO. The FDC also negates the FDDREQ when the FIFO becomes empty (qualified by DACK# and TC overlapping by 50 ns) indicating that no more data is required. FDDREQ is negated after FDDACK# is asserted for the last byte of a data transfer (or on the active edge of WR# of the last byte, if no edge is present on DACK#). A data overrun may occur if FDDREQ is not removed in time to prevent an unwanted cycle.

8.4.3 DATA TRANSFER TERMINATION

The FDC supports terminal count explicitly through the TC signal and implicitly through the underrun/overrun and end-of-track (EOT) functions. For full sector transfers, the EOT parameter can define the last sector to be transferred in a single or multi-sector transfer. If the last sector to be transferred is a partial sector, the host can stop transferring the data in mid-sector and the FDC will continue to complete the sector as if a hardware TC was received. The only difference between these implicit functions and TC is that they return "abnormal termination" result status. Such status indications can be ignored if they were expected.

NOTE:

When the host is sending data to the FIFO, the internal sector count will be complete when the FDC reads the last byte from its side of the FIFO. There may be a delay in the removal of the transfer request signal of up to the time taken for the FDC to read the last 16 bytes from the FIFO. The host must be able to tolerate this. In a DMA system, FDDREQ is removed (negated) as soon as TC is received indicating the termination of the transfer. The reception of TC also generates an interrupt on IRQ6. However, in a non-DMA system the interrupt will not be generated until the FIFO is empty.

The generation of IRQ6 determines the beginning of the result phase. For each of the commands, a defined set of result bytes has to be read from the FDC before the result phase is complete (refer to Section 8.5, Command Set/Descriptions). These bytes of data must be read out for another command to start.

RQM and DIO must both be 1 before the result bytes may be read from the FIFO. After all the result bytes have been read, RQM=1, DIO=0, and CMDBUSY=0 in the MSR. This indicates that the FDC is ready to accept the next command.

8.5 Command Set/Descriptions

Commands can be written whenever the FDC is in the command phase. Each command has a unique set of needed parameters and status results. The FDC checks to see that the first byte is a valid command and, if valid, proceeds with the command. If it was invalid, the next time the RQM bit in the MSR register is 1 the DIO and CB bits will also be 1, indicating the FIFO must be read. A result byte of 80h will be read out of the FIFO, indicating an invalid command was issued. After reading the result byte from the FIFO, the FDC returns to the command phase. Table 23 shows the FDC Command set.

Table 24. FDC Command Set

Phase	R/W	Data Bus D7	D6	D5	D4	D3	D2	D1	D0	Remarks
Read Data										
Command	W	MT	MFM	SK	0	0	1	1	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W	.....			C		.....			Sector ID
	W	.....			H		.....			Information Prior to
	W	.....			R		.....			Command
										Execution
	W	.....			N		.....			
	W	.....			EOT		.....			
	W	.....			GPL		.....			
	W	.....			DTL		.....			
Execution										Data Transfer Between the FDD and System
Result	R	.....			ST 0		.....			Status Information
	R	.....			ST 1		.....			After Command
	R	.....			ST 2		.....			Execution
	R	.....			C		.....			
	R	.....			H		.....			Sector ID
	R	.....			R		.....			Information After
	R	.....			N		.....			Command
										Execution
Read Deleted Data										
Command	W	MT	MFM	SK	0	1	1	0	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W	.....			C		.....			Sector ID
	W	.....			H		.....			Information Prior to
	W	.....			R		.....			Command
										Execution
	W	.....			N		.....			
	W	.....			EOT		.....			
	W	.....			GPL		.....			
	W	.....			DTL		.....			
Execution										Data Transfer Between the FDD and System
Result	R	.....			ST 0		.....			Status Information
	R	.....			ST 1		.....			After Command
	R	.....			ST 2		.....			Execution
	R	.....			C		.....			
	R	.....			H		.....			Sector ID
	R	.....			R		.....			Information After
	R	.....			N		.....			Command
										Execution

5

Table 24. FDC Command Set (Continued)

Phase	R/W	Data Bus								Remarks
		D7	D6	D5	D4	D3	D2	D1	D0	
Write Data										
Command	W	MT	MFM	0	0	0	1	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W	.....			C		.....			Sector ID
	W	.....			H		.....			Information Prior to
	W	.....			R					Command
										Execution
	W	.....			N		.....			
	W	.....			EOT		.....			
	W	.....			GPL		.....			
	W	.....			DTL		.....			
Execution										Data Transfer Between the FDD and System
Result	R	.....			ST 0		.....			Status Information
	R	.....			ST 1		.....			After Command
	R	.....			ST 2		.....			Execution
	R	.....			C		.....			
	R	.....			H		.....			Sector ID
	R	.....			R		.....			Information After
	R	.....			N		.....			Command
										Execution
Write Deleted Data										
Command	W	MT	MFM	0	0	1	0	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W	.....			C		.....			Sector ID
	W	.....			H		.....			Information Prior to
	W	.....			R		.....			Command
										Execution
	W	.....			N		.....			
	W	.....			EOT		.....			
	W	.....			GPL		.....			
	W	.....			DTL		.....			
Execution										Data Transfer Between the FDD and System
Result	R	.....			ST 0		.....			Status Information
	R	.....			ST 1		.....			After Command
	R	.....			ST 2		.....			Execution
	R	.....			C		.....			
	R	.....			H		.....			Sector ID
	R	.....			R		.....			Information After
	R	.....			N		.....			Command
										Execution

Table 24. FDC Command Set (Continued)

Phase	R/W	Data Bus D7	D6	D5	D4		D3	D2	D1	D0	Remarks
Read Track											
Command	W	0	MFM	0	0		0	0	1	0	Command Codes
	W	0	0	0	0		0	HDS	DS1	DS0	
	W	.....				C		.....			Sector ID
	W	.....				H		.....			Information Prior to
	W	.....				R		.....			Command
											Execution
	W	.....				N		.....			
	W	.....				EOT					
	W	.....				GPL		.....			
	W	.....				DTL		.....			
Execution											Data Transfer Between the FDD and System. FDC Reads All Sectors From Index Hole to EOT
Result	R	.....				ST 0		.....			Status Information
	R	.....				ST 1		.....			After Command
	R	.....				ST 2		.....			Execution
	R	.....				C		.....			
	R	.....				H		.....			Sector ID
	R	.....				R		.....			Information After
	R	.....				N		.....			Command
											Execution
Verify											
Command	W	MT	MFM	SK	1		0	1	1	0	Command Codes
	W	EC	0	0	0		0	HDS	DS1	DS0	
	W	.....				C		.....			Sector ID
	W	.....				H		.....			Information Prior to
	W	.....				R		.....			Command
											Execution
	W	.....				N		.....			
	W	.....				EOT		.....			
	W	.....				GPL		.....			
	W	.....				DTL/SC		.....			
Execution											Data Transfer Between the FDD and System
Result	R	.....				ST 0		.....			Status Information
	R	.....				ST 1		.....			After Command
	R	.....				ST 2		.....			Execution
	R	.....				C		.....			
	R	.....				H		.....			Sector ID
	R	.....				R		.....			Information After
	R	.....				N		.....			Command
											Execution

5

Table 24. FDC Command Set (Continued)

Phase	R/W	Data Bus D7	D6	D5	D4		D3	D2	D1	D0	Remarks
Version											
Command	W	0	0	0	1		0	0	0	0	Command Codes
Result	W	1	0	0	1		0	0	0	0	Enhanced Controller
Format Track											
Command	W	0	MFM	0	0		1	1	0	1	Command Codes
	W	0	0	0	0		0	HDS	DS1	DS0	
	W	.....				N		.....			Bytes/Sector
	W	.....				SC		.....			Sector/Cylinder
	W	.....				GPL		.....			Gap 3
	W	.....				D		.....			Filler Byte
Execution											
For Each	W	.....				C		.....			
Sector	W	.....				H		.....			Input Sector
Repeat:	W	.....				R		.....			Parameters
	W	.....				N		.....			
											FDC Formats an Entire Cylinder
Result	R	.....				ST 0		.....			Status Information
	R	.....				ST 1		.....			after Command
	R	.....				ST 2		.....			Execution
	R	.....				Undefined		.....			
	R	.....				Undefined		.....			
	R	.....				Undefined		.....			
	R	.....				Undefined		.....			
Scan Equal											
Command	W	MT	MFM	SK	1		0	0	0	0	Command Codes
	W	0	0	0	0		0	HDS	DS1	DS0	
	W	.....				C		.....			Sector ID
	W	.....				H		.....			Information Prior
	W	.....				R		.....			to Command
	W	.....				N		.....			Execution
	W	.....				EOT		.....			
	W	.....				GPL		.....			
	W	.....				STP		.....			
Execution											Data Compared Between the FDD and Main-System
Result	R	.....				ST 0		.....			Status Information
	R	.....				ST 1		.....			After Command
	R	.....				ST 2		.....			Execution
	R	.....				C		.....			
	R	.....				H		.....			Sector ID
	R	.....				R		.....			Information After
	R	.....				N		.....			Command Execution

Table 24. FDC Command Set (Continued)

Phase	R/W	Data Bus D7	D6	D5	D4	D3	D2	D1	D0	Remarks
Scan Low or Equal										
Command	W	MT	MFM	SK	1	1	0	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W	.....			C		.....			Sector ID
	W	.....			H		.....			Information Prior
	W	.....			R		.....			to Command
	W	.....			N		.....			Execution
	W	.....			EOT		.....			
	W	.....			GPL		.....			
	W	.....			STP		.....			
Execution										Data Compared Between the FDD and Main-System
Result	R	.....			ST 0		.....			Status Information
	R	.....			ST 1		.....			After Command
	R	.....			ST 2		.....			Execution
	R	.....			C		.....			
	R	.....			H		.....			Sector ID
	R	.....			R		.....			Information After
	R	.....			N		.....			Command Execution
Scan High or Equal										
Command	W	MT	MFM	SK	1	1	1	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W	.....			C		.....			Sector ID
	W	.....			H		.....			Information Prior
	W	.....			R		.....			to Command
	W	.....			N		.....			Execution
	W	.....			EOT		.....			
	W	.....			GPL		.....			
	W	.....			STP		.....			
Execution										Data Compared Between the FDD and Main-System
Result	R	.....			ST 0		.....			Status Information
	R	.....			ST 1		.....			After Command
	R	.....			ST 2		.....			Execution
	R	.....			C		.....			
	R	.....			H		.....			Sector ID
	R	.....			R		.....			Information After
	R	.....			N		.....			Command Execution
Recalibrate										
Command	W	0	0	0	0	0	1	1	1	Command Codes
	W	0	0	0	0	0	0	DS0	DS1	Enhanced Controller
Execution										Head Retracted to Track 0 Interrupt

5

Table 24. FDC Command Set (Continued)

Phase	R/W	Data Bus								Remarks
		D7	D6	D5	D4	D3	D2	D1	D0	
Sense Interrupt Status										
Command	W	0	0	0	0	1	0	0	0	Command Codes
Result	R	.....				ST 0		.....		Status Information at the End of Each Seek Operation
	R	.....				PCN		.....		
Specify										
Command	W	0	0	0	0	0	0	1	1	Command Codes
	W	.....		SRT	.....	.....		HUT	.....	
	W	.....			HLT	.....			ND	
Sense Drive Status										
Command	W	0	0	0	0	0	1	0	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
Result	R	.....				ST 3		.....		Status Information About FDD
Drive Specification Command										
Command	W	1	0	0	0	1	1	1	0	Command Code
	W	0	FD1	FD0	PTS	DRT1	DRT0	DT1	DT0	
	:	:	:	:	:	:	:	:	:	0–4 bytes issued
	W	DN	NRP	0	0	0	0	0	0	
Result	R	0	0	0	PTS	DRT1	DRT0	DT1	DT0	Drive 0
	R	0	0	0	PTS	DRT1	DRT0	DT1	DT0	Drive 1
	R	0	0	0	0	0	0	0	0	RSVD
	R	0	0	0	0	0	0	0	0	RSVD
Seek										
Command	W	0	0	0	0	1	1	1	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
Execution	W	.....				NCN		.....		Head is Positioned Over Proper Cylinder on Diskette
Configure										
Command	W	0	0	0	1	0	0	1	1	Command Code
	W	0	0	0	0	0	0	0	0	
	W	0	EIA	EFIFO	POLL	.....	FIFOTHR			
	W	.....				PRETRK		.....		
Relative Seek										
Command	W	1	DIR#	0	0	1	1	1	1	Command Code
	W	0	0	0	0	0	HDS	DS1	DS0	
	W	.....				RCN		.....		

Table 24. FDC Command Set (Continued)

Phase	R/W	Data Bus D7	D6	D5	D4	D3	D2	D1	D0	Remarks
DUMPREG										
Command Execution	W	0	0	0	0	1	1	1	0	Note: Registers placed in FIFO
Result	R	.....				PCN-Drive 0	.....			
	R	.....				PCN-Drive 1	.....			
	R	.....				PCN-Drive 2	.....			
	R	.....				PCN-Drive 3	.....			
	R	.....	SRT	.....		.....	HUT	.....		
	R	.....		HLT	.....				ND	
		.....				SC/EOT	.....			
	R	LOCK	0	0	0	D1	D0	GAP	WGATE	
	R	0	EIS	EFIFO		POLL	FIFOTHR	.....		
	R	.....				PRETRK	.....			
Read ID										
Command	W	0	MFM	0	0	1	0	1	0	Commands
	W	0	0	0	0	0	HDS	DS1	DS0	The First Correct ID Information on the Cylinder is Stored in Data Register
Result	R	.....				ST 0	.....			Status Information
	R	.....				ST 1	.....			After Command
	R	.....				ST 2	.....			Execution
	R	.....				C	.....			
	R	.....				H	.....			Disk Status After the
	R	.....				R	.....			Command has
	R	.....				N	.....			Completed
Perpendicular Mode										
Command	W	0	0	0	1	0	0	1	0	Command Codes
	W	OW	0	0	0	D1	D0	GAP	WGATE	
Lock										
Command	W	LOCK	0	0	1	0	1	0	0	Command Codes
Result	R	0	0	0	LOCK	0	0	0	0	
Part ID										
Command	W	0	0	0	1	1	0	0	0	Command Code
Result	R	0	0	0	.....	Stepping	.....		1	Part ID Number
Powerdown Mode										
Command	W	0	0	0	1	0	1	1	1	Command Code
	W	0	0	EREG EN	0	0	FDI TRI	MIN DLY	AUTO PD	
Result	R	0	0	EREG EN	0	0	FDI TRI	MIN DLY	AUTO PD	

5

Table 24. FDC Command Set (Continued)

Phase	R/W	Data Bus								Remarks
		D7	D6	D5	D4	D3	D2	D1	D0	
Option										
Command	W	0	0	1	1	0	0	1	1	Command Code
	W	.....		RSVD		.....			ISO	
Save										
Command	W	0	0	1	0	1	1	1	0	Command Code
Result	R	RSVD	RSVD	PD OSC	PC2	PC1	PC0	DRATE1	DRATE0	Save Information to Reprogram the FDC
		0	0	0	0	0	0	0	ISO	
	R	.....				PCN-Drive 0			.....	
	R	.....				PCN-Drive 1			.....	
	R	.....				PCN-Drive 2			.....	
	R	.....				PCN-Drive 3			.....	
	R	.....	SRT	.....		.....	HUT		.....	
	R	.....				HLT	.....		ND	
	R	.....				SC/EOT			.....	
	R	LOCK	0	0	0	D1	D0	GAP	WGATE	
	R	0	EIS	EFIFO	POLL	—	.....	FIFOTHR	.....	
	R	.....				PRETRK			.....	
	R	0	0	EREG EN	0	RSVD	FDI TRI	MIN DLY	AUTO PD	
	R	.....				DISK/STATUS			.....	
	R	.....				RSVD			.....	
	R	.....				RSVD			.....	
Restore										
Command	W	0	1	0	0	1	1	1	0	Command Code
	W	0	0	0	PC2	PC1	PC0	DRATE1	DRATE0	Restore Original
	W	0	0	0	0	0	0	0	ISO	Register Status
	W	.....				PCN-Drive 0			.....	
	W	.....				PCN-Drive 1			.....	
	W	.....				PCN-Drive 2			.....	
	W	.....				PCN-Drive 3			.....	
	W	.....	SRT	.....		.....	HUT		.....	
	W	.....			HLT	.....			ND	
	W	.....				SC/EOT			.....	
	W	LOCK	0	0	0	D1	D0	GAP	WGATE	
	W	0	EIS	EFIFO	POLL	.....	FIFOTHR		.....	
	W	.....				PRETRK			.....	
	W	0	0	EREG EN	0	RSVD	FDI TRI	MIN DLY	AUTO PD	
	W	.....				DISK/STATUS			.....	
	W	.....				RSVD			.....	
	W	.....				RSVD			.....	

Table 24. FDC Command Set (Continued)

Phase	R/W	Data Bus D7	D6	D5	D4	D3	D2	D1	D0	Remarks
Format and Write										
Command	W	1	MFM	1	0	1	1	0	1	Command Code
	W	0	0	0	0	0	HDS	DS1	DS0	
	W	.....			N		.....			
	W	.....			SC		.....			
	W	.....			GPL		.....			
	W	.....			D		.....			
Execution	W	.....			C		.....			Input
repeated	W	.....			H		.....			Sector
for each	W	.....			R		.....			Parameters
sector	W	.....			N		.....			
	W	Data Transfer Of N Bytes								FDC Formats and Writes Entire Track
Result	R	.....			ST 0		.....			
	R	.....			ST 1		.....			
	R	.....			ST 2		.....			
	R	.....			Undefined		.....			
	R	.....			Undefined		.....			
	R	.....			Undefined		.....			
	R	.....			Undefined		.....			
Invalid										
Command	W	.....			Invalid Codes		.....			Invalid Command Codes (Noop—FDC goes into Standby State)
Result	R	.....			ST 0		.....			ST 0 = 80

5

Parameter Abbreviations

Symbol	Description
AUTO PD	**AUTO POWERDOWN CONTROL:** When AUTO PD=0, automatic powerdown is disabled. When AUTO PD=1, automatic powerdown is enabled.
C	**CYLINDER ADDRESS:** The currently selected cylinder address, 0 to 255.
D0, D1	**DRIVE SELECT 0-1:** Designates which drives are Perpendicular drives. A 1 indicates Perpendicular drive.
D	**DATA PATTERN:** The pattern to be written in each sector data field during formatting.
DN	**DONE:** This bit indicates that this is the last byte of the drive specification command. The FDC checks to see if this bit is 1 or 0. When DN=0, the FDC expects more bytes. DN=0 FDC expects more subsequent bytes. DN=1 Terminates the command phase and enters the results phase. An additional benefit is that by setting this bit to 1, a direct check of the current drive specifications can be done.
DIR#	**DIRECTION CONTROL:** When DIR#=0, the head steps out from the spindle during a relative seek. When DIR#=1, the head steps in toward the spindle.
DS0, DS1	**DISK DRIVE SELECT:**

DS1	DS0	Drive Slot
0	0	drive 0
0	1	drive 1
1	0	drive 2*
1	1	drive 3*

*Available when FDDQTY=1 in the FCFG1 Register (see Appendix A, FDC Four Drive Support)

DTL	**SPECIAL SECTOR SIZE:** By setting N to zero (00), DTL may be used to control the number of bytes transferred in disk read/write commands. The sector size (N=0) is set to 128. If the actual sector (on the diskette) is larger than DTL, the remainder of the actual sector is read but is not passed to the host during read commands; during write commands, the remainder of the actual sector is written with all zero bytes. The CRC check code is calculated with the actual sector. When N is not zero, DTL has no meaning and should be set to FFh.
DRATE[0:1]	**DATA RATE:** Data rate values from the DSR register.

Symbol **Description**

DRT0, DRT1 **DATA RATE TABLE SELECT:** These two bits select between the different data rate tables. The default is the conventional table. These also provide mapping of the data rates selected in the DSR and CCR. The table below shows this.

Bits in DSR					
DRT1	**DRT0**	**DRATE1**	**DRATE0**	**Data Rate**	**Operation**
0	0	1	1	1 Mbps	Default
		0	0	500 Kbps	
		0	1	300 Kbps	
		1	0	250 Kbps	
0	1	RSVD	RSVD	RSVD	RSVD
1	0	RSVD	RSVD	RSVD	RSVD
1	1	1	1	1 Mbps	Perpendicular mode FDDs
		0	0	500 Kbps	
		0	1	Illegal	
		1	0	250 Kbps	

DT0,DT1 **DRIVE DENSITY SELECT TYPE:** These bits select the outputs on DRVDEN0 and DRVDEN1 (see DRIVE SPECIFICATION Command).

EC **ENABLE COUNT:** When EC=1, the DTL parameter of the Verify Command becomes SC (Number of sectors per track).

EFIFO **Enable FIFO:** When EFIFO=0, the FIFO is enabled. EFIFO=1 puts the FDC in the 8272A compatible mode where the FIFO is disabled.

EIS **ENABLE IMPLIED SEEK:** When EIS=1, a seek operation is performed before executing any read or write command that requires the C parameter in the command phase. EIS=0 disables the implied seek.

EOT **END OF TRACK:** The final sector number of the current track.

EREG EN **ENHANCED REGISTER ENABLE:** When EREG EN=1, the TDR register is extended and SRB is made visible to the user. When EREG EN=0, the standard registers are used.

FDI TRI **FLOPPY DRIVE INTERFACE TRI-STATE:** When FDI TRI=0, the output pins of the floppy disk drive interface are tri-stated. This is also the default state. When FDI TRI=1, the floppy disk drive interface remains unchanged.

Symbol	Description
FD0, FD1	**FLOPPY DRIVE SELECT:** These two bits select which physical drive is being specified. The FDn corresponds to FDSn and FDMEn on the floppy drive interface. The drive is selected independent of the BOOTSEL bit in the TDR. Refer to Section 8.1.3, TDR—Enhanced Tape Drive Register, which explains the distinction between physical drives and their virtual mapping as defined by the BOOTSEL bit.

FD1	FD0	Drive slot
0	0	drive 0
1	0	drive 1
0	1	drive 2*
1	1	drive 3*

*Available if the four floppy drive option is selected in the FCFG1 Register.

GAP	**GAP:** Alters Gap 2 length when using Perpendicular Mode.
GPL	**GAP LENGTH:** The gap 3 size. (Gap 3 is the space between sectors excluding the VCO synchronization field).
H/HDS	**HEAD ADDRESS:** Selected head: 0 or 1 (disk side 0 or 1) as encoded in the sector ID field.
HLT	**HEAD LOAD TIME:** The time interval that FDC waits after loading the head and before initiating a read or write operation. Refer to the SPECIFY Command for actual delays.
HUT	**HEAD UNLOAD TIME:** The time interval from the end of the execution phase (of a read or write command) until the head is unloaded. Refer to the SPECIFY Command for actual delays.
ISO	**ISO FORMAT:** When ISO = 1, the ISO format is used for all data transfer commands. When ISO = 0, the normal IBM system 34 and perpendicular is used. The default is ISO = 0.
LOCK	**LOCK:** Lock defines whether EFIFO, FIFOTHR, and PRETRK parameters of the CONFIGURE Command can be reset to their default values by a software reset (Reset made by setting the proper bit in the DSR or DOR registers).
MFM	**MFM MODE:** A one selects the double density (MFM) mode. A zero is reserved.

Symbol	Description
MIN DLY	**MINIMUM POWERUP TIME CONTROL:** This bit is active only if AUTO PD bit is enabled. When MIN DLY = 0, a 10 ms minimum powerup time is assigned and when MIN DLY = 1, a 0.5 sec. minimum powerup time is assigned.
MT	**MULTI-TRACK SELECTOR:** When MT = 1, the multi-track operating mode is selected. In this mode, the FDC treats a complete cylinder, under head 0 and 1, as a single track. The FDC operates as if this expanded track started at the first sector under head 0 and ended at the last sector under head 1. With this flag set, a multitrack read or write operation will automatically continue to the first sector under head 1 when the FDC finishes operating on the last sector under head 0.
N	**SECTOR SIZE CODE:** This specifies the number of bytes in a sector. When N = 00h, the sector size is 128 bytes. The number of bytes transferred is determined by the DTL parameter. Otherwise the sector size is (2 raised to the "N'th" power) times 128. All values up to 07h are allowable. A value of 07h equals a sector size of 16 Kbytes. It is the users responsibility to not select combinations that are not possible with the drive.

N	Sector Size
00	128 bytes
01	256 bytes
02	512 bytes
03	1024
..	...
07	16 Kbytes

NCN	**NEW CYLINDER NUMBER:** The desired cylinder number.
ND	**NON-DMA MODE FLAG:** When ND = 1, the FDC operates in the non-DMA mode. In this mode, the host is interrupted for each data transfer. When ND = 0, the FDC operates in DMA mode and interfaces to a DMA controller by means of the DRQ and DACK# signals.
NRP	**NO RESULTS PHASE:** When NRP = 1, the result phase is skipped. When NRP = 0, the result phase is generated.
OW	**OVERWRITTEN:** The bits denoted D0 and D1 of the PERPENDICULAR MODE Command can only be overwritten when OW = 1.

Symbol	Description
PCN	**PRESENT CYLINDER NUMBER:** The current position of the head at the completion of SENSE INTERRUPT STATUS Command.
PC2,PC1,PC0	**PRECOMPENSATION VALUES:** Precompensation values from the DSR register.
PDOSC	**POWERDOWN OSCILLATOR:** When this bit is set, the internal oscillator is turned off.
PTS	**PRECOMPENSATION TABLE SELECT:** This bit selects whether to enable the precompensation value programmed in the DSR or not. In the default state, the value programmed in DSR will be used. More information regarding the precompensation is available in Section 8.1.5. PTS = 0 DSR programmed precompensation delays PTS = 1 No precompensation delay is selected for the corresponding drive.
POLL	**POLLING DISABLE:** When POLL = 1, the internal polling routine is disabled. When POLL = 0, polling is enabled.
PRETRK	**PRECOMPENSATION START TRACK NUMBER:** Programmable from track 00 to FFh.
R	**SECTOR ADDRESS:** The sector number to be read or written. In multi-sector transfers, this parameter specifies the sector number of the first sector to be read or written.
RCN	**RELATIVE CYLINDER NUMBER:** Relative cylinder offset from present cylinder as used by the RELATIVE SEEK Command.
SC	**NUMBER OF SECTORS:** The number of sectors to be initialized by the FORMAT Command. The number of sectors to be verified during a Verify Command, when EC = 1.
SK	**SKIP FLAG:** When SK = 1, sectors containing a deleted data address mark will automatically be skipped during the execution of a READ DATA Command. If a READ DELETED DATA Command is executed, only sectors with a deleted address mark will be accessed. When SK = 0, the sector is read or written the same as the read and write commands.
SRT	**STEP RATE INTERVAL:** The time interval between step pulses issued by the FDC. Programmable from 0.5 ms to 8 ms, in increments of 0.5 ms at the 1 Mbit data rate. Refer to the SPECIFY Command for actual delays.
ST0-3	**STATUS REGISTERS 0-3.**Registers within the FDC that store status information after a command has been executed. This status information is available to the host during the result phase after command execution.
WGATE	**WRITE GATE:** Write gate alters timing of WE, to allow for pre-erase loads in perpendicular drives.

8.5.1 STATUS REGISTER ENCODING

The contents of these registers are available only through a command sequence.

8.5.1.1 Status Register 0

Bit #	Symbol	Name	Description
7,6	IC	Interrupt Code	00 Normal termination of command. The specified command was properly executed and completed without error. 01 Abnormal termination of command. Command execution was started, but was not successful completed. 10 Invalid command. The requested command could not be executed. 11 Abnormal termination caused by Polling.
5	SE	Seek End	The 82091AA completed a SEEK or RECALIBRATE command, or a READ or WRITE with implied seek command.
4	EC	Equipment Check	The TRK pin failed to become a "1" after: 1. 80 step pulses in the RECALIBRATE COMMAND. 2. The RELATIVE SEEK command causes the 82078 to step outward beyond Track 0.
3	—	—	Unused. This bit is always "0".
2	H	Head Address	The current head address.
1,0	DS1,0	Drive Select	The current selected drive.

8.5.1.2 Status Register 1

Bit #	Symbol	Name	Description
7	EN	End of Cylinder	The 82078 tried to access a section beyond the final sector of the track (255D). Will be set if TC is not issued after Read or Write Data Command.
6	—	—	Unused. This bit is always "0".
5	DE	Data Error	The 82078 detected a CRC error in either the ID field or the data field of a sector.
4	OR	Overrun/Underrun	Becomes set if the 82078 does not receive CPU or DMA service within the required time interval, resulting in data overrun or underrun.
3	—	—	Unused. Ths bit is always "0".
2	ND	No Data	Any one of the following: 1. READ DATA, READ DELETED DATA command, the 82091AA did not find the specified sector. 2. READ ID command, the 82091AA cannot read the ID field without an error. 3. READ TRACK command, the 82091AA cannot find the proper sector sequence.
1	NW	Not Writable	WP pin became a "1" while the 82091AA is executing a WRITE DATA, WRITE DELETED DATA, or FORMAT TRACK command.
0	MA	Missing Address Mark	Any one of the the following: 1. The 82091AA did not detect an ID address mark at the specified track after encountering the index pulse from the INDX# pin twice. 2. The 82091AA cannot detect a data address mark or a deleted data address mark on the specified track.

8.5.1.3 Status Register 2

Bit #	Symbol	Name	Description
7	—	—	Unused. This bit is always "0".
6	CM	Control Mark	Any one of the following: 1. READ DATA command, the 82078 encounters a deleted data address mark. 2. READ DELETED DATA command, the 82078 encountered a data address mark.
5	DD	Data Error in Data Field	The 82091AA detected a CRC error in the data field.
4	WC	Wrong Cylinder	The track address from the sector ID field is different from the track address maintained inside the 82091AA.
3	—	—	Unused. This bit is always "0".
2	—	—	Unused. This bit is always "0".
1	BC	Bad Cylinder	The track address from the sector ID field is different from the track address maintained inside the 82091AA and is equal to FF hex which indicates a bad track with a hard error according to the IBM soft-sectored format.
0	MD	Missing Data Address Mark	The 82091AA cannot detect a data address mark or a deleted data address mark.

8.5.1.4 Status Register 3

Bit #	Symbol	Name	Description
7	—	—	Unused. This bit is always "0".
6	WP	Write Protected	Indicates the status of the WP pin.
5	—	—	Unused. This bit is always "0".
4	T0	Track 0	Indicates the status of the TRK0 pin.
3	—	—	Unused. This bit is always "0".
2	HD	Head Address	Indicates the status of the HDSEL pin.
1,0	DS1,0	Drive Select	Indicates the status of the DS1, DS0 pins.

8.5.2 DATA TRANSFER COMMANDS

All of the READ DATA, WRITE DATA and VERIFY type commands use the same parameter bytes and return the same results information. The only difference being the coding of bits[4:0] in the first byte.

An implied seek will be executed if the feature was enabled by the CONFIGURE Command. This seek is completely transparent to the user. The Drive Busy bit for the drive will go active in the Main Status Register during the seek portion of the command. A seek portion failure is reflected in the results status normally returned for a READ/WRITE DATA Command. Status Register 0 (ST0) contains the error code and C contains the cylinder that the seek failed.

8.5.2.1 Read Data

A set of nine bytes is required to place the FDC into the Read Data Mode. After the READ DATA Command has been issued, the FDC loads the head (if it is in the unloaded state), waits the specified head settling time (defined in the SPECIFY Command), and begins reading ID address marks and ID fields. When the sector address read from the diskette matches with the sector address specified in the command, the FDC reads the sector's data field and transfers the data to the FIFO.

After completion of the read operation from the current sector, the sector address is incremented by one, and the data from the next logical sector is read and output via the FIFO. This continuous read function is called "Multi-Sector Read Operation". Upon receipt of TC or an implied TC (FIFO overrun/underrun), the FDC stops sending data. However, the FDC will continue to read data from the current sector, check the CRC bytes, and, at the end of the sector, terminate the READ DATA Command.

N determines the number of bytes per sector (Table 25). If N is set to zero, the sector size is set to 128. The DTL value determines the number of bytes to be transferred. If DTL is less than 128, the FDC transfers the specified number of bytes to the host. For reads, it continues to read the entire 128 byte sector and checks for CRC errors. For writes it completes the 128 byte sector by filling in zeroes. If N is not set to 00h, DTL should be set to FFh, and has no impact on the number of bytes transferred.

Table 25. Sector Sizes

N	Sector Size
00	128 Bytes
01	256 Bytes
02	512 Bytes
03	1024 Bytes
...	...
07	16 KBytes

The amount of data that can be handled with a single command to the FDC depends on MT (multi-track) and N (Number of bytes/sector).

Table 26. Effects of MT and N Bits

MT	N	Max. Transfer Capacity	Final Sector Read from Disk
0	1	256 × 26 = 656	26 at side 0 or 1
1	1	256 × 52 = 13312	26 at side 1
0	2	512 × 15 = 7680	15 at side 0 or 1
1	2	512 × 30 = 15360	15 at side 1
0	3	1024 × 8 = 8192	8 at side 0 or 1
1	3	1024 × 16 = 16384	16 at side 1

The Multi-Track function (MT) allows the FDC to read data from both sides of the diskette. For a particular cylinder, data will be transferred starting at sector 1, side 0 and completing at the last sector of the same track at side 1.

If the host terminates a read or write operation in the FDC, the ID information in the result phase is dependent on the state of the MT bit and EOT byte. Refer to Table 29. The termination must be normal.

At the completion of the READ DATA Command, the head is not unloaded until after the Head Unload Time Interval (specified in the SPECIFY Command) has elapsed. If the host issues another command before the head unloads, the head settling time may be saved between subsequent reads.

If the FDC detects a pulse on the INDEX# pin twice without finding the specified sector (meaning that the diskette's index hole passes through index detect logic in the drive twice), the FDC sets the IC

5

code in Status Register 0 to 01 (Abnormal termination), sets the ND bit in Status Register 1 to 1 indicating a sector not found and terminates the READ DATA Command.

After reading the ID and data fields in each sector, the FDC checks the CRC bytes. If a CRC error occurs in the ID or data field, the FDC sets the IC code in Status Register 0 to 01 (Abnormal termination), sets the DE bit flag in Status Register 1 to 1, sets the DD bit in Status Register 2 to 1 if CRC is incorrect in the ID field, and terminates the READ DATA Command.

Table 27 describes the affect of the SK bit on the READ DATA command execution and results.

8.5.2.2 Read Deleted Data

This command is the same as the READ DATA Command, except that it operates on sectors that contain a deleted data address mark at the beginning of a data field. Table 28 describes the affect of the SK bit on the READ DELETED DATA Command execution and results.

Table 27. Skip Bit vs READ DATA Command

SK Bit Value	Data Address Mark Type Encountered	Sector Read	Results CM Bit of ST2 Set?	Description of Results
0	Normal Data	Yes	No	Normal Termination
0	Deleted Data	Yes	Yes	Address Not Incremented. Next Sector Not Searched For.
1	Normal Data	Yes	No	Normal Termination
1	Deleted Data	No	Yes	Normal Termination Sector Not Read ("Skipped")

Except where noted in Table 27, the C or R value of the sector address is automatically incremented (see Table 29).

Table 28. Skip Bit vs READ DELETED DATA Command

SK Bit Value	Data Address Mark Type Encountered	Sector Read	Results CM Bit of ST2 Set?	Description of Results
0	Normal Data	Yes	Yes	Normal Termination
0	Deleted Data	Yes	No	Address Not Incremented. Next Sector Not Searched For.
1	Normal Data	No	Yes	Normal Termination Sector Not Read ("Skipped")
1	Deleted Data	Yes	No	Normal Termination

Except where noted in Table 28, the C or R value of the sector address is automatically incremented (see Table 29).

Table 29. Result Phase

MT	Head	Final Sector Transferred to Host	ID Information at Result Phase			
			C	H	R	N
	0	Less than EOT	NC	NC	R+1	NC
0		Equal to EOT	C+1	NC	01	NC
	1	Less than EOT	NC	NC	R+1	NC
		Equal to EOT	C+1	NC	01	NC
	0	Less than EOT	NC	NC	R+1	NC
1		Equal to EOT	NC	LSB	01	NC
	1	Less than EOT	NC	NC	R+1	NC
		Equal to EOT	C+1	LSB	01	NC

NOTE:
1. NC=no change; the same value as the one at the beginning of command execution.
2. LSB=least significant bit; the LSB of H is complemented.

8.5.2.3 Read Track

This command is similar to the READ DATA Command except that the entire data field is read continuously from each of the sectors of a track. Immediately after encountering a pulse on the INDEX# pin, the FDC starts to read all data fields on the track as continuous blocks of data without regard to logical sector numbers. If the FDC finds an error in the ID or DATA CRC check bytes, it continues to read data from the track and sets the appropriate error bits at the end of the command. The FDC compares the ID information read from each sector with the specified value in the command and sets the ND flag to 1 in Status Register 1 if there is no comparison. Multi-track or skip operations are not allowed with this command. The MT and SK bits (Bits D7 and D5 of the first command byte respectively) should always be set to 0.

This command terminates when the EOT specified number of sectors have been read. If the FDC does not find an ID address mark on the diskette after the second occurrence of a pulse on the INDEX# pin, then it sets the IC code in Status Register 0 to 01 (Abnormal termination), sets the MA bit in Status Register 1 to 1, and terminates the command.

8.5.2.4 Write Data

After the WRITE DATA Command has been issued, the FDC loads the head (if it is in the unloaded state), waits the specified head load time if unloaded (defined in the SPECIFY Command), and begins reading ID fields. When the sector address read from the diskette matches the sector address specified in the command, the FDC reads the data from the host via the FIFO, and writes it to the sector's data field.

After writing data into the current sector, the FDC computes the CRC value and writes it into the CRC field at the end of the sector transfer. The sector number stored in R is incremented by one, and the FDC continues writing to the next data field. The FDC continues this multi-sector write operation. If a terminal count signal is received or a FIFO over/under run occurs while a data field is being written, the remainder of the data field is filled with zeros.

The FDC reads the ID field of each sector and checks the CRC bytes. If the FDC detects a CRC error in one of the ID fields, it sets the IC code in Status Register 0 to 01 (Abnormal termination), sets the DE bit of Status Register 1 to 1, and terminates the WRITE DATA Command.

The WRITE DATA Command operates in much the same manner as the READ DATA Command. The following items are the same. Please refer to the READ DATA Command for details:

- Transfer Capacity
- EN (End of Cylinder) bit
- ND (No Data) bit
- Head Load, Unload Time Interval
- ID information when the host terminates the command
- Definition of DTL when N=0 and when N does not=0

8.5.2.5 Verify

The VERIFY Command is used to verify the data stored on a disk. This command acts exactly like a READ DATA Command except that no data is transferred to the host. Data is read from the disk, and CRC is computed and checked against the previously stored value.

Because no data is transferred to the host, the TC signal cannot be used to terminate this command. By setting the EC bit to 1, an implicit TC will be issued to the FDC. This implicit TC occurs when the SC value has decrement to 0 (a SC value of 0 verifies 256 sectors). This command can also be terminated by setting the EC bit to 0 and the EOT value equal to the final sector to be checked. When EC=0, DTL/SC should be programmed to 0FFh. Refer to Table 29 and Table 30 for information concerning the values of MT and EC versus SC and EOT value.

Definitions:

Sectors Per Side = Number of formatted sectors per each side of the disk.

Sectors Remaining = Number of formatted sectors left that can be read, including side 1 of the disk when MT=1.

Table 30. Verify Command Result Phase

MT	EC	SC/EOT Value	Termination Result
0	0	SC=DTL EOT ≤ # Sectors Per Side	Successful Termination Result Phase Valid
0	0	SC=DTL EOT > # Sectors Per Side	Unsuccessful Termination Result Phase Invalid
0	1	SC ≤ # Sectors Remaining AND EOT ≤ # Sectors Per Side	Successful Termination Result Phase Valid
0	1	SC > # Sectors Remaining OR EOT > # Sectors Per Side	Unsuccessful Termination Result Phase Invalid
1	0	SC=DTL EOT ≤ # Sectors Per Side	Successful Termination Result Phase Valid
1	0	SC=DTL EOT > # Sectors Per Side	Unsuccessful Termination Result Phase Invalid
1	1	SC ≤ # Sectors Remaining AND EOT ≤ # Sectors Per Side	Successful Termination Result Phase Valid
1	1	SC > # Sectors Remaining OR EOT > # Sectors Per Side	Unsuccessful Termination Result Phase Invalid

NOTE:
When MT=1 and the SC value is greater than the number of remaining formatted sectors on Side 0, verification continues on Side 1 of the disk.

8.5.2.6 Format Track

The FORMAT TRACK Command allows an entire track to be formatted. After a pulse from the INDEX# pin is detected, the FDC starts writing data on the disk including gaps, address marks, ID fields and data fields, per the IBM* System 34 (MFM). The particular values written to the gap and data field are controlled by the values programmed into N, SC, GPL, and D which are specified by the host during the command phase. The data field of the sector is filled with the data byte specified by D. The ID field for each sector is supplied by the host. That is, four data bytes per sector are needed by the FDC for C, H, R, and N (cylinder, head, sector number, and sector size, respectively).

After formatting each sector, the host must send new values for C, H, R, and N to the FDC for the next sector on the track. The R value (sector number) is the only value that must be changed by the host after each sector is formatted. This allows the disk to be formatted with nonsequential sector addresses (inter-leaving). This incrementing and formatting continues for the whole track until the FDC encounters a pulse on the INDEX# pin again and it terminates the command.

Table 31 contains typical values for gap fields that are dependent on the size of the sector and the number of sectors on each track. Actual values can vary due to drive electronics.

Table 31. Typical PC/AT Values for Formatting

Drive Form	MEDIA	Sector Size	N	SC	GPL1	GPL2
5.25″	1.2 MB	512	02	0F	2A	50
	360 KB	512	02	09	2A	50
3.5″	2.88 MB	512	02	24	38	53
	1.44 MB	512	02	18	1B	54
	720 KB	512	02	09	1B	54

NOTES:
1. All values are in hex, except sector size.
2. Gap3 is programmable during reads, writes, and formats.
3. GPL1 = suggested Gap3 values in read and write commands to avoid splice point between data field and ID field of contiguous sections.
4. GPL2 = suggested Gap3 value in FORMAT TRACK Command.

5

8.5.2.7 Format Field

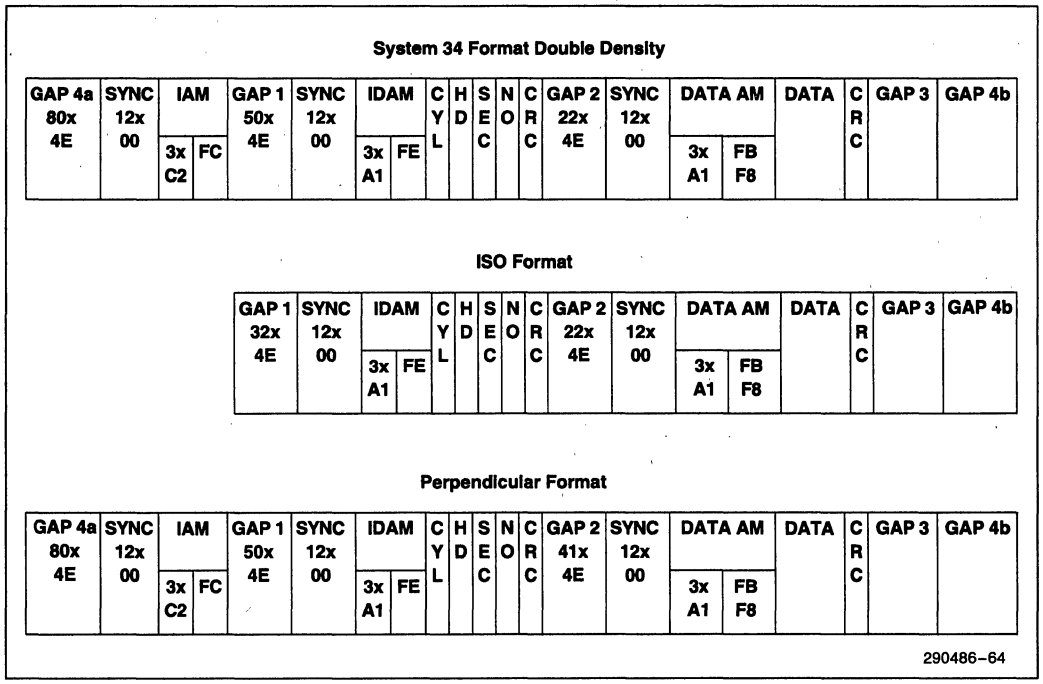


Figure 64. System 34, ISO and Perpendicular Formats

8.5.3 CONTROL COMMANDS

Control commands differ from the other commands in that no data transfer takes place. Three commands generate an interrupt when complete; READ ID, RECALIBRATE and SEEK. The other control commands do not generate an interrupt.

8.5.3.1 READ ID Command

The READ ID Command is used to find the present position of the recording heads. The FDC stores the values from the first ID field it is able to read into its registers. If the FDC does not find an ID address mark on the diskette after the second occurrence of a pulse on the INDEX# pin, it then sets the IC code in Status Register 0 to 01 (Abnormal termination), sets the MA bit in Status Register 1 to 1, and terminates the command.

The following commands will generate an interrupt upon completion. They do not return any result bytes. It is recommended that control commands be followed by the SENSE INTERRUPT STATUS Command. Otherwise, valuable interrupt status information will be lost.

8.5.3.2 RECALIBRATE Command

This command causes the read/write head within the FDC to retract to the track 0 position. The FDC clears the contents of the PCN counter, and checks the status of the TRK0 pin from the FDD. As long as the TRK0 pin is low, the DIR# pin remains 0 and step pulses are issued. When the TRK0 pin goes high, the SE bit in Status Register 0 is set to 1, and the command is terminated. If the TRK0 pin is still low after 79 step pulses have been issued, the FDC sets the SE and the EC bits of Status Register 0 to 1 and terminates the command. Disks capable of handling more than 80 tracks per side may require more than one RECALIBRATE Command to return the head back to physical Track 0.

The RECALIBRATE Command does not have a result phase. The SENSE INTERRUPT STATUS Command must be issued after the RECALIBRATE Command to effectively terminate it and to provide verification of the head position (PCN). During the command phase of the recalibrate operation, the FDC is in the busy state, but during the execution phase it is in a non-busy state. At this time another RECALIBRATE Command may be issued, and in this manner, parallel RECALIBRATE operations may be done on up to 2 drives simultaneously.

After powerup, software must issue a RECALIBRATE Command to properly initialize all drives and the controller.

8.5.3.3 DRIVE SPECIFICATION Command

The FDC uses two pins, DRVDEN0 and DRVDEN1 to select the density for modern drives. These signals inform the drive of the type of diskette in the drive. The DRIVE SPECIFICATION Command specifies the polarity of the DRVDEN0 and DRVDEN1 pins. It also enables/disables DSR programmed precompensation.

This command removes the need for a hardware work-around to accommodate differing specifications among drives. By programming this command during BIOS's POST routine, the floppy disk controller internally configures the correct values for DRVDEN0 and DRVDEN1 with corresponding precompensation value and data rate table enabled for the particular type of drive.

This command is protected from software resets. After executing the DRIVE SPECIFICATION Command, subsequent software resets will not clear the programmed parameters. Only another DRIVE SPECIFICATION Command or hard reset can reset it to default values. The 6 LSBs of the last byte of this command are reserved for future use.

The DRATE0 and DRATE1 are values as programmed in the DSR register. See Table 32 for pin decoding at different data rates.

Table 32 describes the drives that are supported with the DT0, DT1 bits of the DRIVE SPECIFICATION Command:

Table 32. DRVDENn Polarities

DT1	DT0	Data Rate	DRVDEN1	DRVDEN0
0*	0*	1 Mbps	1	1
		500 Kbps	0	1
		300 Kbps	1	0
		250 Kbps	0	0
0	1	1 Mbps	1	0
		500 Kbps	0	0
		300 Kbps	1	1
		250 Kbps	0	1
1	0	1 Mbps	1	1
		500 Kbps	0	0
		300 Kbps	1	0
		250 Kbps	0	1
1	1	1 Mbps	1	1
		500 Kbps	0	0
		300 Kbps	0	1
		250 Kbps	1	0

NOTE:
(*) Denotes the default setting

8.5.3.4 SEEK Command

The read/write head within the drive is moved from track to track under the control of the SEEK Command. The FDC compares the PCN which is the current head position with the NCN and performs the following operation if there is a difference:

PCN < NCN: Direction signal to drive set to 1 (step in), and issues step pulses.

PCN > NCN: Direction signal to drive set to 0 (step out), and issues step pulses.

The rate at which step pulses are issued is controlled by SRT (Stepping Rate Time) in the SPECIFY Command. After each step pulse is issued, NCN is compared against PCN, and when NCN = PCN, then the SE bit in Status Register 0 is set to 1, and the command is terminated.

During the command phase of the seek or recalibrate operation, the FDC is in the busy state, but during the execution phase it is in the non-busy state.

Note that if implied seek is not enabled, the read and write commands should be preceded by:

1. SEEK Command;
   Step to the proper track
2. SENSE INTERRUPT STATUS Command;
   Terminate the SEEK Command
3. READ ID.
   Verify head is on proper track
4. Issue READ/WRITE Command.

The SEEK Command does not have a result phase. Therefore, it is highly recommended that the SENSE INTERRUPT STATUS Command be issued after the SEEK Command to terminate it and to provide verification of the head position (PCN). The H bit (Head Address) in ST0 will always return a 0. When exiting DSR Powerdown mode, the FDC clears the PCN value and the status information to zero. Prior to issuing the DSR POWERDOWN Command, it is highly recommended that the user service all pending interrupts through the SENSE INTERRUPT STATUS Command.

8.5.3.5 SENSE INTERRUPT STATUS Command

An interrupt signal on the INT pin is generated by the FDC for one of the following reasons:

1. Upon entering the Result Phase of:
   a. READ DATA Command
   b. READ TRACK Command
   c. READ ID Command
   d. READ DELETED DATA Command
   e. WRITE DATA Command
   f. FORMAT TRACK Command
   g. WRITE DELETED DATA Command
   h. VERIFY Command
2. End of SEEK, RELATIVE SEEK or RECALIBRATE Command
3. FDC requires a data transfer during the execution phase in the non-DMA Mode

The SENSE INTERRUPT STATUS Command resets the interrupt signal and via the IC code and SE bit of Status Register 0, identifies the cause of the interrupt. If a SENSE INTERRUPT STATUS Command is issued when no active interrupt condition is present, the status register ST0 will return a value of 80h (invalid command).

The SEEK, RELATIVE SEEK and the RECALIBRATE Commands have no result phase. The SENSE INTERRUPT STATUS Command must be issued immediately after these commands to terminate them and to provide verification of the head position (PCN). The H (Head Address) bit in ST0 will always return a 0. If a SENSE INTERRUPT STATUS is not issued, the drive, will continue to be busy and may effect the operation of the next command.

8.5.3.6 SENSE DRIVE STATUS Command

The SENSE DRIVE STATUS Command obtains drive status information. It has no execution phase and goes directly to the result phase from the command phase. STATUS REGISTER 3 contains the drive status information.

8.5.3.7 SPECIFY Command

The SPECIFY Command sets the initial values for each of the three internal timers. The HUT (Head Unload Time) defines the time from the end of the execution phase of one of the read/write commands to the head unload state. The SRT (Step Rate Time) defines the time interval between adjacent step pulses. Note that the spacing between the first and second step pulses may be shorter than the remaining step pulses. The HLT (Head Load Time) defines the time between the command phase to the execution phase of a READ DATA or Write Data Command. The Head Unload Time (HUT) timer goes from the end of the execution phase to the begining of the result phase of a READ Data or Write Data Command. The values change with the data rate speed selection and are documented in Table 34.

Table 33. Interrupt Identification

SE	IC	Interrupt Due To
0	11	Polling
1	00	Normal Termination of SEEK or RECALIBRATE Command
1	01	Abnormal Termination of SEEK or RECALIBRATE Command

Table 34. Drive Control Delays (ms)

	HUT				SRT			
	1 M	500K	300K	250K	1 M	500K	300K	250K
0	128	256	426	512	8.0	16	26.7	32
1	8	16	26.7	32	7.5	15	25	30
..	..	..	..	..	..	..	..	..
A	80	160	267	320	3.0	6.0	10.2	12
B	88	176	294	352	2.5	5.0	8.3	10
C	96	192	320	384	2.0	4.0	6.68	8
D	104	208	346	416	1.5	3.0	5.01	6
E	112	224	373	448	1.0	2.0	3.33	4
F	120	240	400	480	0.5	1.0	1.67	2

Table 35. Head Load Time (ms)

	HLT			
	1M	500K	300K	250K
00	128	256	426	512
01	1	2	3.3	4
02	2	4	6.7	8
..	..	..	..	..
7E	126	252	420	504
7F	127	254	423	508

The choice of DMA or non-DMA operations is made by the ND bit. When ND = 1, the non-DMA mode is selected, and when ND = 0, the DMA mode is selected. In DMA mode, data transfers are signalled by the DRQ pin. Non-DMA mode uses the RQM bit and the IRQ6 pin to signal data transfers.

8.5.3.8 CONFIGURE Command

Issue the configure command to enable features like the programmable FIFO and set the begining track for precompensation. A CONFIGURE Command need not be issued if the default values of the FDC meets the system requirements.

CONFIGURE DEFAULT VALUES:

EIS	**No Implied Seeks**
EFIFO	FIFO Disabled
POLL	Polling Enabled
FIFOTHR	FIFO Threshold Set to 1 Byte
PRETRK	Pre-Compensation Set to Track 0

EIS—Enable Implied Seek. When EIS = 1, the FDC will perform a SEEK operation before executing a read/write command. The default value is 0 (no implied seek).

EFIFO—Enable FIFO. When EFIFO = 1, the FIFO is disabled (8272A compatible mode). This means data transfers are asked for on a byte by byte basis. The default value is 1 (FIFO disabled). The threshold defaults to one.

POLL—Disable Polling. When POLL = 1, polling of the drives is disabled. POLL Defaults to 0 (polling enabled). When enabled, a single interrupt is generated after a reset. No polling is performed while the drive head is loaded and the head unload delay has not expired.

FIFOTHR—The FIFO threshold in the execution phase of a read/write command. This is programmable from 1 to 16 bytes. FIFOTHR defaults to one byte. A 00 selects one byte and a 0F selects 16 bytes.

PRETRK—Precompensation start track number. Programmable from track 0 to 255. PRETRK defaults to track 0. A 00h selects track 0 and a FFh selects 255.

8.5.3.9 VERSION Command

The VERSION Command checks to see if the controller is an enhanced type (82077, 82077AA, 82077SL) or the older type (8272A/765A). A value of 90h is returned as the result byte, defining an enhanced FDD controller is in use. No interrupts are generated.

8.5.3.10 RELATIVE SEEK Command

The RELATIVE SEEK Command is coded the same as for the SEEK Command, except for the MSB of the first byte and the DIR# bit.

DIR# Head Step Direction Control

DIR#	ACTION
0	Step Head Out
1	Step Head In

RCN Relative Cylinder Number that determines how many tracks to step the head in or out from the current track number.

The RELATIVE SEEK Command differs from the SEEK Command in that it steps the head the absolute number of tracks specified in the command instead of making a comparison against an internal register. The SEEK Command is good for drives that support a maximum of 256 tracks. RELATIVE SEEKs cannot be overlapped with other RELATIVE SEEKs. Only one RELATIVE SEEK can be active at a time. Bit 4 of Status Register 0 (EC) will be set to 1 if RELATIVE SEEK attempts to step outward beyond Track 0.

As an example, assume that a floppy drive has 300 useable tracks and that the host needs to read track 300 and the head is on any track (0–255). If a SEEK Command is issued, the head stops at track 255. If a RELATIVE SEEK Command is issued, the FDC moves the head the specified number of tracks, regardless of the internal cylinder position register (but increments the register). If the head had been on track 40 (D), the maximum track that the FDC could position the head on using RELATIVE SEEK, is 296 (D), the initial track, +256 (D). The maximum count that the head can be moved with a single RELATIVE SEEK Command is 256 (D).

The internal register, PCN, would overflow as the cylinder number crossed track 255 and would contain 40 (D). The resulting PCN value is thus (NCN + PCN) mod 256. Functionally, the FDC starts counting from 0 again as the track number goes above 255(D). It is the users responsibility to compensate FDC functions (precompensation track number) when accessing tracks greater than 255. The FDC does not keep track that it is working in an "extended track area" (greater than 255). Any command issued uses the current PCN value, except for the RECALIBRATE Command that only looks for the TRACK0 signal. RECALIBRATE returns an error if the head is farther than 79 due to its limitation of issuing a maximum 80 step pulses. The user simply needs to issue a second RECALIBRATE Command. The SEEK Command and implied seeks function correctly within the 44 (D) track (299–255) area of the extended track area. It is the users responsibility not to issue a new track position that exceeds the maximum track that is present in the extended area.

To return to the standard floppy range (0–255) of tracks, a RELATIVE SEEK is issued to cross the track 255 boundary.

A RELATIVE SEEK Command can be used instead of the normal SEEK Command but the host is required to calculate the difference between the current head location and the new (target) head location. This may require the host to issue a READ ID Command to ensure that the head is physically on the track that software assumes it to be. Different FDC commands return different cylinder results which may be difficult to keep track of with software without the READ ID Command.

8.5.3.11 DUMPREG Command

The DUMPREG Command is designed to support system run-time diagnostics and application software development and debug. The command returns pertinent information regarding the status of many of the programmed fields in the FDC. This can be used to verify the values initialized in the FDC.

8.5.3.12 PERPENDICULAR MODE Command

An added capability of the FDC is the ability to interface directly to perpendicular recording floppy drives. Perpendicular recording differs from the traditional longitudinal method by orienting the magnetic bits vertically. This scheme packs in more data bits for the same area.

The PERPENDICULAR MODE Command allows the system designers to designate specific drives as Perpendicular recording drives. Data transfers be-

tween Conventional and Perpendicular drives are allowed without having to issue PERPENDICULAR MODE Commands between the accesses of the two different drives, nor having to change write precompensation values.

With this command, the length of the Gap2 field and VCO enable timing can be altered to accommodate the unique requirements of these drives. Table 36 describes the effects of the WGATE and GAP bits for the PERPENDICULAR MODE Command.

When both GAP and WGATE equal 0 the PERPENDICULAR MODE Command will have the following effect on the FDC:

1. If any of the new bits D0 and D1 are programmed to 1, the corresponding drive is automatically programmed for Perpendicular mode (ie: GAP2 being written during a write operation, the programmed Data Rate will determine the length of GAP2), and data will be written with 0 ns write precompensation.
2. Any of the new bits (DO/D1) are programmed for 0, the designated drive is programmed for Conventional Mode and data will be written with the currently programmed write precompensation value.
3. Bits D0 and D1 can only be over-written when the OW bit is 1. The status of these bits can be determined by interpreting the eighth result byte of the DUMPREG Command. (Note: if either the GAP or WGATE bit is 1, bits D0 and D1 are ignored.)

Software and Hardware reset have the following effects on the enhanced PERPENDICULAR MODE Command:

1. A software reset (Reset via DOR or DSR registers) only sets GAP and WGATE bits to 0; D0 and D1 retain their previously programmed values.
2. A hardware reset (Reset via pin 32) sets all bits (GAP, Wgate, D0, and D1) to 0 (All Drives Conventional Mode).

8.5.3.13 POWERDOWN MODE Command

The POWERDOWN MODE Command allows the automatic power management and enables the enhanced registers (EREG EN) of the FDC. The use of the command can extend the battery life in portable PC applications. To enable auto powerdown the command may be issued during the BIOS power on self test (POST).

This command includes the ability to configure the FDC into the enhanced mode extending the SRB and TDR registers. These extended registers accommodate bits that give more information about floppy drive interface, allow for boot drive selection, and identify the values of the PD and IDLE status.

As soon as the command is enabled, a 10 ms or a 0.5 sec minimum powerup timer is initiated, depending on whether the MIN DLY bit is set to 0 or 1. This timer is one of the required conditions that has to be satisfied before the FDC will enter auto powerdown.

Table 36. Effects of WGATE and GAP Bits

GAP	WGATE	MODE	VCO Low Time after Index Pulse	Length of Gap2 Format Field	Portion of Gap2 Written by Write Data Operation	Gap2 VCO Low Time for Read Operations
0	0	Conventional Mode	33 Bytes	22 Bytes	0 Bytes	24 Bytes
0	1	Perpendicular Mode (500 Kbps and Lower Data Rates)	33 Bytes	22 Bytes	19 Bytes	24 Bytes
1	0	Reserved (Conventional)	33 Bytes	22 Bytes	0 Bytes	24 Bytes
1	1	Perpendicular Mode (1 Mbps Data Rate)	18 Bytes	41 Bytes	38 Bytes	43 Bytes

NOTE:
When either GAP or WGATE bit is set, the current value of precompensation in the DSR is used.

Any software reset will re-initialize the timer. The timer countdown is also extended by up to 10 ms if the data rate is changed during the timer's countdown. Without this timer, the FDC would have been put to sleep immediately after FDC is idle. The minimum delay gives software a chance to interact with the FDC without incurring an additional overhead due to recovery time.

The command also allows the output pins of the floppy disk drive interface to be tri-stated or left unaltered during auto powerdown. This is done by the FDI TRI bit. In the default condition (FDI TRI = 0) the output pins of the floppy disk drive are tri-stated. Setting this bit leaves the interface unchanged from the normal state.

The results phase returns the values programmed for MIN DLY, FDI TRI and AUTO PD. The auto powerdown mode is disabled by a hardware reset. Software results have no effect on the POWERDOWN MODE Command parameters.

8.5.3.14 PART ID Command

This command can be used to identify the floppy disk controller as an enhanced controller. The first stepping of the FDC (all versions) will yield 0x02 in the result phase of this command. Any future enhancements on these parts will be denoted by the 5 LSBs (0x01 to 0x1F).

8.5.3.15 OPTION Command

The standard IBM format includes an index address field consisting of 80 bytes of GAP 4a, 12 bytes of the sync field, four bytes identifying the IAM and 50 bytes of GAP 1. Under the ISO format most of this preamble is not used. The ISO format allows only 32 bytes of GAP 1 after the index mark. The ISO bit in this command allows the FDC to configure the data transfer commands to recognize this format. The MSBs in this command are reserved for any other enhancements made available to the user in the future.

8.5.3.16 SAVE Command

The first byte corresponds to the values programmed in the DSR with the exception of CLKSEL. The DRATE1, DRATE0 used here are unmapped. The second byte is used for configuring the bits from the OPTION Command. All future enhancements to the OPTION Command will be reflected in this byte as well. The next nine result bytes are explained in the Parameter Abbreviations section after the command summary. The 13th byte is the value associated with the POWERDOWN MODE Command. The disk status is used internally by the FDC. There are two reserved bytes at the end of this command for future use.

This command is similar to the DUMPREG Command but it additionally allows the user to read back the precompensation values as well as the programmed data rate. It also allows the user to read the values programmed in the POWERDOWN MODE Command. The precompensation values will be returned as programmed in the DSR register. This command, used in conjunction with the RESTORE Command, should prove very useful for SMM power management. This command reserves the last two bytes for future enhancements.

8.5.3.17 RESTORE Command

Using the RESTORE Command with the SAVE Command, allows the SMM power management to restore the FDC to its original state after a system powerdown. It also serves as a succinct way to provide most of the initialization requirements normally handled by the system. The sequence of initializing the FDC after a reset occurred and assuming a SAVE Command was issued follows:

- Issue the DRIVE SPECIFICATION Command (if the design utilizes this command)
- Issue the RESTORE Command (pass the 16 bytes retrieved previously during SAVE)

The RESTORE Command programs the data rate and precompensation value via the DSR. It then restores the values normally programmed through the CONFIGURE, SPECIFY, and PERPENDICULAR Commands. It also enables the previously selected values for the POWERDOWN Mode Command. The PCN values are set restored to their previous values and the user is responsible for issuing the SEEK and RECALIBRATE Commands to restore the head to the proper location. There are some drives that do not recalibrate in which case the RESTORE Command restores the previous state completely. The PDOSC bit is retrievable using the SAVE Command, however, the system designer must set it correctly. The software must allow at least 20 μs to execute the RESTORE Command. When using the BOOTSEL bits in the TDR, the user must restore or reinitialize these bits to their proper values.

8.5.3.18 FORMAT AND WRITE Command

The FORMAT AND WRITE Command is capable of simultaneously formatting and writing data to the diskette. It is essentially the same as the normal FORMAT Command. With the exception that included in the execution for each sector is not only the C, H, R, and N but also the data transfer of N bytes. The D value is ignored. This command formats the entire track. High speed floppy diskette duplication can be done fast and efficiently with this command. The user can format the diskette and put data on it in a single pass. This is very useful for software duplication applications by reducing the time required to format and copy diskettes.

9.0 IDE INTERFACE

The 82091AA supports the IDE (Integrated Drive Electronics) interface by providing two chip selects, and lower and upper data byte controls. DMA and 16-bit data transfers are supported. Minimal external logic is required to complete the optional 16-bit IDE I/O and DMA interfaces. With external logic, a fully buffered interface is also supported.

9.1 IDE Registers

The 82091AA does not contain IDE registers. All of the IDE device registers are located in the IDE device, except bit 7 of the Drive Address Register which is the Floppy Controller Disk Change status bit and is driven by the 82091AA.

The IDE interface contains two chip selects (IDECS0# and IDECS1#). These signals are asserted for accesses to the Command and Control Block registers located at 01Fxh and 03Fxh, respectively (Table 37).

Table 37. IDE Register Set (Located in IDE Device)

Primary Address	Secondary Address	Chip Select	Registers	Access
1F0h	170h	IDECS0#	Data Register	R/W
1F1h	171h	IDECS0#	Error Register	RO
1F1h	171h	IDECS0#	Write Precomp/Features Register	WO
1F2h	172h	IDECS0#	Sector Count Register	R/W
1F3h	173h	IDECS0#	Sector Number Register	R/W
1F4h	174h	IDECS0#	Cylinder Low Register	R/W
1F5h	175h	IDECS0#	Cylinder High Register	R/W
1F6h	176h	IDECS0#	Drive/Head Register	R/W
1F7h	177h	IDECS0#	Status Register	RO
1F7h	177h	IDECS0#	Command Register	WO
3F6h	376h	IDECS1#	Alternate Status Register	RO
3F6h	376h	IDECS1#	Digital Output Register	WO
3F7h	377h	IDECS1#	Drive Address Register	RO
3F7h	377h	IDECS1#	Not Used	

9.2 IDE Interface Operation

The 82091AA implements the chip select signals for the IDE interface and decodes the standard PC/AT primary and secondary I/O locations.

The 82091AA provides a data buffer enable signal (DEN#) to control the lower data byte path for buffered designs. Buffering the lower data byte path is an application option that requires an external transceiver/buffer. For buffered applications, DEN# controls an external transceiver and enables data bits IDED[7:0] onto the system data bus SD[7:0]. For non-buffered applications (typically the X-Bus configuration), IDED[7:0] are connected directly to the bus and DEN# is not used and becomes a no-connect. For 16-bit applications the upper data byte path (IDED[15:8]) is controlled by the HEN# signal.

Figure 65 shows an example IDE interface without DMA capability. In this case all IDE accesses for setting up the IDE registers and transferring data is programmed via I/O. The 82091AA generates the chip selects (IDECS0# and IDECS1#). The 82091AA also generates the DEN# and HEN# signals to enable the data buffers.

Figure 66 shows an example DMA IDE interface for type "F" DMA cycles. To set up the IDE interface, the host accesses the IDE registers on the IDE device. For programmed I/O accesses, the 82091AA generates the chip selects (IDECS0# and IDECS1#) to access the IDE registers and the DEN# and HEN# signals to control the data buffers. During DMA transfers the DMA handshake is between the DMA controller and IDE device via the DREQ and DACK# signals. The DACK# signal is ORed with the DEN# and HEN# signals to control the upper and lower byte buffers during DMA transfers.

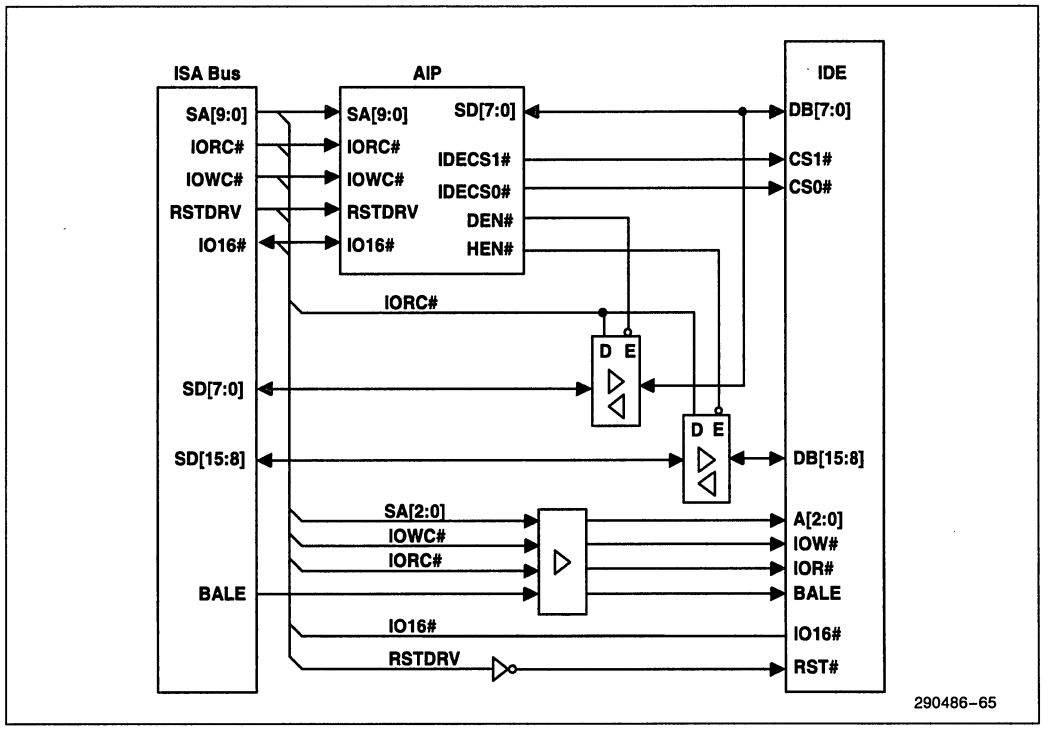


Figure 65. IDE Interface Example (without DMA)

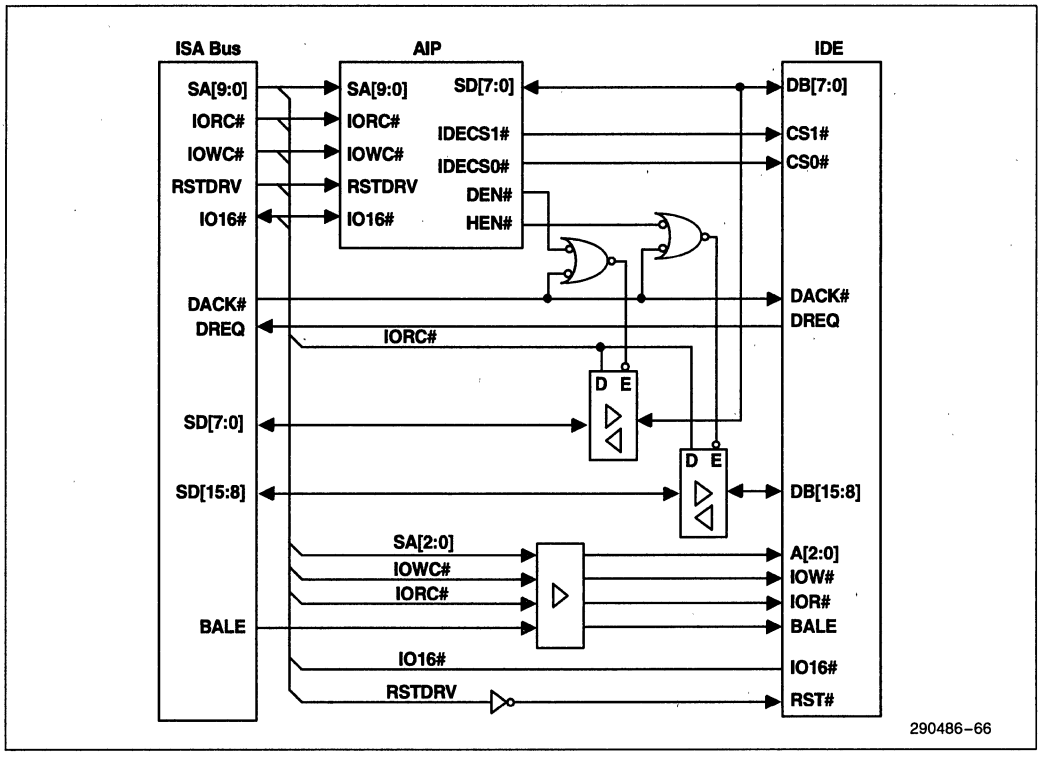


Figure 66. IDE Interface Example (with DMA)

10.0 POWER MANAGEMENT

The 82091AA provides power management capabilities for its primary functional modules (parallel port, floppy disk controller, serial port A, and serial port B). For each module, the 82091AA implements two types of power management—direct powerdown and auto powerdown. Direct powerdown, enabled via control bits in the 82091AA configuration registers, immediately places the module in a powerdown mode by turning off the clock to the associated module. Direct powerdown removes the clock regardless of the activity or status of the module. By contrast, when auto powerdown is enabled (via control bits in the 82091AA configuration registers), the associated module only enters a powerdown mode if it is in an idle state.

NOTE:
The entire 82091AA can be placed in direct powerdown by writing to the CLKOFF bit in the AIPCFG1 Register.

10.1 Power Management Registers

The floppy disk controller, parallel port, serial port A, and serial port B each have two 82091AA configuration registers. For each module, three configuration register bits control power management—xDPDN, xIDLE, and xAPDN.

- xAPDN: auto-powerdown, shuts off the oscillator to the module when the module is idle.
- xIDLE: idle status, a read only pin that indicates idle status.
- xDPDN: direct powerdown, shuts off module oscillator when active regardless of module status.

The 82091AA exits any powerdown mode after a hardware reset (RSTDRV asserted) or reset via the xRESET bit in the 82091AA configuration registers. Direct powerdown can also be exited by writing the corresponding xPDN bit in the configuration register to 0. Auto powerdown is exited by events at the module (e.g., CPU read/write or module interface activity).

NOTE:
The configuration registers also contain the xEN bit. This bit is used to completely disable an unused module. Enabling a disabled module takes much longer than restoring a module from powerdown. Therefore, this bit is not recommend for temporarily disabling a module as a powerdown scheme.

10.2 Clock Power Management

The internal clock circuitry of the 82091AA can be turned on or off as part of a power management scheme. The clock circuitry is controlled via the CLKOFF bit in the AIPCFG1 Register. If an external clock source exists, the user may want to turn off the internal oscillator to save power and provide minimum recovery time.

Auto powerdown and direct powerdown (in each module) have no effect on the state of internal oscillator.

10.3 FDC Power Management

This section describes the FDC direct and auto powerdown modes and recovery from the powerdown modes.

Auto Powerdown

Automatic powerdown (APDN) has an advantage over direct powerdown (PDN) since the register contents are not lost under APDN. Automatic powerdown is invoked by either the Auto Powerdown command, or by enabling the FAPDN bit in the FDC configuration register. There are four conditions required before the FDC will enter powerdown:

1. The motor enable pins ME[3:0] must be inactive.
2. The FDC must be in an idle state. FDC idle is indicated by MSR = 80h and the IRQ6 signal is negated (IRQ6 may be asserted even if MSR = 80h due to polling interrupt).
3. The head unload timer (HUT, explained in the SPECIFY Command) must have expired.
4. The auto powerdown timer must have timed out.

An internal timer is initiated when the POWERDOWN MODE Command is executed. The amount of time can be set by the user via the MIN DLY bits in the POWERDOWN MODE Command. The module is then powered down, provided all the remaining conditions are met. A software reset reinitializes the timer. When using the FDC FAPDN bit to enable the automatic powerdown feature, the MIN DLY bit is set to the default condition.

Recovery from Auto Powerdown

When the FDC is in auto powerdown, the module is awakened by a reset or access to the DOR, MSR or FIFO registers. The module remains in auto powerdown mode after a software reset (i.e., it will power-

down again after being idle for the time specified by MIN DLY). However, the FDC does not remain in auto powerdown mode after a hardware reset or DSR reset.

Direct Powerdown

Direct powerdown is invoked via the Powerdown bit in the Data Rate Select Register (bit 6), or the FDPDN bit in the FCFG2 Register. Setting FDPDN to 1 will powerdown the FDC. All status is lost when this type of powerdown mode is used. The FDC exits powerdown mode after any hardware or software reset. Direct powerdown overrides automatic powerdown.

Recovery from Direct Powerdown

The FDC exits the direct powerdown state by setting the FDPDN bit to 0 followed by a software or hardware reset.

After reset, the FDC goes through a normal sequence. The drive status is initialized. The FIFO mode is set to default mode on a hardware or software reset if the LOCK Command has not blocked it. Finally, after a delay, the polling interrupt is issued.

10.4 Serial Port Power Management

This section describes the serial port direct and auto powerdown modes and recovery from the powerdown modes.

Auto Powerdown

When auto powerdown is enabled in the SxCFG2 Register (SxAPDN bit is 1), the serial port enters auto powerdown based on monitoring line interface activity. During auto powerdown, the status of the serial port is maintained (the FIFO and registers are not reset). Access to any serial port register is allowed during auto powerdown. The transmitter and the receiver enter powerdown individually, depending on certain conditions. When there are no characters to transmit (TEMPTY = 1 in the LSR), the transmitter clock is shut off placing the transmitter in auto powerdown. In the case of the receiver, when serial input signal is inactive for approximately 5 character times, indicating that no character is being received, the receiver goes into auto powerdown.

Recovery from Auto Powerdown

The serial port recovers from auto powerdown when either the transmitter or receiver are active. If data is written to the transmitter or data is present at the receiver, the serial port exits from auto powerdown.

Direct Powerdown

Direct Powerdown is invoked via the SxCFG2 Register (setting the SxDPDN bit to 1). When in direct powerdown, the clock to the module is shut off. All registers are accessible while in direct powerdown. A host read of the Receiver Buffer Register or a write to the Transmitter Holding Register should not be performed during powerdown. The SINx input should remain static.

When direct powerdown is invoked, the transmit and receive sections of the serial port are reset, including the transmit and receive FIFOs. Thus, to prevent possible data loss when the FIFOs are reset, software should not invoke direct powerdown until the serial port is in the idle state as indicated by the SxIDLE bit in the SxCFG2 Register.

Recovery from Direct Powerdown

Recovery from direct powerdown is accomplished by writing the SxDPDN bit in the configuration register to 0 or by a module reset.

10.5 Parallel Port Power Management

Auto Powerdown

Auto powerdown is enabled via the PAPDN bit in the PCFG2 Register. When enabled, the parallel port enters auto powerdown when the module is in an idle state. If the parallel port FIFO is being used to transfer data, the parallel port is in an idle state when the FIFO is empty.

Recovery from Auto Powerdown

Recovery from auto powerdown occurs when the FIFO is written or as a result of parallel port interface activity.

Direct Powerdown

Direct powerdown is invoked via the PCFG2 Register (setting the PDPDN bit to 1). When PDPDN = 1, the clock to the printer state machine is disabled and the state machine goes into an idle state.

Recovery from Direct Powerdown

Recovery from direct powerdown is accomplished by setting the PDPDN bit to 0 or the PRESET bit to a 1 in the PCFG2 Register. An 82091AA hard reset (RSTDRV asserted) also brings the part out of direct powerdown.

11.0 ELECTRICAL CHARACTERISTICS

NOTICE: This data sheet contains information on products in the sampling and initial production phases of development. 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.

11.1 Absolute Maximum Ratings

Storage Temperature ..... −65°C to +150°C
Supply Voltage ..... −0.5V to +8.0V
Voltage on Any Input ..... GND−2V to 6.5V
Voltage on Any Output ... GND−0.5V to V_{CC} +0.5V
Power Dissipation 1W

**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.*

11.2 DC Characteristics

Table 38. DC Specifications (V_{CC} = 5V ±10%, T_{amb} = 0°C to 70°C)

Symbol	Parameter	V_{CC} = +5V ±10			V_{CC} = 3.3V ±0.3V		
		Min(V)	Max(V)	Notes	Min(V)	Max(V)	Notes
V_{ILC}	Input Low Voltage, X1	−0.5	0.8		−0.3	0.8	
V_{IHC}	Input High Voltage, X1	3.9	V_{CC} + 0.5		2.4	V_{CC} + 0.3	
V_{IL}	Input Low Voltage (all pins except X1)	−0.5	0.8		−0.3	0.8	
V_{IH}	Input High Voltage (all pins except X1)	2.0	V_{CC} + 0.5		2.0	V_{CC} + 0.3	
I_{CC}	V_{CC} Supply Current −1 Mbps FDC Data Rate V_{IL} = 0.45V, V_{IH} = 2.4V		50 mA	1, 2		40 mA	1, 2
I_{CCSB}	I_{CC} in Powerdown		100 µA	3, 4, 5		100 µA	3, 4, 5
I_{IL}	Input Load Current (all input pins)		+10µA −10 µA	6		+10 µA −10 µA	6
I_{OFL}	Data Bus Output Float Leakage		+10 µA −10 µA	7		+10 µA −10 µA	8
I_{BPL}	Parallel Port Back-Power Leakage (All Parallel Port Signals)		+10 µA	9		+10 µA	9

NOTES:
1. Test Conditions: Only the data bus inputs may float. All outputs are open.
2. Test Conditions: Tested while reading a sync field of "00". Outputs not connected to DC loads. This specification reflects the supply current when all modules within the 82091AA are active.
3. Test Conditions: $V_{IL} = V_{SS}$, $V_{IH} = V_{CC}$; Outputs not connected to DC loads.
4. Test Conditions: Typical value with the oscillator off.
5. Test Conditions: All 82091AA modules are in their powerdown state.
6. Test Conditions: 10 µA ($V_{IN} = V_{CC}$), −10µA (V_{IN} = 0V)
7. Test Conditions: $0V < V_{OH} < V_{CC}$
8. Test Conditions: $0.45V < V_{OH} < V_{CC}$
9. Test Conditions: Device in Circuit V_{CC} = 0V, V_{IN} = 5.5V max.

5

Table 39. Capacitance Specifications (V_{CC} = 5V ±10%, T_{amb} = 0°C to 70°C)

C_{IN}	Input Capacitance	10	pF	F = 1 MHz, T_A = 25°C
C_{IN1}	Clock Input Capacitance	20	pF	Sampled, not 100% Tested
$C_{I/O}$	Input/Output Capacitance	20	pF	

NOTE:
All pins except pins under test are tied to AC ground.

The following pin groupings are used in Table 40 and Table 41.

DMA FDDREQ, PPDREQ

IRQx IRQ3, IRQ4, IRQ5, IRQ6, IRQ7

Serial Port SOUTA, SOUTB, DTRA#, DTRB#, RTSA#, RTSB#

Parallel Port PD[7:0], STROBE#, AUTOFD#, INIT#, SELECTIN#

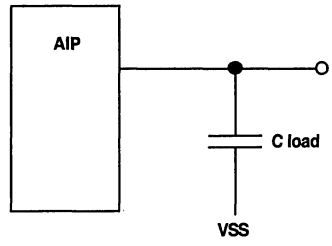
FDC Interface WRDATA, HDSEL#, STEP#, DIR#, WE#, FDME0#, FDME1#, FDS0#, FDS1#, DRVDEN[1:0]

Table 40. V_{OL} Specifications (V_{CC} = 5V ±10%, T_{amb} = 0°C to 70°C)

Symbol	Signal	V_{CC} = 5V ± 10%			V_{CC} 3.3V ± 0.3V		
		Min	Max	I_{OL}	Min	Max	I_{OL}
V_{OL}	SD[7:0]		0.45V	24 mA		0.45V	12 mA
V_{OL}	NOWS#, IOCHRDY		0.45V	24 mA		0.45V	12 mA
V_{OL}	DMA,IRQx		0.45V	12 mA		0.45V	6 mA
V_{OL}	Serial Port		0.45V	4 mA		0.45V	2 mA
V_{OL}	Parallel Port		0.45V	16 mA		0.45V	8 mA
V_{OL}	PPDIR,GCS#		0.45V	4 mA		0.45V	2 mA
V_{OL}	FDC Interface		0.45V	12 mA		0.45V	6 mA
V_{OL}	DEN#,HEN#		0.45V	4 mA		0.45V	2 mA
V_{OL}	IDECS[1:0]#		0.45V	12 mA		0.45V	6 mA

Table 41. V_{OH} Specifications (V_{CC} = 5V ±10%, T_{amb} = 0°C to 70°C)

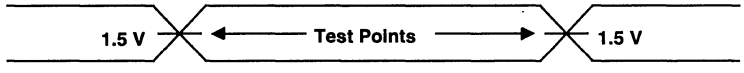
Symbol	Signal	V_{CC} = 5V ± 10%			V_{CC} 3.3V ± 0.3V		
		Min	Max	I_{OH}	Min	Max	I_{OH}
V_{OH}	SD[7:0]	2.4V		4 mA	2.4V		2 mA
V_{OH}	DMA,IRQx	2.4V		4 mA	2.4V		2 mA
V_{OH}	Serial Port	2.4V		1 mA	2.4V		1 mA
V_{OH}	Parallel Port	2.4V		4 mA	2.4V		50 μA
V_{OH}	PPDIR,GCS#	2.4V		1 mA	2.4V		1 mA
V_{OH}	FDC Interface	2.4V		4 mA	2.4V		2 mA
V_{OH}	DEN#, HEN#	2.4V		1 mA	2.4V		1 mA
V_{OH}	IDECS[1:0]#	2.4V		4 mA	2.4V		2 mA



C load = 50 pF for all logic outputs and
100 pF for the data bus.

290486-67

Figure 67. Load Circuit



290486-68

Figure 68. AC Testing Input, Output

5

11.3 Oscillator

The 24 MHz clock can be supplied either by a crystal (Figure 69) or a MOS level square wave. All internal timings are referenced to this clock or a scaled count that is data rate dependent. The crystal oscillator must be allowed to run for 10 ms after V_{CC} has reached 4.5V or exiting the POWERDOWN mode to guarantee that it is stable.

Crystal Specifications:

Freq:	24 MHz ± 0.1%
Mode:	Parallel Resonant
	Fundamental Mode
Series Resistance:	< 40Ω
Shunt Capacitance:	< 5 pF
C1, C2:	20 pF–25 pF

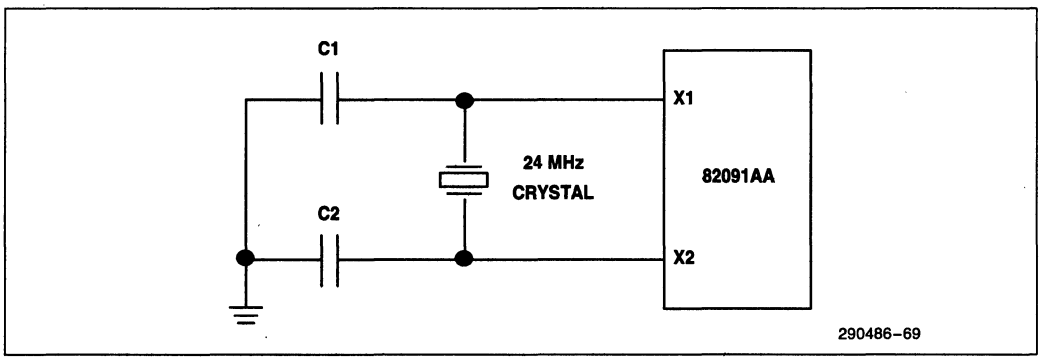


Figure 69. Crystal Connections

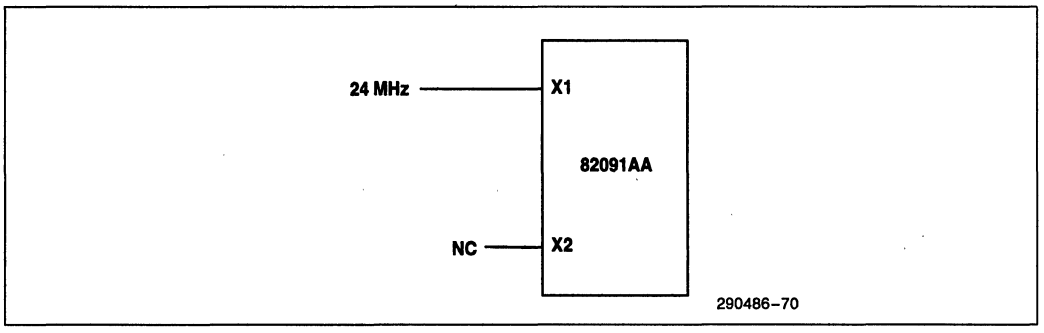


Figure 70. Oscillator Connections

11.4 AC Characteristics

Table 42. AC Specifications (V_{CC} = 5V ± 10%, T_{amb} = 0°C to 70°C)

Symbol	Parameter	24 MHz		Units	Notes	Figure
		Min	Max			
t1a	Clock Rise and Fall Time		10	ns	1	71
t1b	Clock High Time	16		ns	1	71
t1c	Clock Low Time	16		ns	1	71
t1d	Clock Period	41.66	41.66	ns	2	71
t1e	Internal Clock Period				3	

NOTES:
1. Clock input high level test points for clock high time and clock rise/fall times are 3.5V with V_{CC} at 5V ±10% and 2.0V with V_{CC} at 3.3V V_{CC} ±10%. Clock input low level test point for clock low time and clock rise/fall time is 0.8V.
2. Clock input test point for clock period is 0.8V.
3. Certain Floppy Disk Controller module timings are a function of the selected data rate. The nominal values for the internal clock period (t1e) for the various data rates are:

Disk Drive Disk Rate	Internal Clock Period (*nominal values)
	24 MHz
1 Mbps	125 ns
500 Kbps	250 ns
300 Kbps	420 ns
250 Kbps	500 ns

All information contained in ( ) in the following tables represents 3.3V specifications.

Table 43. AC Specifications (V_{CC} = 5V ± 10%, or [3.3V ± 0.3V] T_{amb} = 0°C to 70°C)

Symbol	Parameter	Min	Max	Units	Notes	Figure
Host						
SA[10:0]						
t2a	SA[10:0] Setup to IORC#/IOWC# Active	18 (25)		ns		72, 73
t2b	SA[10:0] Hold from IORC#/IOWC# Inactive	0		ns		72, 73
SD[7:0]						
t3a	SD[7:0] Valid Delay from IORC# Active		70 (100)	ns	1	72
t3b	SD[7:0] Float Delay from IORC# Inactive	5	35 (40)	ns		72
t3c	SD[7:0] Setup to IOWC# Inactive	35		ns		73
t3d	SD[7:0] Hold from IOWC# Inactive	0		ns		73

5

Table 43. AC Specifications (V_{CC} = 5V ± 10%, or (3.3V ± 0.3V) T_{amb} = 0°C to 70°C) (Continued)

Symbol	Parameter	Min	Max	Units	Notes	Figure
	IOCHRDY					
t4a	IOCHRDY Propagation Delay from IORC#/ IOWC# Active		55 (75)	ns	EPP	82, 83
t4b	IOCHRDY Propagation Delay from BUSY		34 (65)	ns	EPP	82, 83
	IORC#					
t5a	IORC# Active Pulse Width	90		ns		72
t5b	IORC# Recovery Time	60		ns		72
	IOWC#					
t6a	IOWC# Active Pulse Width	90		ns		73
t6b	IOWC# Recovery Time	60		ns		73
	AEN					
t7a	AEN Setup to IORC#/IOWC# Active	18		ns		72, 73
t7b	AEN Hold from IORC#/IOWC# Inactive	0		ns		72, 73
	NOWS#					
t8a	NOWS# Delay from IORC#/IOWC#		35 (50)	ns		72, 73
	TC					
t9a	TC Active Pulse Width	50		ns	6	74
	RESET					
	RSTDRV					
t10a	RSTDRV Active Pulse Width	0.5		μs		75
t10b	Hardware Configuration Input Setup to RSTDRV Inactive	100		ns	All Configuration Modes	76
t10c	Hardware Configuration Input Hold from RSTDRV Inactive	0			All Configuration Modes	76
	INTERRUPTS					
	RQ[4,3] (Serial Ports)					
t11b	IRQ[4,3] Inactive Delay from IORC#/ IOWC# Active		100	ns	THR wr, RBR rd, MSR rd	90, 91
t11c	IRQ[4,3] Inactive Delay from IORC# Inactive		100	ns	IIR rd, LSR rd	90
t11d	IRQ[4,3] Active Delay from DCD#/DSR#/ CTS#/RI#		80	ns		91

Table 43. AC Specifications (V_{CC} = 5V ± 10%, or (3.3V ± 0.3V) T_{amb} = 0°C to 70°C) (Continued)

Symbol	Parameter	Min	Max	Units	Notes	Figure
INTERRUPTS						
IRQ[7,5] (Parallel Port)						
t12b	IRQ[7,5] Inactive Delay from IORC#/ IOWC# Active		70 (90)	ns	ECP rev, fwd to FIFO	81
t12c	IRQ[7,5] Inactive Delay from IOWC# Inactive		70 (95)	ns	ECP fwd to ECR	81
t12d	IRQ[7,5] Delay from ACK#		70 (90)	ns	All Modes	81
t12e	IRQ[7,5] Delay from FAULT#		70 (90)	ns	ECP	81
IRQ6 (FDC)						
t13b	IRQ6 Inactive Delay from IORC#/ IOWC# Active		t1e+125	ns	2	80
DMA						
FDDREQ, PPDREQ						
t14a	xDREQ Inactive Delay from xDACK# Active		75 (100)	ns	4	74
t14b	FDREQ Cycle Time (Non-Burst DMA)	6.25		µs	3	74
t14c	xDREQ Active from IORC#/IOWC# Inactive	100		ns		74
t14d	xDREQ Setup IORC#/IOWC#	0		ns	3	74
t14e	xDREQ Delay from IORC#/IOWC# Active		75 (100)	ns	5	74
t14f	FDREQ Inactive Delay from TC Active PPDREQ Inactive Delay from TC Active		110 80 (90)	ns		74
t14g	xDREQ to xDACK# Inactive	⅔ t1e				74
FDDACK#, PPDACK#						
t15a	xDACK# Active Delay from xDREQ Active	0		ns		74
t15b	xDACK# Setup to IORC#/IOWC# Active	18		ns		74
t15c	xDACK# Hold from IORC#/IOWC# Inactive	0		ns		74

5

Table 43. AC Specifications (V_{CC} = 5V ± 10%, or (3.3V ± 0.3V) T_{amb} = 0°C to 70°C) (Continued)

Symbol	Parameter	Min	Max	Units	Notes	Figure
	PARALLEL PORT					
	PD[7:0]					
t16a	PD[7:0] Delay from IOWC# Inactive		60 (90)	ns	ISA,PS/2 wr	87
t16b	PD[7:0] Delay from IOWC# Active		70 (100)	ns	EPP wr	82
t16c	PD[7:0] Float Delay from AUTOFD#/SELECTIN# Inactive	50		ns	EPP wr	82
t16d	PD[7:0] Delay from IORC# Active		70 (100)	ns	EPP rd	83
t16e	PD[7:0] Float Delay from AUTOFD#/SELECTIN# Inactive	50		ns	EPP rd	83
t16f	PD[7:0] Setup to STROBE# Active	450			ISA FIFO	84
t16g	PD[7:0] Hold from STROBE# Inactive	450			ISA FIFO	84
t16h	PD[7:0] Hold from BUSY Inactive	0			ECP fwd	85
t16i	PD[7:0] Setup to ACK# High	0			ECP rev	86
t16j	PD[7:0] Hold from AUTOFD# Low	0			ECP rev	86
	STROBE#					
t17a	STROBE# Delay from IOWC# Inactive		60/ 90		ISA, PS/2	87
t17b	STROBE# Delay from IORC#/IOWC# Active		60/ 90		EPP	82, 83
t17c	STROBE# Active from BUSY Inactive	500			ISA FIFO	84
t17d	STROBE Active Pulse Width	450			ISA FIFO	84
t17e	STROBE# Active from BUSY Inactive	0			ECP fwd	85
t17f	STROBE# Inactive Delay from BUSY Active	0			ECP fwd	85
	AUTOFD#					
t18a	AUTOFD# Delay from IOWC# Inactive		60 (90)	ns	ISA,PS/2	82, 87
t18b	AUTOFD# Delay from IORC#/IOWC# Active		60 (90)	ns	EPP	82, 83
t18c	AUTOFD# Hold from BUSY Inactive	80		ns	ECP fwd	85
t18d	AUTOFD# Low Delay from ACK# Inactive	0		ns	ECP rev	86
t18e	AUTOFD# High Delay from ACK# Active	0		ns	ECP rev	86
	INIT#					
t19a	INIT# Delay from IOWC# Inactive		60 (90)	ns	All Modes	87

Table 43. AC Specifications (V_{CC} = 5V ± 10%, or (3.3V ± 0.3V) T_{amb} = 0°C to 70°C) (Continued)

Symbol	Parameter	Min	Max	Units	Notes	Figure
	SELECTIN#					
t20a	SELECTIN# Delay from IOWC#/IORC# Inactive		60 (90)	ns	ISA, PS/2	82, 83, 87
t20b	SELECTIN# Delay from IOWC#/IORC# Active		60 (90)		EPP	82, 83
	BUSY					
t21a	BUSY Active Delay from STROBE# Active		500		ISA, PS/2	84
t21b	BUSY Active Delay from STROBE# Active	0				85
t21c	BUSY Inactive Delay from STROBE# Inactive	0			ECP fwd	85
t21d	BUSY Setup to ACK# Active	0			ECP rev	86
t21f	BUSY Hold from AUTOFD# Inactive	0			ECP rev	86
	ACK#					
t22a	ACK# Active Hold from AUTOFD# High	0			ECP rev	86
t22b	ACK# Inactive Hold from AUTOFD# Low	0			ECP rev	86
	PPDIR/GCS#					
t23a	GCS# Delay from SA[10:0]		60 (90)			89
t23b	PPDIR Delay from IOWC# Inactive		60 (90)		ISA, PS/2, ECP	87
t23c	PPDIR Delay from IOWC# Active		60 (90)		EPP	82
	IDE Interface					
	IDECS[1:0]#					
t24a	IDECSx# Delay from SA[10:0]		40 (70)			88
	DEN#					
t25a	DEN# Delay from SA[10:0]		40 (70)			72, 73, 88
t25b	DEN# Delay from xDACK#		40 (70)			74
	HEN#					
t26a	HEN# Delay from IO16#		35 (65)			88

5

Table 43. AC Specifications (V_{CC} = 5V ± 10%, or (3.3V ± 0.3V) T_{amb} = 0°C to 70°C) (Continued)

Symbol	Parameter	Min	Max	Units	Notes	Figure
	SERIAL PORTS					
	DTRx#, RTSx#, DCDx#					
t27a	DTRx#/RTSx#/DCDx# Active Delay from IOWC#		55 (70)	ns	MCR wr	91
	FLOPPY DISK CONTROLLER					
	RDDATA#					
t28a	Read Data Pulse Width	50		ns		95
t28c	PLL Data Rate		1M	bits/sec		na
t28d	Lockup Time		64	t28c		na
	WRDATA#					
t29a	Data Width	see note	see note		7	77
	HDSEL#					
t30a	WE# to HDSEL# Change	see note	see note		10	78
	STEP#					
t31a	STEP# Active Time	2.5		µs		78
t31b	STEP# Cycle Time	see note	see note	µs	9	78
	DIR#					
t32a	DIR# Setup to STEP# Active	1		µs	8	78
t32b	DIR# Hold from STEP# Inactive	10		µs		78
	WE#					
t33a	WE# Inactive Delay from RSTDRV Inactive Edge		2	µs		75
	INDEX#					
t34a	INDEX# Pulse Width	5	t1e			78

NOTES:
1. The FDC Status Register's status bits which are not latched may be updated during a host read operation.
2. The timing t13b is specified for the FDC interrupt signal in the polling mode only. These timings in case of the result phase of the read and write commands are microcode dependent.
3. This timing is for FDC FIFO threshold = 1. When FIFO threshold is N bytes, the value should be multiplied by N and subtract 1.5 µs. The value shown is for 1 Mbps, scales linearly with data rate.
4. This timing is a function of the internal clock period (t1e) and is given as (⅔) t1e. The values of t1e are shown in Note 3.
5. If DACK# transitions before RD#, then this specification is ignored. If there is no transition on DACK#, then this becomes the DRQ inactive delay.
6. TC width is defined as the time that both TC and DACK# are active. Note that TC and DACK# must overlap at least 50 ns.

NOTES: (Continued)
7. Based on the internal clock period (t1e). For various data rates, the read and write data width minimum values are:

Disk Drive Data Rate	24 MHz
1 Mbps	150 ns
500 Kbps	360 ns
300 Kbps	615 ns
250 Kbps	740 ns

8.This timing is a function of the selected data rate as follows:

Disk Drive Data Rate	Timing
1 Mbps	1.0 μs Min
500 Kbps	2.0 μs Min
300 Kbps	3.3 μs Min
250 Kbps	4.0 μs Min

9. This value can range from 0.5 ms to 8.0 ms and is dependent upon data rate and the Specify Command value.
10. The minimum MFM values for WE# to HDSEL# change for the various data rates are:

Disk Drive Data Rate	Min MFM Value
1 Mbps	0.5 ms + [8 × GPL]
500 Kbps	1.0 ms + [16 × GPL]
300 Kbps	1.6 ms + [26.66 × GPL]
250 Kbps	2.0 ms + [32 × GPL]

GPL is the size of gap 3 defined in the sixth byte of a Write Command.

11. Based on internal clock period.
12. Jitter tolerance is defined as:
    (Maximum bit shift from nominal position ÷ ¼ period of nominal data rate) × 100 percent is a measure of the allowable bit jitter that may be present and still be correctly detected. The data separator jitter tolerance is measured under dynamic conditions that jitters the bit stream according to a reverse precompensation algorithm.
13. The minimum reset active period for a software reset is dependent on the data rate, after the FDC module has been properly reset using the t10a spec. The minimum software reset period then becomes:

Disk Drive Data Rate	Minimum Software Reset Active Period
	24 MHz
1 Mbps	125 ns
500 Kbps	250 ns
300 Kbps	420 ns
250 Kbps	500 ns

11.4.1 CLOCK TIMINGS

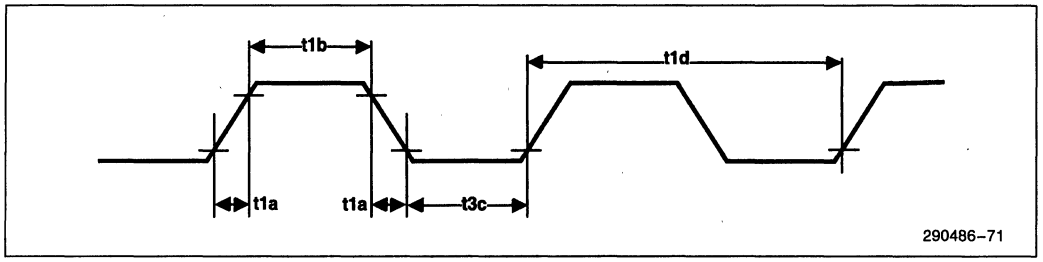


Figure 71. Clock Timing

11.4.2 HOST TIMINGS

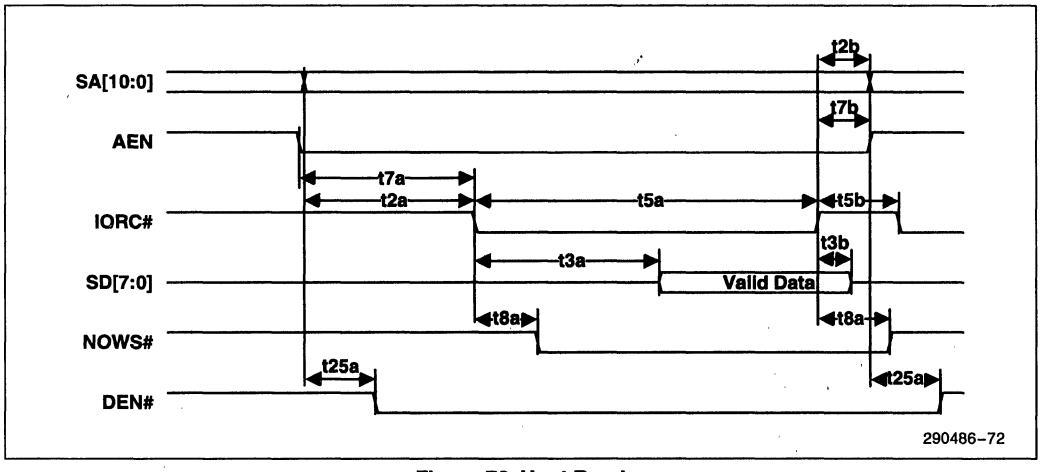


Figure 72. Host Read

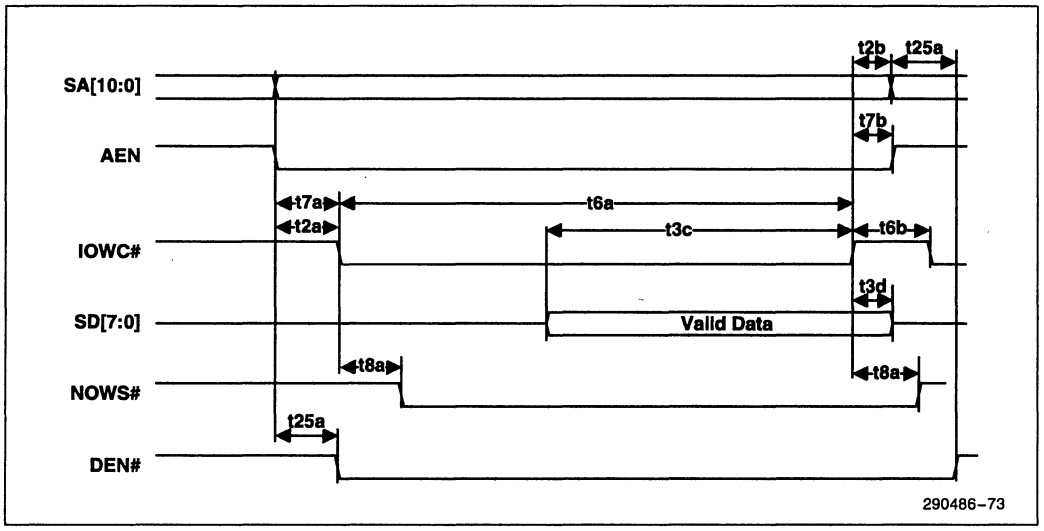


Figure 73. Host Write

5

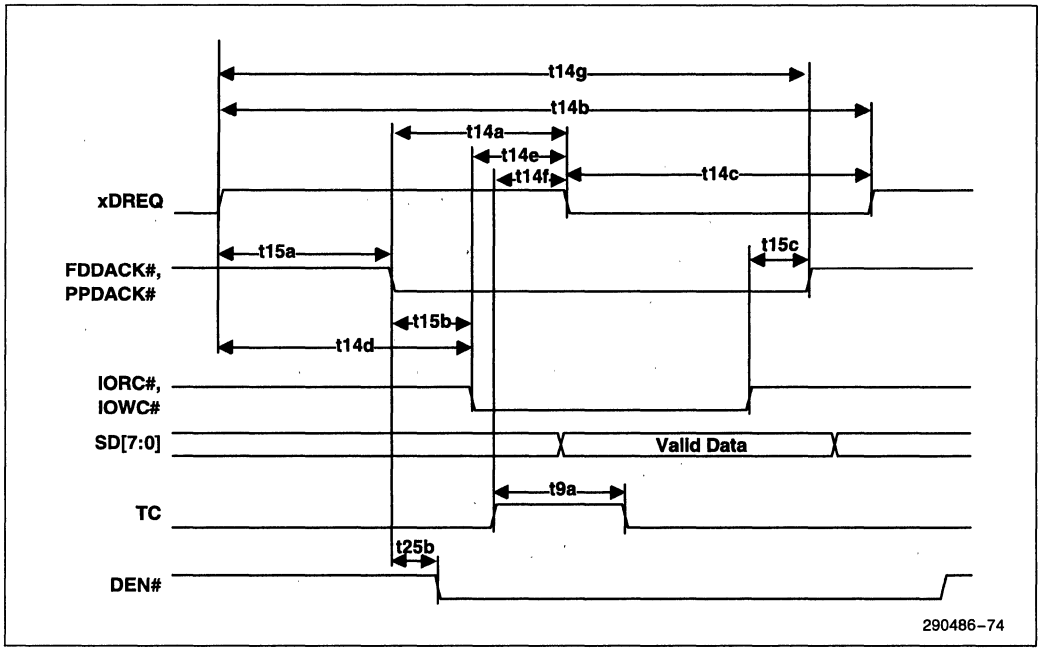
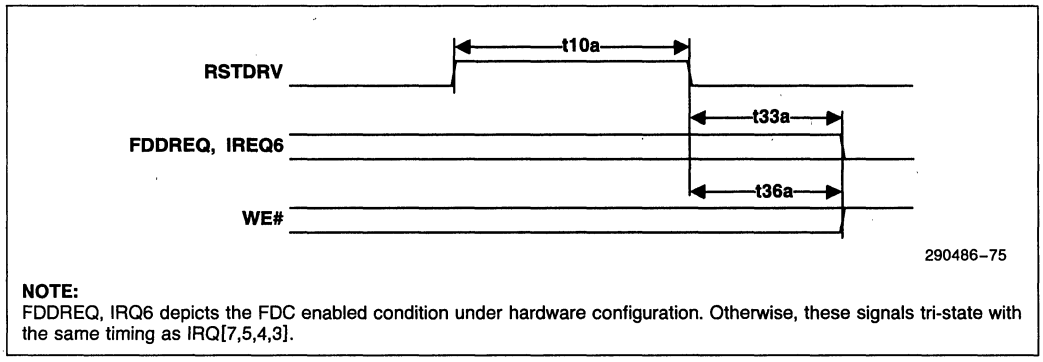


Figure 74. DMA Timing



NOTE:
FDDREQ, IRQ6 depicts the FDC enabled condition under hardware configuration. Otherwise, these signals tri-state with the same timing as IRQ[7,5,4,3].

Figure 75. Reset Timing

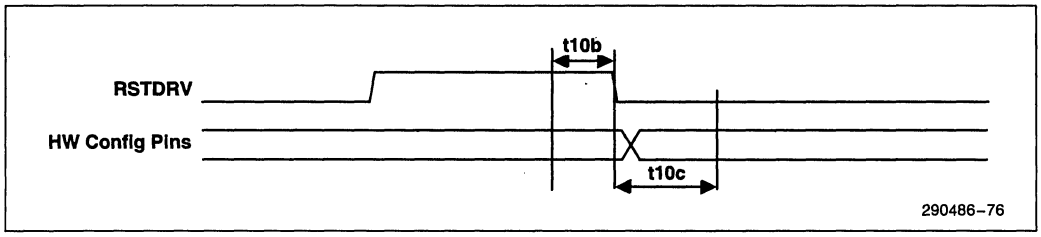


Figure 76. Reset Timing (Hardware Extended Configuration Mode)

11.4.3 FDC TIMINGS

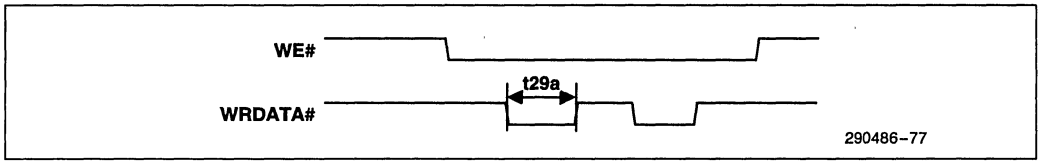
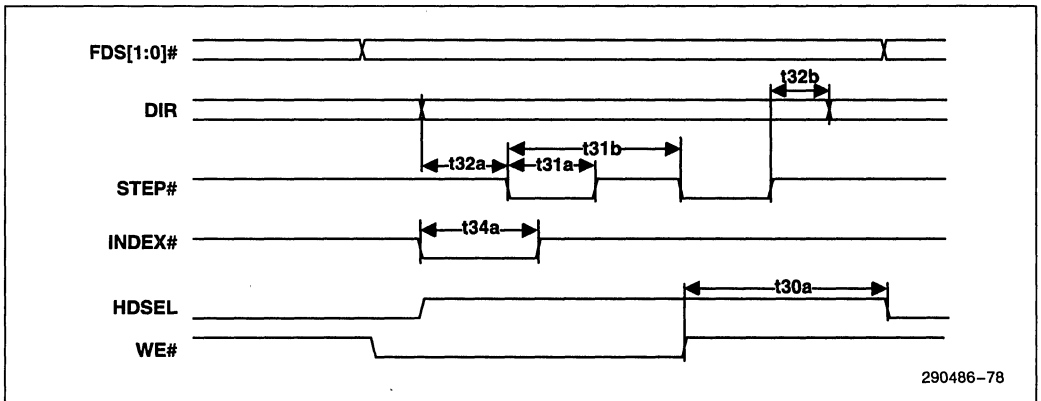


Figure 77. Write Data Timing



NOTE:
For overlapped seeks, only one step pulse per drive selection is issued. Non-overlapped seeks will issue all programmed step pulses.

Figure 78. FDC Drive Control/Timing

5

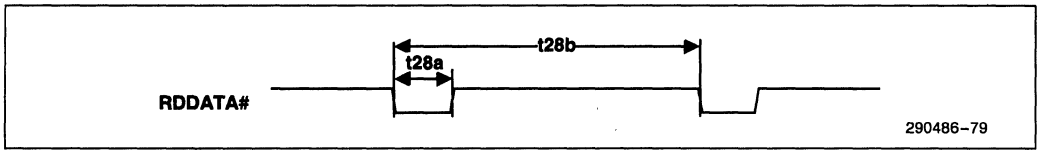


Figure 79. FDC Internal PLL Timing

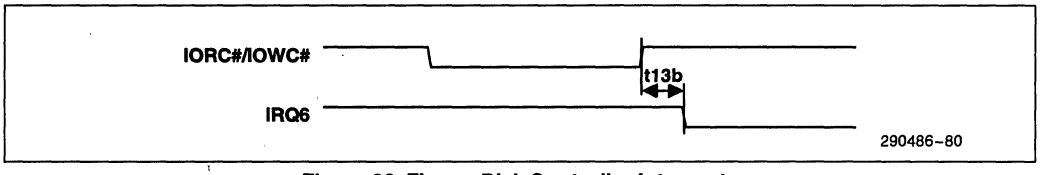


Figure 80. Floppy Disk Controller Interrupts

11.4.4 PARALLEL PORT TIMINGS

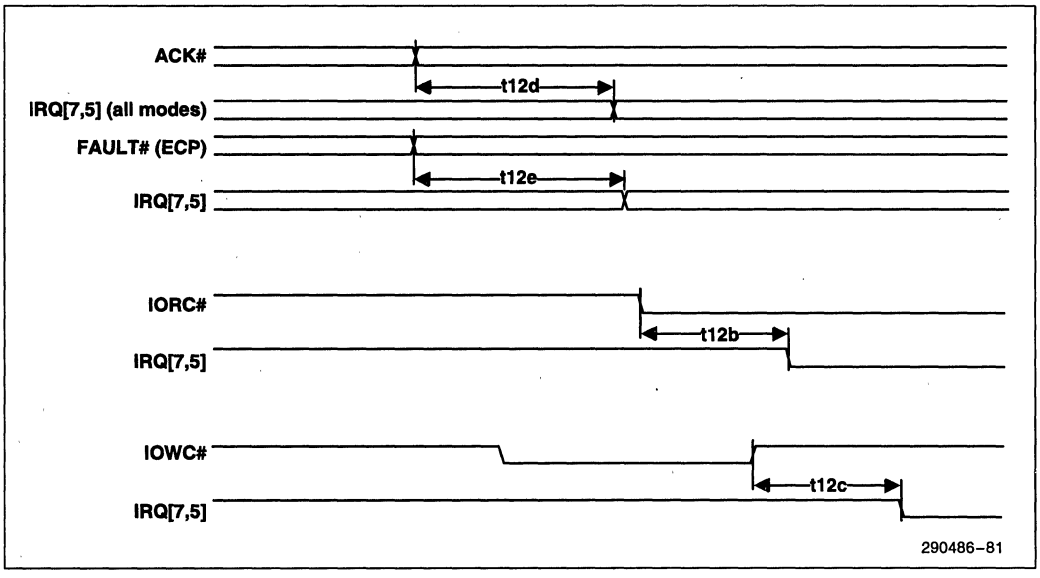


Figure 81. Parallel Port Interrupt Timing

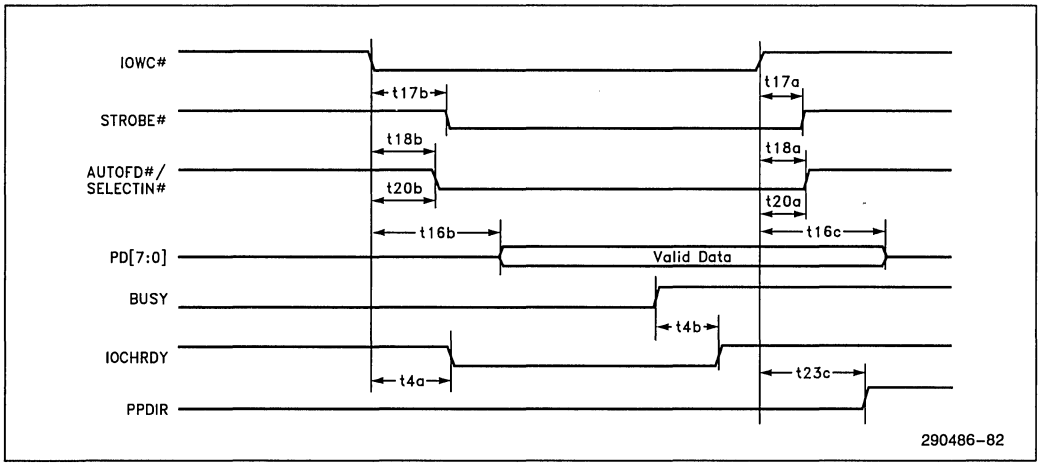


Figure 82. EPP Write Timing

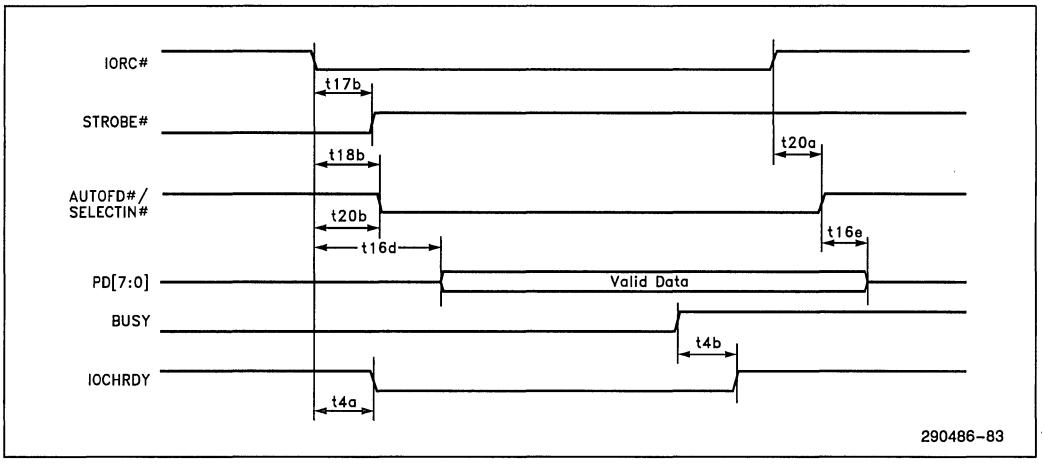


Figure 83. EPP Read Timing

5

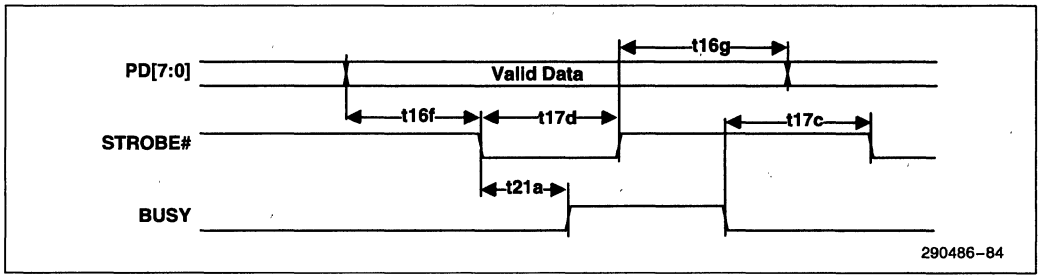


Figure 84. ISA-Compatible FIFO Timing

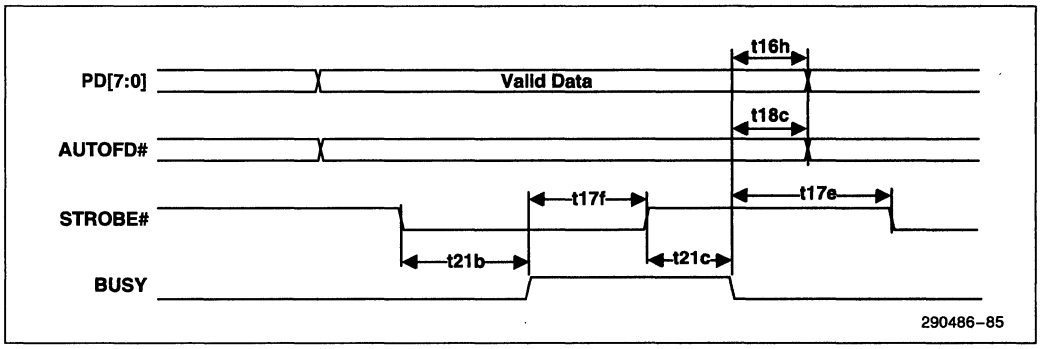


Figure 85. ECP Write Timing (Forward Direction)

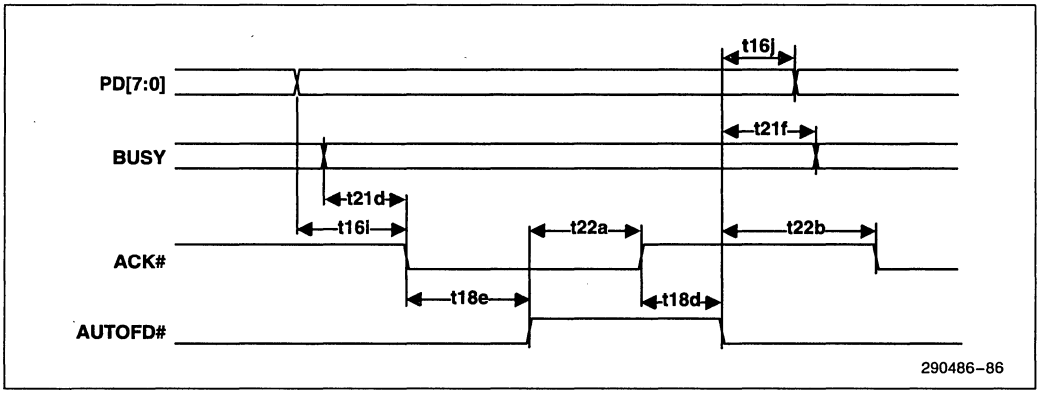


Figure 86. ECP Read Timing (Reverse Direction)

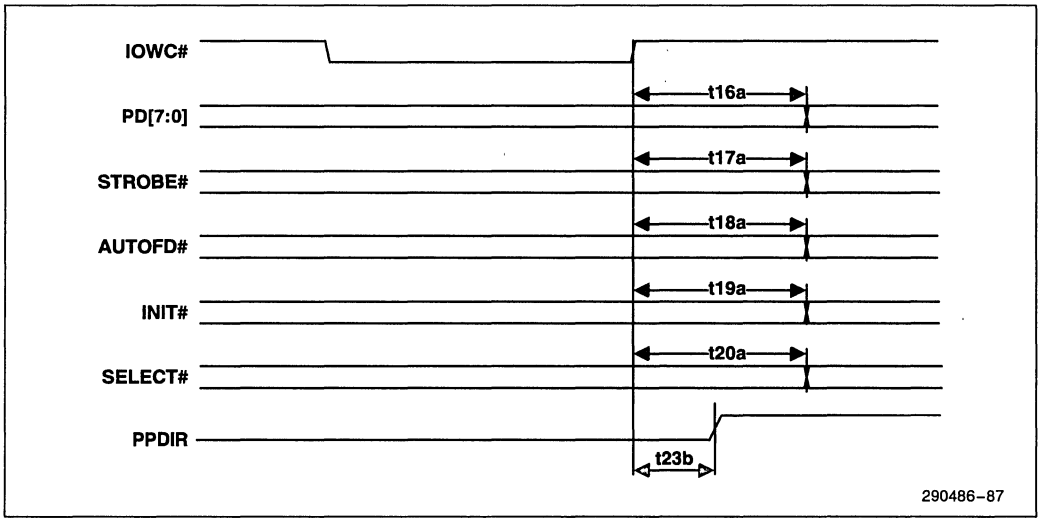


Figure 87. ISA-Compatible Write Timing

5

11.4.5 IDE TIMINGS

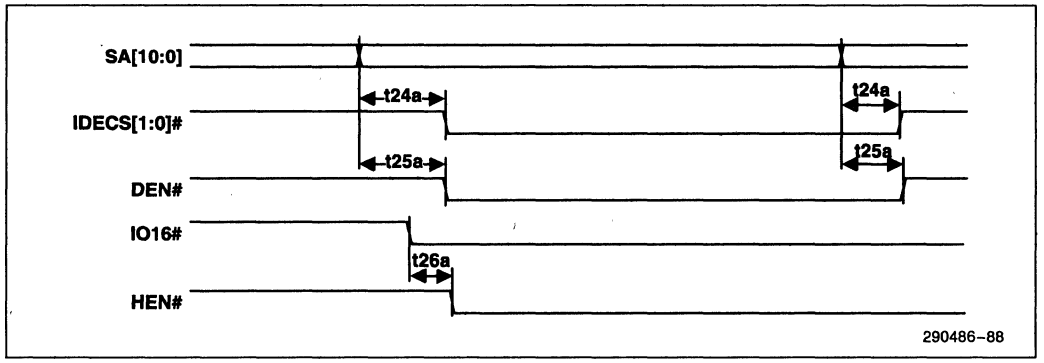


Figure 88. IDE Timing

11.4.6 GAME PORT TIMINGS

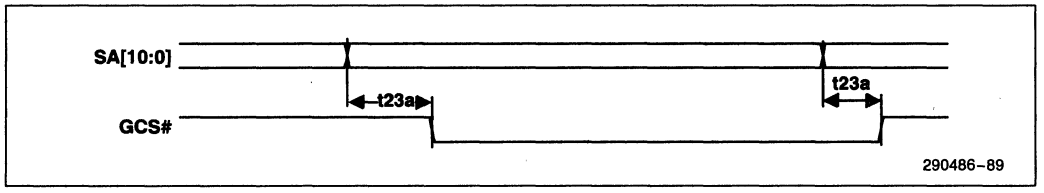


Figure 89. Game Port Timing

11.4.7 SERIAL PORT TIMINGS

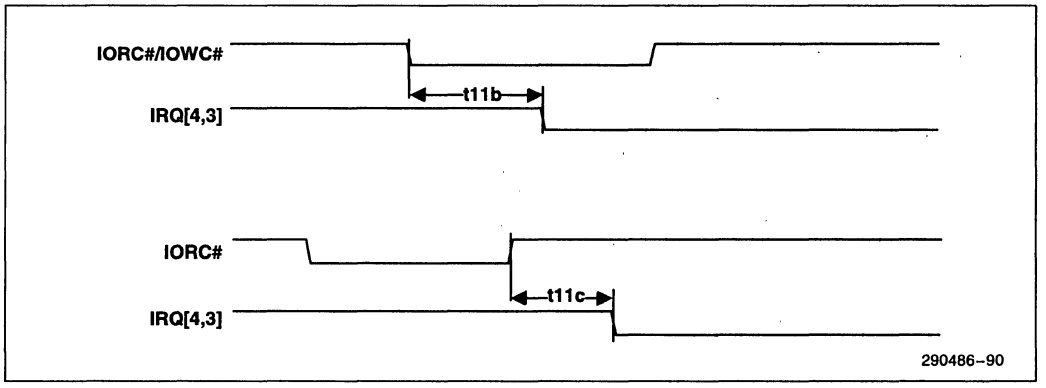


Figure 90. Serial Port Interrupt Timing

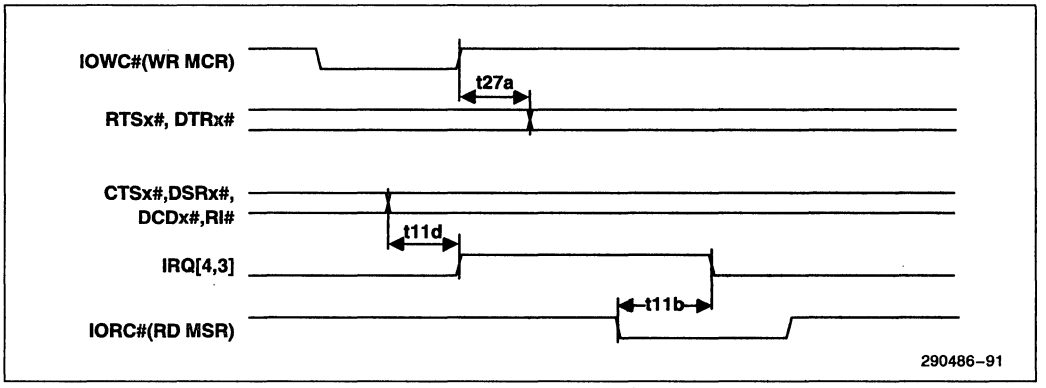


Figure 91. Modem Control Timing

5

12.0 PINOUT AND PACKAGE INFORMATION

12.1 Pin Assignment

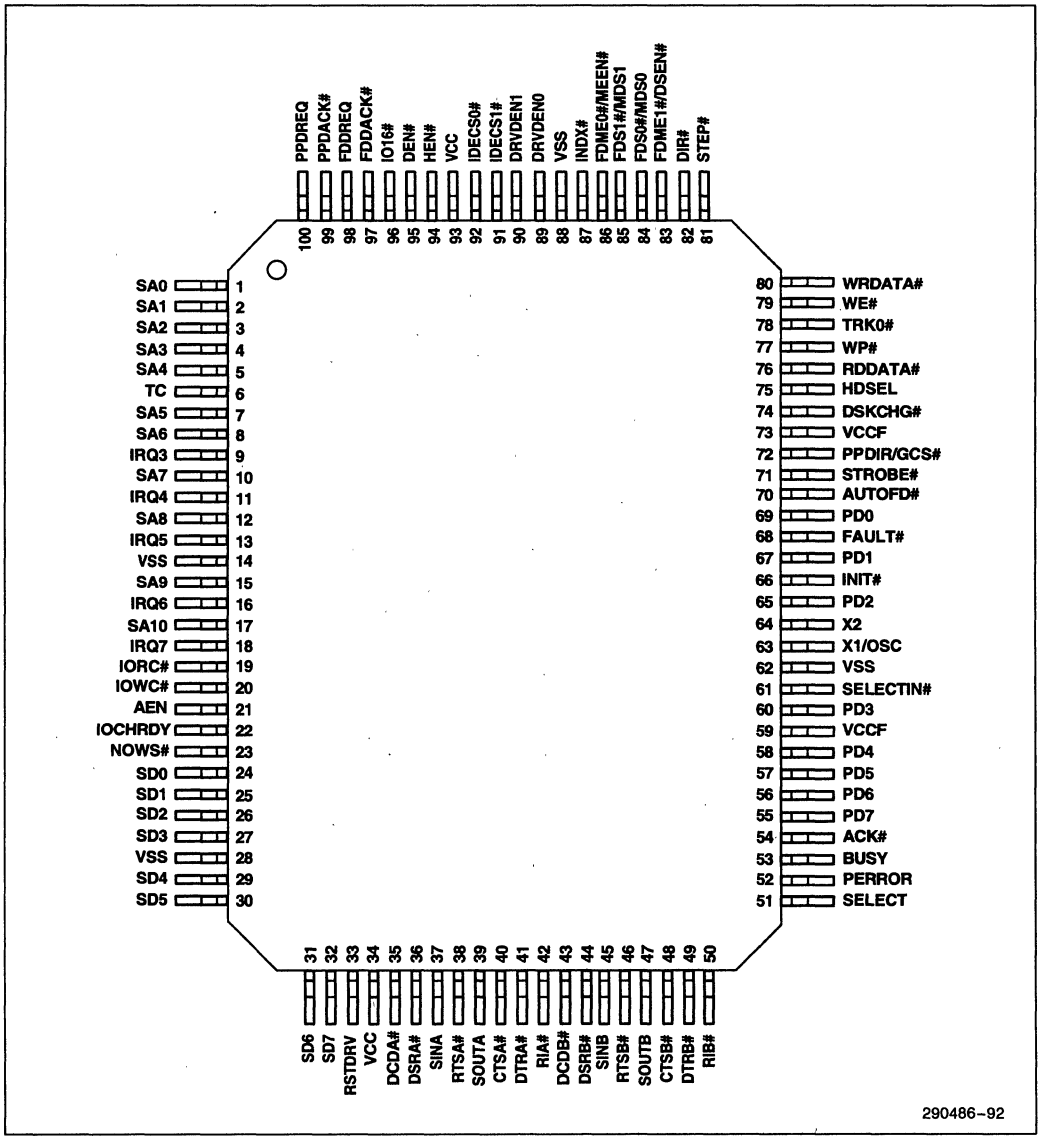


Figure 92. 82091AA Pin Diagram

Table 44. Alphabetical 82091AA Pin Assignment

Signal Name	Pin #	Type
ACK#	54	I
AEN	21	I
AUTOFD#	70	O
BUSY	53	I
CTSA#	40	I
CTSB#	48	I
DCDA#	35	O
DCDB#	43	O
DEN#	95	I/O
DIR#	82	O
DRVDEN0	89	O
DRVDEN1	90	O
DSKCHG#	74	I
DSRA#	36	I
DSRB#	44	I
DTRA#	41	I/O
DTRB#	49	I/O
FAULT#	68	I
FDDACK#	97	I
FDDREQ	98	O
FDME0#/MEEN#	86	O
FDME1#/DSEN#	83	O
FDS0#/MDS0	84	O
FDS1#/MDS1	85	O
HDSEL	75	O
HEN#	94	I/O
IDECS0#	92	I/O
IDECS1#	91	I/O
INDX#	87	I
INIT#	66	O
IO16#	96	I
IOCHRDY	22	O
IORC#	19	I
IOWC#	20	I

Signal Name	Pin #	Type
IRQ3	9	O
IRQ4	11	O
IRQ5	13	O
IRQ6	16	O
IRQ7	18	O
NOWS#	23	O
PD0	69	I/O
PD1	67	I/O
PD2	65	I/O
PD3	60	I/O
PD4	58	I/O
PD5	57	I/O
PD6	56	I/O
PD7	55	I/O
PERROR	52	I
PPDACK#	99	I
PPDREQ	100	O
PPDIR/GCS#	72	I/O
RDDATA#	76	I
RIA#	42	I
RIB#	50	I
RSTDRV	33	I
RTSA#	38	I/O
RTSB#	46	I/O
SA0	1	I
SA1	2	I
SA2	3	I
SA3	4	I
SA4	5	I
SA5	7	I
SA6	8	I
SA7	10	I
SA8	12	I
SA9	15	I

Table 44. Alphabetical 82091AA Pin Assignment (Continued)

Signal Name	Pin #	Type
SA10	17	I
SD0	24	I/O
SD1	25	I/O
SD2	26	I/O
SD3	27	I/O
SD4	29	I/O
SD5	30	I/O
SD6	31	I/O
SD7	32	I/O
SINA	37	I
SINB	45	I
SELECT	51	I
SELECTIN#	61	O
SOUTA	39	I/O
SOUTB	47	I/O
STEP#	81	O

Signal Name	Pin #	Type
STROBE#	71	O
TC	6	I
TRK0#	78	I
V_{CC}	34	V
V_{CC}	93	V
V_{CCF}	59	V
V_{CCF}	73	V
V_{SS}	14	V
V_{SS}	28	V
V_{SS}	62	V
V_{SS}	88	V
WE#	79	O
WP#	77	I
WRDATA#	80	O
X1/OSC	63	I
X2	64	I

Table 45. Numerical 82091AA Pin Assignment

Pin #	Signal Name	Type
1	SA0	I
2	SA1	I
3	SA2	I
4	SA3	I
5	SA4	I
6	TC	I
7	SA5	I
8	SA6	I
9	IRQ3	O
10	SA7	I
11	IRQ4	O
12	SA8	I
13	IRQ5	O
14	V_{SS}	V
15	SA9	I

Pin #	Signal Name	Type
16	IRQ6	O
17	SA10	I
18	IRQ7	O
19	IORC#	I
20	IOWC#	I
21	AEN	I
22	IOCHRDY	O
23	NOWS#	O
24	SD0	I/O
25	SD1	I/O
26	SD2	I/O
27	SD3	I/O
28	V_{SS}	V
29	SD4	I/O
30	SD5	I/O

Table 45. Numerical 82091AA Pin Assignment (Continued)

Pin #	Signal Name	Type
31	SD6	I/O
32	SD7	I/O
33	RSTDRV	I
34	V_{CC}	V
35	DCDA#	O
36	DSRA#	I
37	SINA	I
38	RTSA#	I/O
39	SOUTA	I/O
40	CTSA#	I
41	DTRA#	I/O
42	RIA#	I
43	DCDB#	O
44	DSRB#	I
45	SINB	I
46	RTSB#	I/O
47	SOUTB	I/O
48	CTSB#	I
49	DTRB#	I/O
50	RIB#	I
51	SELECT	I
52	PERROR	I
53	BUSY	I
54	ACK#	I
55	PD7	I/O
56	PD6	I/O
57	PD5	I/O
58	PD4	I/O
59	V_{CCF}	V
60	PD3	I/O
61	SELECTIN#	O
62	V_{SS}	V
63	X1/OSC	I
64	X2	I
65	PD2	I/O

Pin #	Signal Name	Type
66	INIT#	O
67	PD1	I/O
68	FAULT#	I
69	PD0	I/O
70	AUTOFD#	O
71	STROBE#	O
72	PPDIR/GCS#	I/O
73	V_{CCF}	V
74	DSKCHG#	I
75	HDSEL	O
76	RDDATA#	I
77	WP#	I
78	TRK0#	I
79	WE#	O
80	WRDATA#	O
81	STEP#	O
82	DIR#	O
83	FDME1#/DSEN#	O
84	FDS0#/MDS0	O
85	FDS1#/MDS1	O
86	FDME0#/MEEN#	O
87	INDX#	I
88	V_{SS}	V
89	DRVDEN0	O
90	DRVDEN1	O
91	IDECS1#	I/O
92	IDECS0#	I/O
93	V_{CC}	I/O
94	HEN#	V
95	DEN#	I/O
96	IO16#	I
97	FDDACK#	I
98	FDDREQ	O
99	PPDACK#	I
100	PPDREQ	O

12.2 Package Characteristics

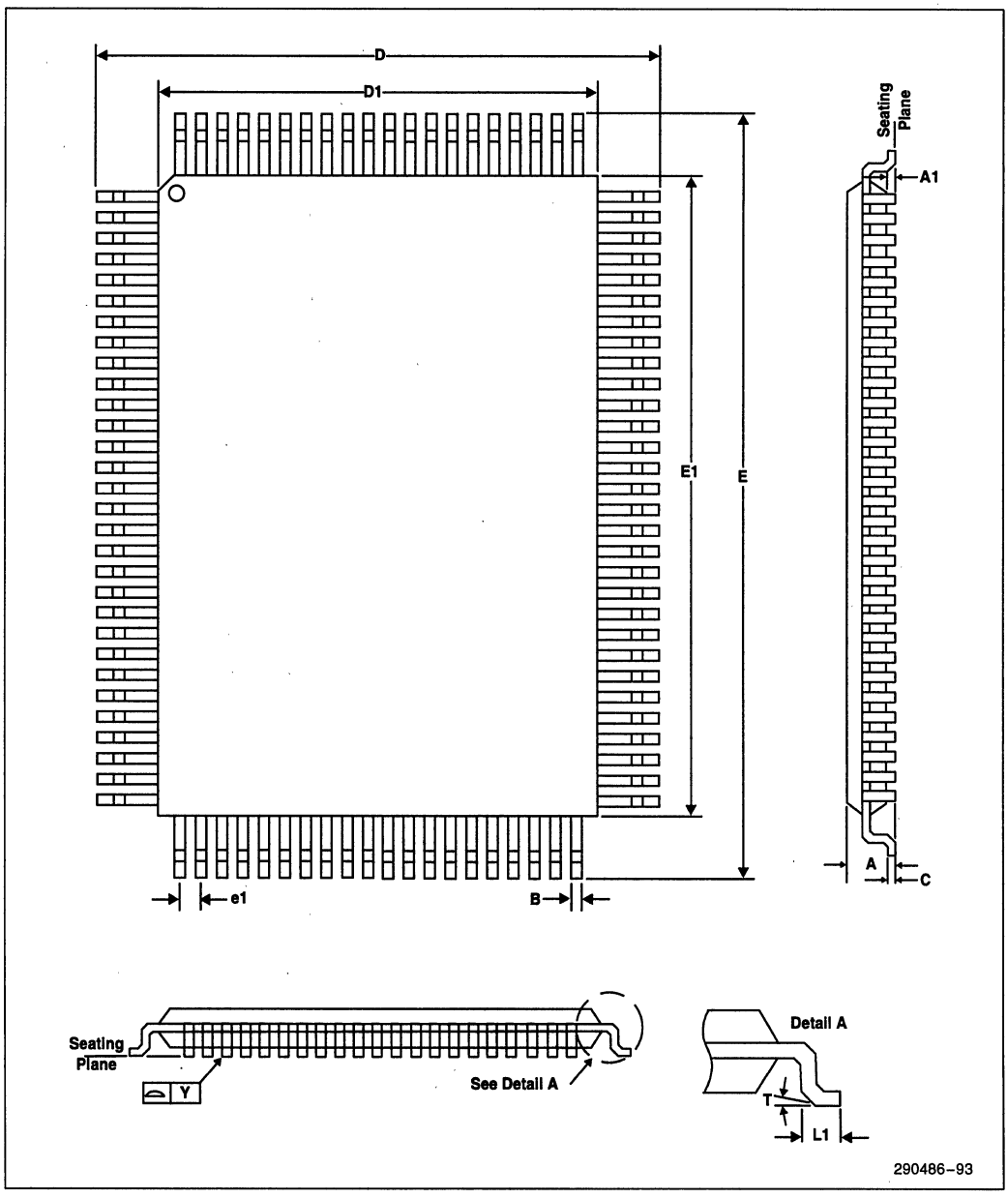


Figure 93. 100-Pin Quad Flat Pack (QFP) Dimensions

Quad Flat Pack Package				
Symbol	Millimeters			
	Minimum	Nominal	Maximum	Notes
A			3.15	
A1	0.0			
B	0.20	0.30	0.40	
C	0.10	0.15	0.20	
D	17.5	17.9	18.3	
D1		14.0		
E	23.5	23.9	24.3	
E1		20.0		
e1	0.53	0.65	0.77	
L1	0.60	0.80	1.00	
N		100		Rectangle
T	0.00		10.0	
Y			0.10	
ISSUE	JEDEC			

13.0 DATA SEPARATOR CHARACTERISTICS FOR FLOPPY DISK MODE

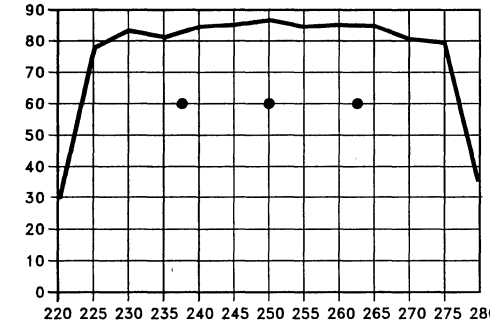


Figure 94. Typical Jitter Tolerance vs Data Rate (Capture Range 250 Kbps)

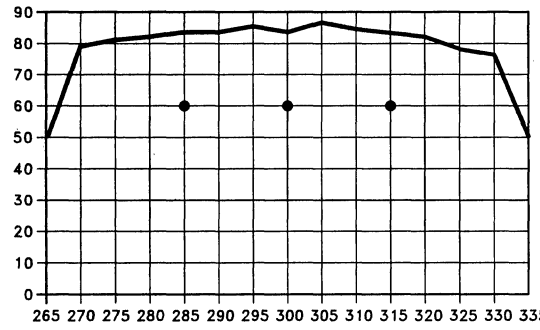


Figure 95. Typical Jitter Tolerance vs Data Rate (Capture Range 300 Kbps)

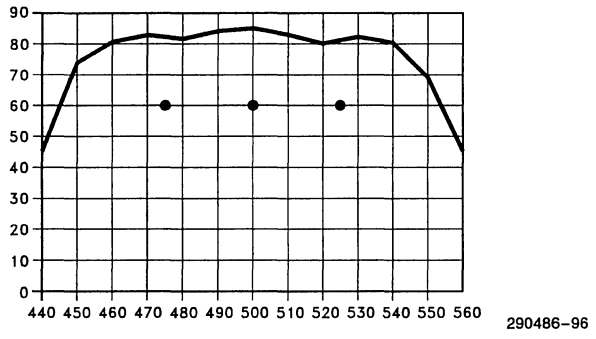


Figure 96. Typical Jitter Tolerance vs Data Rate (Capture Range 500 Kbps)

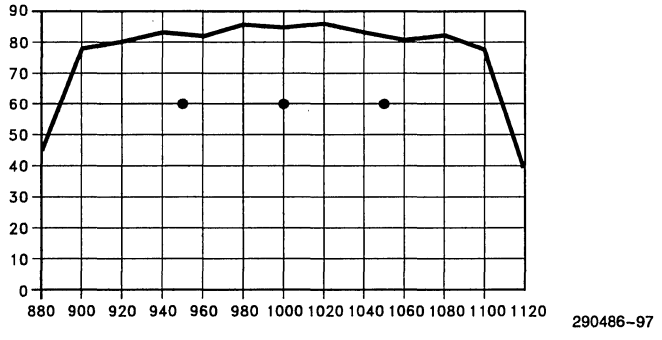


Figure 97. Typical Jitter Tolerance vs Data Range (Capture Range 1 Mbps)

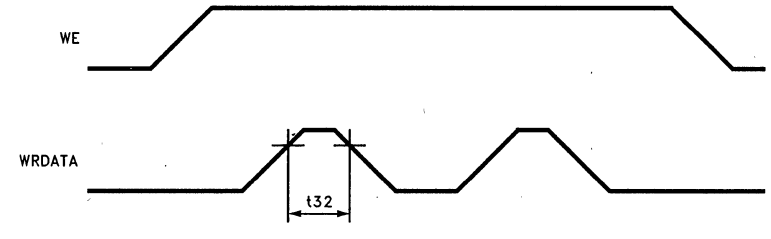
Jitter Tolerance measured in percent. Capture range expressed as a percent of data rate, i.e., ±3% percent.

• = Test Points: 250 Kbps, 300 Kbps, 500 Kbps and 1 Mbps are center, ±5 percent @ 60 percent jitter.

Test points are tested at temparture and V_{CC} limits. Refer to the datasheet. Typical conditions are: room temperature, nominal V_{CC}.

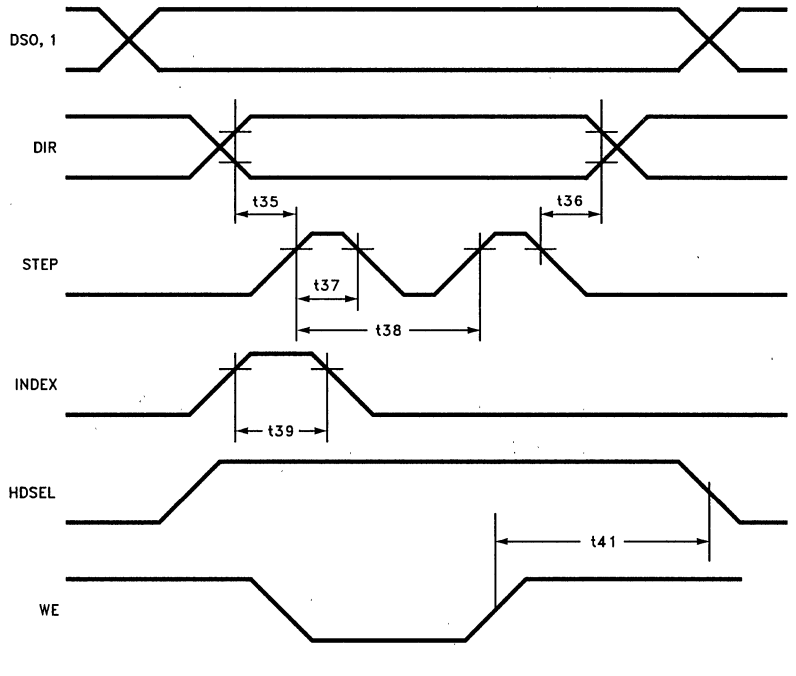
5

13.1 Write Data Timing



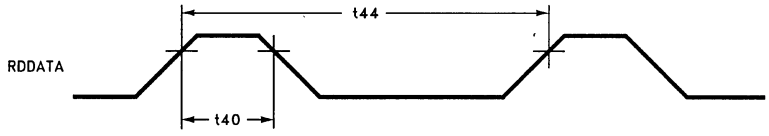
NOTE:
Invert high.

13.2 Drive Control



NOTE:
For overlapped seeks, only one step pulse per drive selection is issued. Non-overlapped seeks will issue all programmed step pulses. Invert high.

13.3 Internal PLL



290486-A0

NOTE:
Invert high.

APPENDIX A
FDC FOUR DRIVE SUPPORT

Section 8.0 of this document completely describes the FDC when the module is configured for two drive support. In addition, the FDC commands in Section 8.0 provide four drive support information. This appendix provides additional information concerning four drive support. The signal pins that are affected by four drive support are described in Section A.1. Note that the FDC signals not discussed in this appendix operate the same for both two and four drive systems. The following registers are described in this appendix; Digital Output Register (DOR), Enhanced Tape Drive Register (TDR), and the Main Status Register (MSR). Some bits in these registers operate differently in a four drive configuration than a two drive configuration.

NOTES:

- The descriptions in this appendix assume that four floppy drive support has been selected by setting FDDQTY to 1 in the AIPCFG1 Register.
- Only drive 0 or drive 1 can be selected as the boot drive.

A.1 Floppy Disk Controller Interface Signals

These signal descriptions are for a four drive system (FDDQTY = 1 in the AIPCFG1 Register). See Section 2.0 for two drive system signal descriptions.

Signal Name	Type	Description
FDME1#/DSEN#(1)	O	**FLOPPY DRIVE MOTOR ENABLE 1, or DRIVE SELECT ENABLE:** In a four drive system, this signal functions as a drive select enable (DSEN#). When DSEN# is asserted, MDS1 and MDS0 reflect the selection of the drive.
FDS1#/MDS1(1)	O	**FLOPPY DRIVE SELECT1, or MOTOR DRIVE SELECT 1:** In a four drive system, this signal functions as a motor drive select (MDS1). MDS1, together with MDS0, indicate which of the four drives is selected, as shown in note 1.
FDME0#/MEEN#(1)	O	**FLOPPY DRIVE MOTOR ENABLE 0 or MOTOR ENABLE ENABLE:** In a four drive system, this signal functions as a motor enable enable (MEEN#). MEEN# is asserted to enable the external decoding of MDS1 and MDS0 for the appropriate motor enable (see note 1).
FDS0#/MDS0(1)	O	**FLOPPY DRIVE SELECT 0 or MOTOR DRIVE SELECT 0:** In a four drive system, this signal functions as motor drive select (MDS0). MDS0, together with MDS1, indicate which of the four drives is selected as shown in note 1.

NOTE:

1. These signal pins are used to control an external decoder for four floppy disk drives as shown below. Refer to the DOR Register Description in Section A.2 for details.

MDS1	MDS0	DSEN# = 0	MEEN# = 0
0	0	Drive 0	ME0
0	1	Drive 1	ME1
1	0	Drive 2	ME2
1	1	Drive 3	ME3

A.2 DOR—Digital Output Register

I/O Address: Base + 2h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

The Digital Output Register enables/disables the floppy disk drive motors, selects the disk drives, enables/disables DMA, and provides a FDC module reset. The DOR reset bit and the Motor Enable bits have to be inactive when the 82091AA's FDC is in powerdown. The DMAGATE# and Drive Select bits are unchanged. During powerdown, writing to the DOR does not wake up the 82091AA's FDC, except for activating any of the motor enable bits. Setting the motor enable bits to 1 will wake up the module. The four internal drive select and four internal motor enable signals are encoded to a total of four output pins as described in Table 47. Figure 99 shows an example of how these four output pins can be decoded to provide four drive select and four motor enable signals. Note that only drive 0 or drive 1 can be used as the boot drive when four disk drives are enabled.

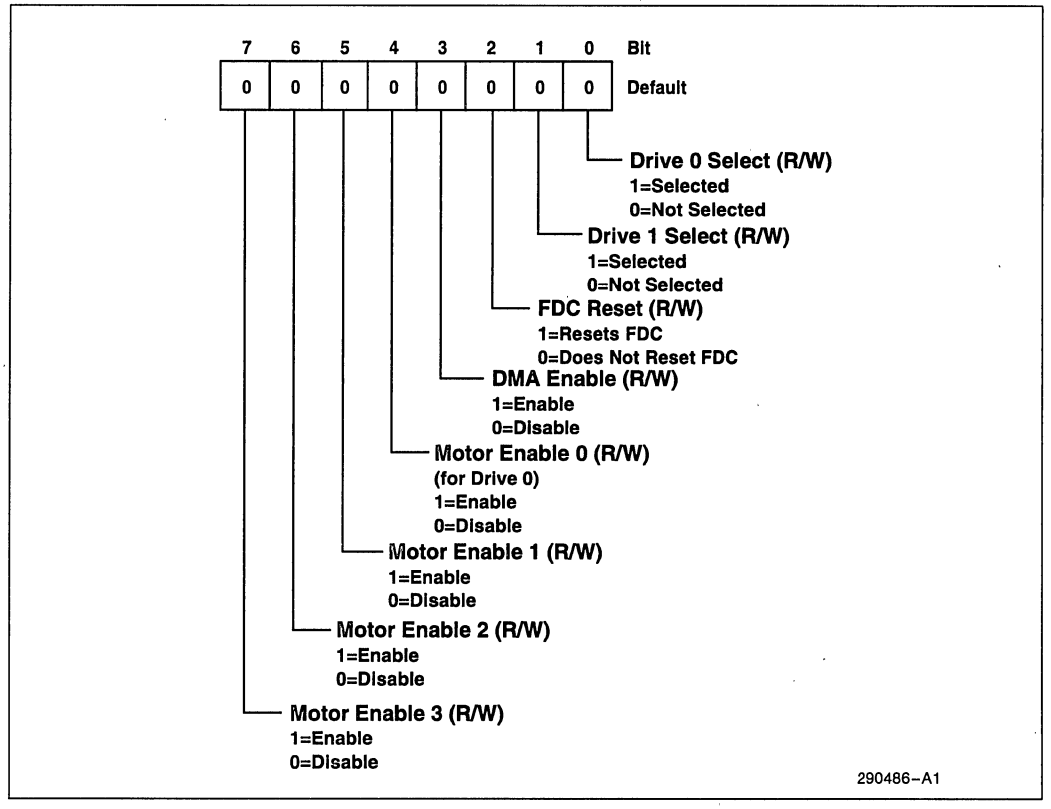


Figure 98. Digital Output Register

5

Bit	Description
7	**Motor Enable 3 (ME3):** This bit controls a motor drive enable output signal and provides the signal output for the floppy drive 3 motor (via external decoding) as shown in Table 46.
6	**Motor Enable 2 (ME2):** This bit controls a motor drive enable output signal and provides the signal output for the floppy drive 2 motor (via external decoding) as shown in Table 46.
5	**Motor Enable 1 (ME1):** This bit controls a motor drive enable signal and provides the signal output for the floppy drive 1 motor (via external decoding) as shown in Table 46.
4	**Motor Enable 0 (ME0):** This bit controls a motor drive enable signal and provides the signal output for the floppy drive 0 motor (via external decoding) as shown in Table 46.
3	**DMA Gate (DMAGATE):** This bit enables/disables DMA for the FDC. When DMAGATE = 1, DMA for the FDC is enabled. In this mode FDDREQ, TC, IRQ6, and FDDACK# are enabled. When DMAGATE = 0, DMA for the FDC is disabled. In this mode, the IRQ6 and DRQ outputs are tri-stated and the DACK# and TC inputs are disabled to the FDC. Note that the TC input is only disabled to the FDC module. Other functional units in the 82091AA (e.g., parallel port or IDE interface) can still use the TC input signal for DMA activities.
2	**FDC Reset (DORRST):** DORRST is a software reset for the FDC module. When DORRST is set to 0, the basic core of the 82091AA's FDC and the FIFO circuits are cleared conditioned by the LOCK bit in the Configure Command. This bit is set to 0 by software or a hard reset (RSTDRV asserted). The FDC remains in a reset state until software sets this bit to 1. This bit does not affect the DSR, CCR and other bits of the DOR. DORRST must be held active for at least 0.5 μs at 250 Kbps. This is less than a typical ISA I/O cycle time. Thus, in most systems consecutive writes to this register to toggle this bit allows sufficient time to reset the FDC.
1:0	**Drive Select (DS[1:0]):** This field provides the output signals to select a particular floppy drive (via external decoding) as shown in Table 47. Note that the drive motor can be enabled separately without selecting the drive. This permits the motor to come up to speed before selecting the drive. Note also that only one drive can be selected at a time. However, the drive should not be selected without enabling the appropriate drive motor via bits[7:4] of this register.

Table 46. Output Pin Status for Four Disk Drives

	FDC DOR Register Bits						Signal Pins			
Description	**ME3**	**ME2**	**ME1**	**ME0**	**DS1**	**DS0**	**MDS1#**	**MDS0#**	**DSEN#**	**MEEN#**
ME0 and DS0 enable	X	X	X	1	0	0	0	0	0	0
ME1 and DS1 enable	X	X	1	X	0	1	0	1	0	0
ME2 and DS2 enable	X	1	X	X	1	0	1	0	0	0
ME3 and DS3 enable	1	X	X	X	1	1	1	1	0	0
ME0 enable only	X	X	X	1	DS[1:0]≠00		0	0	1	0
ME1 enable only	X	X	1	0	DS[1:0]≠01		0	1	1	0
ME2 enable only	X	1	0	0	DS[1:0]≠10		1	0	1	0
ME3 enable only	1	0	0	0	DS[1:0]≠11		1	1	1	0
No ME or DS enable	0	0	0	0	X	X	1	1	1	1

NOTE:
To enable a particular drive motor and select the drive, the value for DS[1:0] must match the appropriate motor enable bit selected as indicated in the first four rows of the table. For example, to enable the drive 0 motor and select the drive, ME0 is set to 1 and DS[1:0] must be set to 00. To enable the drive motor and keep the drive de-selected the value for DS[1:0] must not match the particular motor enable as shown in the first four rows. For example, to enable the motor for drive 0 while the drive remains de-selected, ME0 is set to 1 and DS[1:0] is set to 01, 10, or 11.

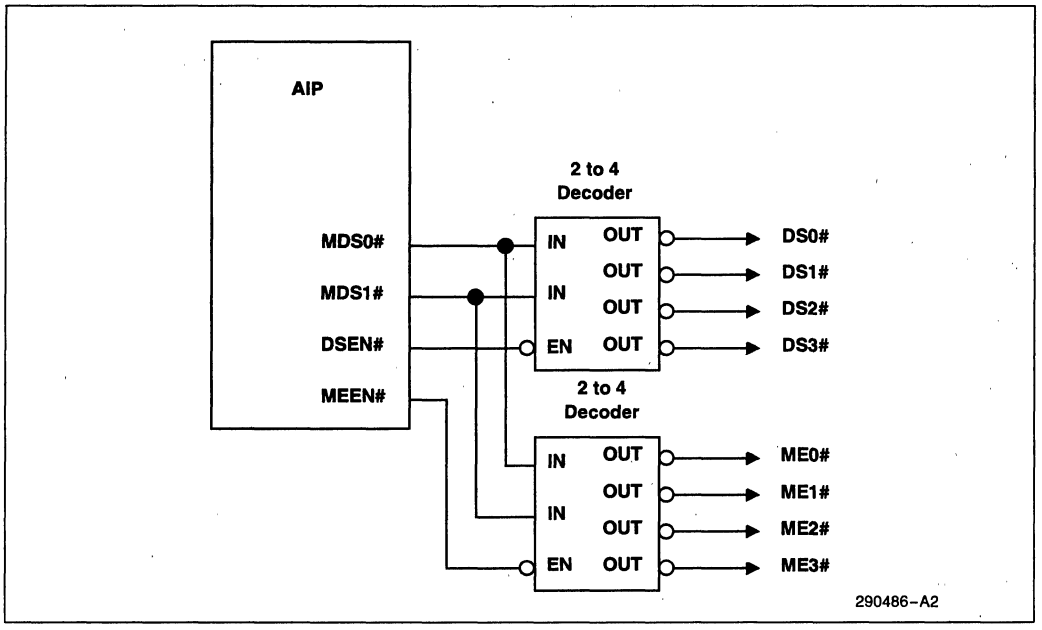


Figure 99. Example External Decoder (Four Drive System)

A.3 TDR—Enhanced Tape Drive Register

I/O Address: Base +3h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

This register allows the user to assign tape support to a particular drive during initialization. Any future references to that drive number automatically invokes tape support. A hardware reset sets all bits in this register to 0 making drive 0 not available for tape support. A software reset via bit 2 of the DOR does not affect this register. Drive 0 is reserved for the floppy boot drive. Bits[7:2] are only available when EREG EN=1; otherwise the bits are tri-stated.

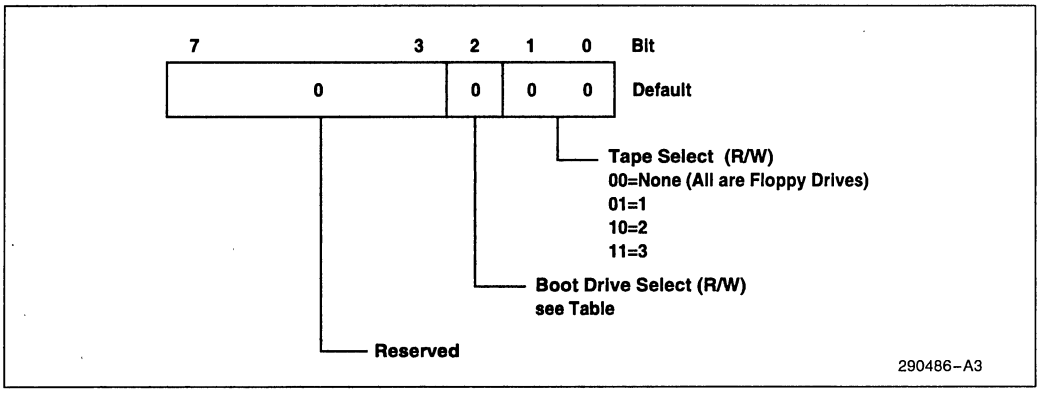


Figure 100. Enhanced Tape Drive Register

Bit	Description
7:3	**Reserved:**
2	**Boot Drive Select (BOOTSEL):** The BOOTSEL bit is used to remap the drive selects and motor enables. The functionality is shown below: **BOOTSEL** **Mapping** 0 DS0 → FDS0, ME0 → FDME0 (default) DS1 → DS1, ME1 → FDME1 1 DS0 → DS1, ME0 → FDME1 DS1 → FDS0, ME1 → FDME0 Only drive 0 or drive 1 can be selected as the boot drive.
1:0	**Tape Select (TAPESEL[1:0]):** These two bits are used by software to assign a logical drive number to be a tape drive. Other than adjusting precompensation delays for tape support, these two bits do not affect the FDC hardware. They can be written and read by software as an indication of the tape drive assignment. Drive 0 is not available as a tape drive and is reserved as the floppy disk boot drive. The tape drive assignments are as follows: **Bits[1:0]** **Drive Selected** 0 0 None (all are floppy disk drives) 0 1 1 1 0 2 1 1 3

5

A.4 MSR—Main Status Register

I/O Address: Base + 4h
Default Value: 00h
Attribute: Read Only
Size: 8 bits

This read only register provides FDC status information. This information is used by software to control the flow of data to and from the FIFO (accessed via the FDCFIFO Register). The MSR indicates when the FDC is ready to send or receive data through the FIFO. During non-DMA transfers, this register should be read before each byte is transferred to or from the FIFO.

After a hard or soft reset or recovery from a powerdown state, the MSR is available to be read by the host. The register value is 00h until the oscillator circuit has stabilized and the internal registers have been initialized. When the FDC is ready to receive a new command, MSR[7:0] = 80h. The worst case time allowed for the MSR to report 80h (i.e., RQM is set to 1) is 2.5 μs after a hard or soft reset.

Main Status Register is used for controlling command input and result output for all commands. Some example values of the MSR are:

- MSR = 80H; The controller is ready to receive a command.
- MSR = 90H; Executing a command or waiting for the host to read status bytes (assume DMA mode).
- MSR = D0H; Waiting for the host to write status bytes.

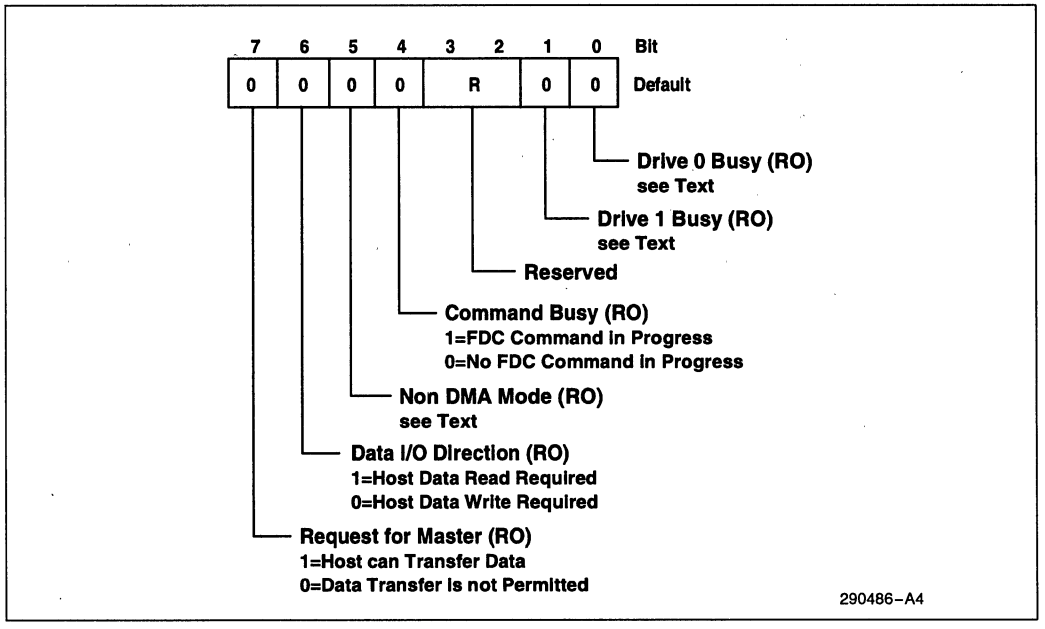


Figure 101. Main Status Register

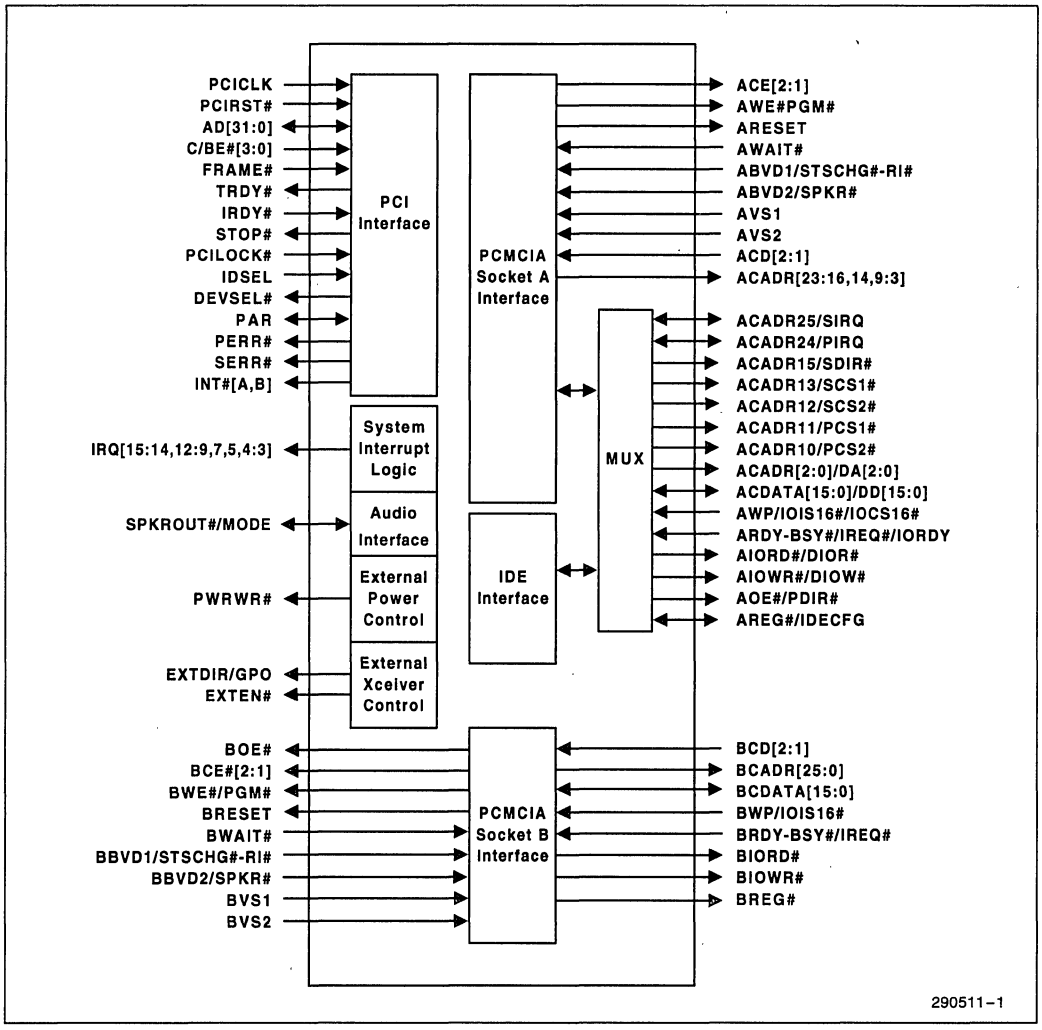
Bit	Description
7	**Request For Master (RQM):** When RQM = 1, the FDC is ready to send/receive data through the FIFO (FDCFIFO Register). The FDC sets this bit to 0 after a byte transfer and then sets the bit to 1 when it is ready for the next byte. During non-DMA execution phase, RQM indicates the status of IRQ6.
6	**Direction I/O (DIO):** When RQM = 1, DIO indicates the direction of a data transfer. When DIO = 1, the FDC is requesting a read of the FDCFIFO. When DIO = 0, the FDC is requesting a write to the FDCFIFO.
5	**NON-DMA (NONDMA):** Non-DMA mode is selected via the SPECIFY Command. In this mode, the FDC sets this bit to a 1 during the execution phase of a command. This bit is for polled data transfers and helps differentiate between the data transfer phase and the reading of result bytes.
4	**Command Busy (CMDBUSY):** CMDBUSY indicates when a command is in progress. When the first byte of the command phase is written, the FDC sets this bit to 1. CMDBUSY is set to 0 after the last byte of the result phase is read. If there is no result phase (e.g., SEEK or RECALIBRATE Commands), CMDBUSY is set to 0 after the last command byte is written.
3	**Drive 3 Busy (DRV1BUSY):** The FDC module sets this bit to 1 after the last byte of the command phase of a SEEK or RECALIBRATE Command is issued for drive 3. This bit is set to 0 after the host reads the first byte in the result phase of the SENSE INTERRUPT Command for this drive.
2	**Drive 2 Busy (DRV1BUSY):** The FDC module sets this bit to 1 after the last byte of the command phase of a SEEK or RECALIBRATE Command is issued for drive 2. This bit is set to 0 after the host reads the first byte in the result phase of the SENSE INTERRUPT Command for this drive.
1	**Drive 1 Busy (DRV1BUSY):** The FDC module sets this bit to 1 after the last byte of the command phase of a SEEK or RECALIBRATE Command is issued for drive 1. This bit is set to 0 after the host reads the first byte in the result phase of the SENSE INTERRUPT Command for this drive.
0	**Drive 0 Busy (DRV0BUSY):** The FDC module sets this bit to 1 after the last byte of the command phase of a SEEK or RECALIBRATE Command is issued for drive 0. This bit is set to 0 after the host reads the first byte in the result phase of the SENSE INTERRUPT Command for this drive.

82092AA
PCI TO PCMCIA/ENHANCED-IDE CONTROLLER

- **Provides the Ultimate Plug and Play Solution for High Performance PCI Desktop Systems**
  - **Supports Combinations of PCMCIA and Enhanced Local Bus IDE Interfaces**
  - **Contains a 32-bit PCI Local Bus Slave Interface Running at 25/33 MHz**
  - **Supports Motherboard and Add-In Card Implementations**
- **Compliant with PCMCIA 2.1/JEIDA 4.1 Interface Standard**
  - **Supports up to Four 68-pin Standard PC Card Sockets, Cascadable for Additional Sockets**
  - **Each Socket Interchangeably Supports Either Memory or I/O PC Cards**
  - **Software Compatible with the Industry Standard 82365SL PCIC**
  - **Supports PCMCIA-ATA Disk Drive Devices**
  - **Features Prefetch Read and Post Write Data Buffer for PCMCIA Memory Cycles**
- **System Bus Timings Compatible with Pentium™ Processors and Intel486™ Processors**
  - **Supports All Intel PCIsets**
  - **Programmable PCMCIA and IDE Interface Timing**
  - **Easily Configured to Support Other Standard Architectures**
- **Dual Voltage Operation**
  - **Each PCMCIA Socket Automatically Configures to Support Either 3.3V or 5.0V PC Cards**
- **Power Management**
  - **Individual Socket Power Control**
  - **Hot Insertion and Removal Capability**
- **PCI Local Bus Interfaces for Four Enhanced IDE Devices**
  - **Provides Write Posting and Read Prefetching to Support High Performance IDE Devices**
  - **Primary and Secondary IDE Devices can be Independently Programmed for Various Speed Selections**
- **Flexible System Configuration Options**
  - **Two-Socket Configuration with On-Chip Buffering**
  - **Four-Socket Configuration with Partial External Buffering**
- **Eliminates the Need for System Configuration Jumpers**
  - **Address Mapping for PCMCIA 2.1/JEIDA 4.1 PC Card Memory**
  - **Address Windowing for I/O Space**
  - **Full 4-GByte PCI Address Range**
  - **Selectable Interrupt Steering from PC Cards to System Interrupt Lines**
- **208-Pin QFP Package**

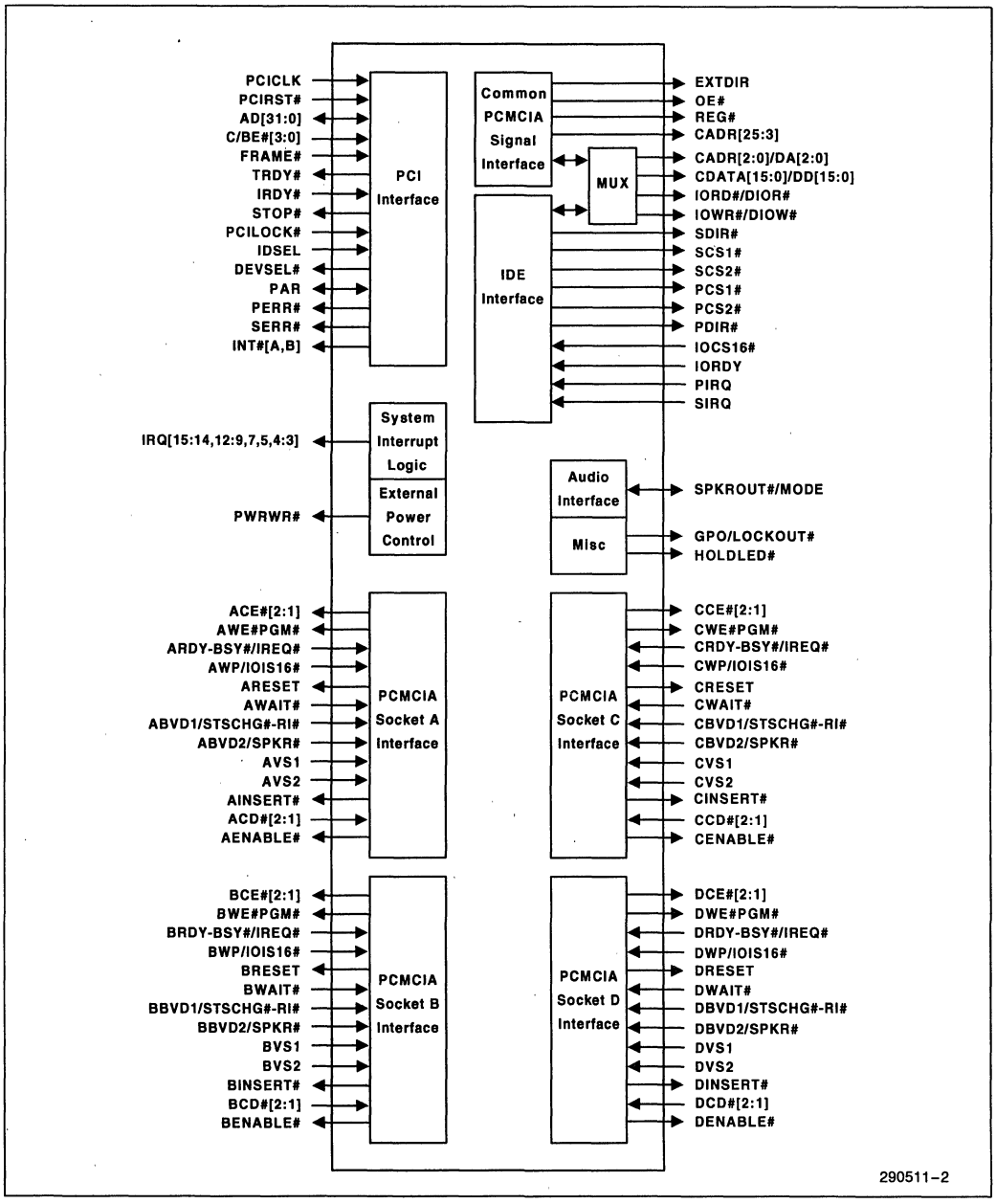
The 82092AA is a high-bandwidth, software-configurable bridge that interfaces as many as four PCMCIA/ExCA (PC Memory Card International Association/Exchangeable Card Architecture) PC cards and four enhanced IDE devices to the Peripheral Component Interconnect (PCI) Bus. It is software compatible with the Intel 82365SL PC Card Interface Controller, but features a 32-bit PCI interface for maximum system performance. The PPEC simplifies system design by reducing component count between the PCI Bus, PC cards, and IDE devices, and maximizes system flexibility by providing such benefits as PC card select decoding, multiple memory address translation maps, power management, and I/O interrupt steering. The PPEC also supports auto-configuration, allowing dynamic system setup when inserting and removing PC Cards.

October 1995
Order Number: 290511-002



Mode 0 (Two Socket) Block Diagram

5



Mode 1 (Four Socket) Block Diagram

82092AA
PCI TO PCMCIA/ENHANCED-IDE CONTROLLER

CONTENTS

PAGE

CONTENTS

CONTENTS

PAGE

CONTENTS

PAGE

1.0 INTRODUCTION

The PPEC is a follow-on product to the Intel industry standard 82365SL PC Card Interface Controller (PCIC) used in mobile systems. It enhances the 82365SL definition by providing a full 32-bit PCI interface for increased system performance, and by supporting up to four PCMCIA sockets and four local bus IDE devices for greater flexibility. The PPEC is 82365SL compatible, and provides a standard system interface for PC Cards at the hardware and data interchange level.

Figure 1 shows a typical PPEC system implementation. The PPEC interfaces directly to the 5.0V PCI Local Bus, and supports four PCMCIA cards and four IDE devices in the configuration that is shown. The PPEC operates as a slave over the full PCI frequency range, but is optimized for 25 MHz and 33 MHz.

PPEC Configuration Modes

The PPEC supports two configuration modes that allow selection of the number of PC Card sockets supported, the number of IDE interfaces supported, and the type of buffering for each socket.

Mode 0 configures the PPEC for two independent PCMCIA sockets, or for one PCMCIA socket and two IDE interfaces. The sockets are internally buffered, and allow hot (power-on) card insertion and extraction. Each of the two IDE interfaces supports two IDE devices.

Mode 1 configures the PPEC for up to four PCMCIA sockets and two IDE interfaces. External buffering is used for the address, data, and shared control signals if fully-buffered interfaces are required. Each of the two IDE interfaces supports two IDE devices, as in Mode 0.

Each PPEC PCMCIA/JEIDA card interface consists of 60 signal and 8 power connections. In Mode 0, each of the two PCMCIA sockets has its own set of signals and buses, but the IDE interface signals are multiplexed with Socket A signals. In Mode 1, one address bus, one data bus, and several control signals are used as common signals for all four sockets, and several IDE interface signals are multiplexed with the common signals.

1.1 Enhanced PCI Local Bus IDE Interface

The local bus IDE interfaces, designated the Primary IDE Interface and the Secondary IDE Interface, and their corresponding drives #0 and #1 are independently programmed to operate in the enhanced (programmable) timing mode, or in standard compatible timing mode. The IDE physical interface is multiplexed with existing PCMCIA signals to reduce cost, and the IDE controller uses internal PCMCIA Post-Write data buffers and separate Read-Prefetch buffers to improve the system performance. The interfaces are externally buffered in both PPEC configuration modes.

The PPEC provides an IDE Hardware Configuration mechanism using the Power-On IDE Configuration Register which is mapped in the PCI configuration space. This allows full use of the PPEC's Enhanced Fast Local Bus IDE without any requirement for BIOS upgrade or modification if the BIOS is not aware of the PPEC IDE.

The higher performance of the PCI local bus IDE architecture with respect to the ISA IDE architecture results from the faster timing modes that are available in the PCI local bus IDE architecture. The PPEC provides improved timing even when the IDE controller is configured to run cycles that correspond to IDE ATA specification timing modes 0, 1 or 2 (which are originally defined for ISA-based systems) because of the proximity to the host CPU, and because of the clock granularity (PCI 33 MHz) at which timing is controlled. A more significant improvement is obtained when the PPEC is configured for the drives that support enhanced timing mode 3 or any arbitrarily-defined faster timing mode.

Enhanced Timing

The PPEC features programmable timing (see Primary and Secondary IDE Timing Control Register descriptions) in the form of clock counts for the following timing parameters:

- T_{su} (address/data set-up time)
- T_{pw} (command pulse width)
- T_{cyc} (overall cycle length)

5

The programmable timing allows support of currently defined and future IDE Modes, and for optimizations based on PCI clock frequencies other than 33 MHz. IDE Primary and Secondary interface timing is independently controlled, allowing IDE drives with different timing characteristics to be supported by the same physical interface while still operating at their optimum speeds.

Further timing configuration is provided at the level of the Primary and Secondary interface. Each drive (drive 0 and drive 1) can be independently programmed to run in enhanced timing mode or in standard compatible Mode #0, so that any combination of fast and slow drives is possible.

1.2 Internal Register and PCMCIA Address Window Access

The PCI-PCMCIA Bridge PCI Configuration Registers and the PCI-IDE PCI Configuration Registers conform to the Peripheral Component Interconnect (PCI) specification. The specification describes the essential registers that must be supported by PCI devices and functions, and should be referenced for a detailed explanation of the PCI-PCMCIA Bridge and PCI-IDE PCI Configuration Register access.

1.2.1 PCMCIA SOCKET CONFIGURATION REGISTER ACCESS

The PCMCIA Socket Configuration registers, comprised of general setup registers, interrupt registers, I/O mapping control registers, and memory mapping control registers, are used for control of the PCMCIA socket functions. They are a superset of the 82365SL register set, and are accessed using the same indexing method that is used in the 82365SL.

Two read/write ports, an *index port* and a *data port,* are used to access the registers. The index port is written with the register offset that is used to access the register. Data is then written to the register or read from the register via the data port.

The index port address is loaded into the PCI-PCMCIA Bridge Base Address Register, which is located in PCI Configuration space. The data port address is the next location (index port address +1). When writing to the configuration registers, the index and data registers may be accessed simultaneously with a 16-bit write operation to increase performance. The index value is placed on data bits[7:0], and the data value to be written is placed on data bits[15:8].

The PPEC does not respond to a data port read or write operation unless a valid index has first been written to the index port.

1.2.2 PCMCIA MEMORY WINDOWS

The PPEC supports five independently enabled and configured memory and I/O address mapping windows for each PCMCIA PC Card socket. The windows allow portions of 64 MByte common memory and/or 64 MByte attribute memory spaces on the PC Cards to be mapped to portions of the PCI Memory address. Each window's data bus width, PCMCIA interface timing, software write protect, and enable can be independently controlled. Mapping of each memory window can start and stop on any 4 KByte boundary of PCI memory space within a 16 MByte page.

A Card Memory Page Address Register associated with each socket allows selection of the 16 MByte window page anywhere in the 4 GByte PCI address space (see Figure 2). The PC Card memory offset address is added to the PCI address bits 23:12 to generate the address for the PC Card. The address mapping is compatible with the 82365SL.

A window is opened by writing the PCI Memory start address, PCI Memory stop address, and PC Card memory offset to the window's *System Memory Address Mapping Start, System Memory Address Mapping Stop, and Card Memory Offset Address* high and low byte registers. The Card Memory Page Address Register must be written with the window's 16 MByte page address in PCI memory space, and the window must be enabled in the Address Window Enable Register.

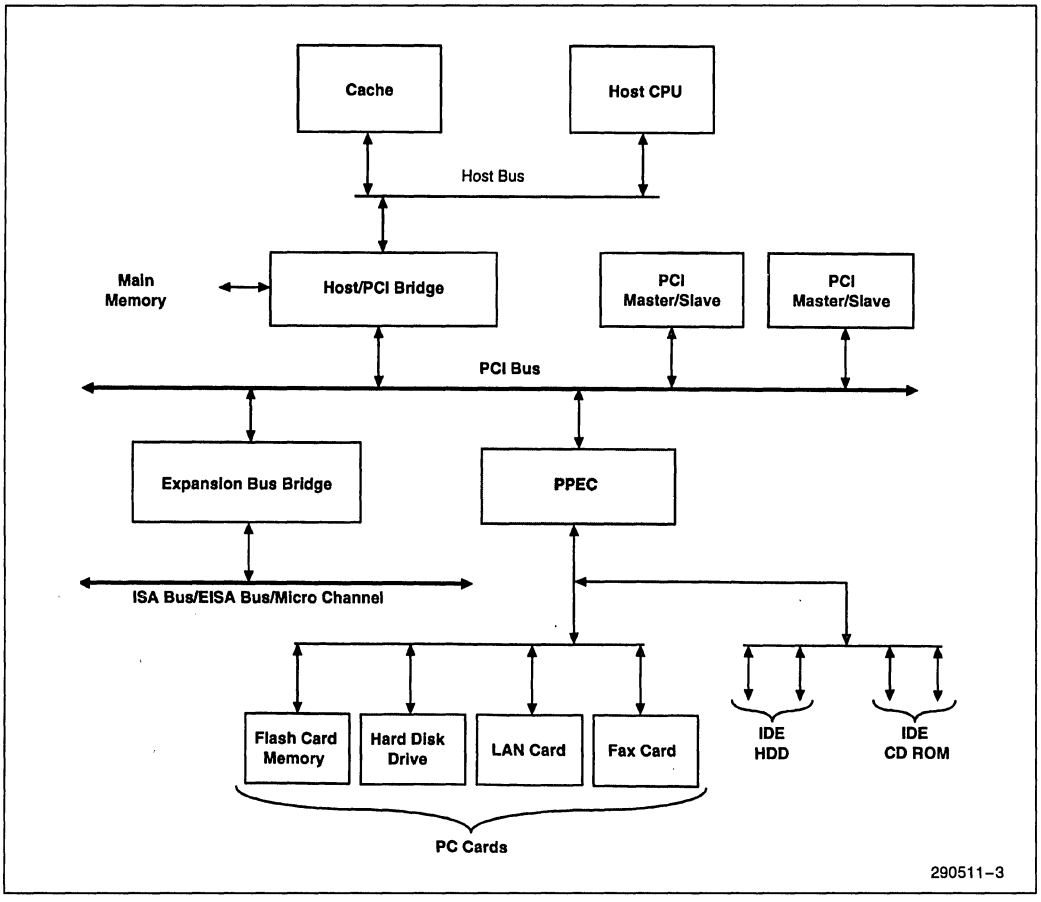


Figure 1. Typical PPEC System Implementation

5

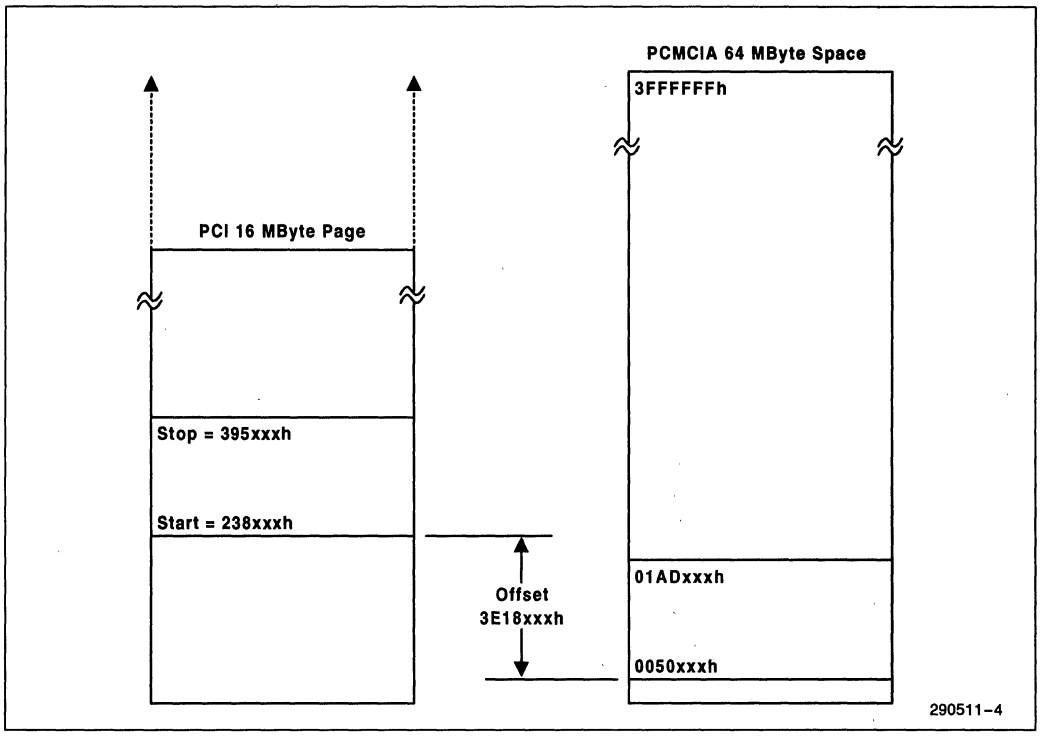


Figure 2. PCMCIA Memory Address Mapping

Common/Attribute Memory Address Mapping

Both Common and Attribute memory can be accessed on the PC Card through any of the PCI Memory address mapping windows according to the state of the *Register Active* bit in the Card Memory Offset Address High Byte Register. When this bit is set to 0, common memory can be accessed; when set to one, attribute memory can be accessed. The PCI memory window to common or attribute memory can be mapped from any PCI address to any PC Card address.

Several PCI Memory address mapping windows to different common memory address spaces can be opened simultaneously. Each of these windows can be configured to use a different timing mode, software write protect, and data width.

1.2.3 PCMCIA I/O WINDOWS

The PPEC features two independently enabled and controlled I/O address windows for each PCMCIA PC Card socket. The windows can be non-contiguous, and each window's I/O data bus width can be independently controlled. The windows have a 1 byte addressing resolution.

I/O addressing of PC Cards is very similar to memory addressing. Each I/O address window has a 16-bit *start address* and a 16-bit *stop address* located in the window's *I/O Address Start* and *I/O Address Stop* high and low byte registers. PCI I/O Address bits[15:0] are compared with the start and stop addresses and must be greater than or equal to the start address, and less than or equal to the stop address for access to the window. PCI address bits[31:16] must be 0 when addressing I/O Cards.

Indirect offset addressing is not supported for I/O windows. PCI I/O Address bits[15:0] are passed directly to the PC Card address pins if they fall within an I/O Window. Bits[25:16] of the PC Card address are driven low.

1.3 Data Buffers

The PPEC features read prefetching and write posting data buffering for up to four Dwords. This allows high speed 32-bit data transfers between the PCI Local Bus and the PPEC, and lower-speed 8-bit and 16-bit data transfers between the PPEC and the PC Cards and the IDE devices. Assembly/disassembly logic translates 32-bit data from the PCI bus into 8- and 16-bit data required by the PCMCIA PC Cards and the IDE devices.

Figure 3 shows a representation of the data buffer and the IDE prefetch buffers. The same physical buffers are used for both PCMCIA/IDE Posted Write operations, and for PCMCIA Prefetch operations. Separate 32-bit latches are used for storing IDE Prefetch data: one for the Primary drive interface, and one for the Secondary drive interface.

The operation of the data buffers is controlled by two bits in the PCI-PCMCIA Configuration Control Register that enable and disable Posted Write Buffer and IDE Prefetch operation. Both modes may be active simultaneously, but the write posting function has priority over the prefetch function.

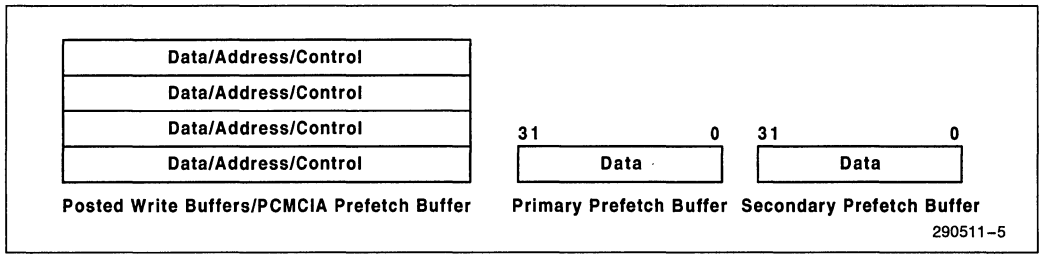


Figure 3. Data and IDE-Prefetch Buffers

Posted Writes and Prefetch Reads for IDE cycles are globally enabled in the PCI-PCMCIA PCI Configuration Control Register, and individually enabled for each IDE device in the PCI-IDE Configuration Control Register. IDE Posted Write cycles operate the same way as PCMCIA Posted Write cycles. However, the PPEC does not support burst data transfers for IDE Posted Write operations.

The data buffers are not used for Attribute Memory cycles because of potential timing dependencies.

1.4 PCMCIA Interface

The PPEC configuration mode is selected by an external 10-20K Ohm resistor on the SPKROUT#/MODE pin. The voltage level applied to the pin is sampled at the end of the reset sequence, and the state is stored internally in the PPEC-PCMCIA PCI Configuration Control Register (PCICON). The pin is then reconfigured as an output to support the SPKROUT function. The AREG#/IDECFG pin is also sampled at the end of the reset sequence to determine whether IDE is supported in Mode 0.

Table 1 shows mode selection with the SPKROUT#/MODE and AREG#/IDECFG pins.

Mode 0

Two PCMCIA sockets are supported in Mode 0. Each socket interface (Socket A and Socket B) has its own data bus, address bus, and control pins.

If the AREG#/IDECFG pin is sampled high at the end of the reset sequence, the Socket A PCMCIA registers are disabled and the Socket A interface pins are reconfigured for IDE support, allowing the implementation of two Fast Local Bus IDE interfaces capable of supporting up to four IDE devices. Signal mapping for the IDE Interface in Mode 0 is described in Section 2.8, IDE Interface Signals.

Mode 1

Four PCMCIA sockets are supported in Mode 1 by allowing sockets A, B, C, and D to share a common data bus (CDATA[15:0]), a common address bus (CADR[25:0]), and four common control signals (REG#, OE#, IORD#, and IOWR#). The other control signals are not shared.

External data buffers for each PCMCIA Socket may be used in Mode 1 to prevent the PC Cards from driving an excessive load, and to allow both 3.3V and 5V cards to be used simultaneously in a system. External buffers for the shared address and control lines are optional. They can be fully buffered for each socket or, by utilizing the INSERT# pins, can be either driven directly by the PPEC, or buffered by a single buffer to increase drive strength. See the *PPEC Design Guide* for specific application information.

Table 1. Configuration Mode Selection

SPKROUT#/MODE	AREG#/IDECFG	Mode	Configuration
0	0	0	2 Fully-Buffered PCMCIA Sockets; No IDE
0	1	0	PCMCIA Socket B Only; Dedicated IDE Interface
1	X	1	4-PCMCIA Sockets; Shared Address, Data, Control

Two Fast Local Bus IDE interfaces can be enabled in Mode 1 by programming the PPEC-IDE Interface PCI Configuration Registers. When an IDE cycle is initiated, the PDIR# and SDIR# pins provide direction control for the external IDE data buffers, and the PCS1#, PCS2#, SCS1# and SCS2# signals enable the IDE devices. The PCMCIA sockets are not affected during IDE cycles because the card enables for each socket remain inactive.

The AREG#/IDECFG pin is ignored in this mode.

1.4.1 CARD INSERTION AND EXTRACTION

The INSERT# function allows the PPEC to prepare for the insertion of a PC Card, preventing the insertion from affecting other sockets in implementations requiring shared signal lines (Mode 1, partially buffered). The INSERT# pin is driven active at least 1.5 μs prior to the PC Card signal-pin contact with a PCMCIA socket by an external circuit described in the PPEC Design Guide.

The PPEC empties the Posted Write Buffers (PWBs) and tri-states the shared PCMCIA address and control lines when a falling edge occurs on INSERT#. The PPEC drives the shared signals after power has been applied to the new PC Card, and normal operation resumes.

When a rising edge occurs on a CDx pin, the PPEC again empties the Posted Write Buffers and tri-states the shared address and control lines, and the dedicated PCMCIA socket signals. A rising edge on INSERT# indicates that the PC Card has been disconnected from the signal pins, allowing the PPEC to drive the shared signals and resume normal PCMCIA operation.

Hot insertion and extraction is possible in Mode 0 because all PCMCIA interface signals are independent of each other in this mode. When a card insertion is detected via the Card Detect inputs (xCD#[2:1]) with the Card Detect interrupt enabled, a Card Status Change interrupt is initiated, and power is applied to the PCMCIA socket under software control. When a card removal is detected via the Card Detect inputs, the voltage to the socket is turned off under software control.

1.4.2 POWER CONTROL

The PPEC implements power management for each PCMCIA socket individually. Socket power management is controlled through the Power Control Register and the VS1/VS2 pins.

1.4.2.1 Power Control Register Operation

The Power Control Register controls the routing of power to the PCMCIA socket and enabling of the PCMCIA socket interface pins. A PCIRST clears all of the bits in this register. A detailed description of all of the bits in this register is found in Section 3.2.1.3. This section describes only the V_{CC} Control and V_{PP} Control bits.

V_{CC} Control

The V_{CC} Control bits in the Power Control Register control power to the PC Cards via an external latch. The control signals $V_{CC}5V$ and $V_{CC}3V$ are routed to the external latch through the ACDATA lines during Power Control Write cycles. The value of these bits is modified by software writes to the Power Control Register or by hardware according to the value of the VS1/VS2 pins. Hardware modifies the V_{CC} Control bits only when the INSERT# signals are asserted, indicating partially buffered, Mode 1 implementations.

The V_{CC} Control bits control V_{CC} routing using the following encoding:

Power Control Register Bits		$V_{CC}5V$	$V_{CC}3V$	Description
Bit 4	Bit 3			
0	0	0	0	No Connect
0	1	0	0	Reserved
1	0	1	0	5.0 V
1	1	0	1	3.3 V

If the INSERT# signals are used in Mode 1 partially-buffered implementations, the VS1/VS2 pins are sampled during a PC Card insertion and are used to determine the values of $V_{CC}5V$ and $V_{CC}3V$. The VS1/VS2 pins affect the V_{CC} Control bits in the following way:

VS1	VS2	Power Control Register Bits		$V_{CC}5V$	$V_{CC}3V$	Description
		Bit 4	Bit 3			
0	0	1	1	0	1	3.3V
0	1	1	1	0	1	3.3V
1	0	0	0	0	0	Not Supported; (SERR# Asserted)
1	1	1	0	1	0	5.0V

In Mode 0 and in fully-buffered Mode 1 implementations, the INSERT# signals are not used, and power control is completely dependent upon software.

V_{PP} Control

The V_{PP} Control bits in the Power Control Register are latched externally, along with the V_{CC} Control bits. They are modified with software writes to the Power Control Register only.

1.4.2.2 External Power Control Latch

The PPEC is designed to directly interface with Maxim's MAX780 Dual-Slot PCMCIA Power Controller. This device provides V_{PP} power and controls V_{CC} power for 2 PCMCIA sockets. Two MAX780's are required for 4-socket implementations. The MAX780 contains an internal register for latching the power control signals provided by the 82365SL and compatible controllers $V_{PP}EN[1:0]$ and $V_{CC}EN[1:0]$. This latch allows the PPEC to write the values of the power control signals via the PCMCIA data lines, thus eliminating the need for dedicated power control signals. If the Maxim device is not used in a system, an external register (74ALS273 or equivalent) is required to latch the power control signals. This external latch, whether implemented using the Maxim device or a 74ALS273, is referred to in this document as the External Power Control Latch.

The following are the signals that transfer power control information to the External Power Control Latch, and the bus signal pins through which they are transferred.

Power Signal	Transfer Pin Name
Power Control Write Signal	PWRWR#
Socket A V_{PP}EN0 Control Signal	ACDATA[0]
Socket A V_{PP}EN1 Control Signal	ACDATA[1]
Socket B V_{PP}EN0 Control Signal	ACDATA[2]
Socket B V_{PP}EN1 Control Signal	ACDATA[3]
Socket A V_{CC}3V Control Signal	ACDATA[4]
Socket A V_{CC}5V Control Signal	ACDATA[5]
Socket B V_{CC}3V Control Signal	ACDATA[6]
Socket B V_{CC}5V Control Signal	ACDATA[7]
Socket C V_{PP}EN0 Control Signal (Mode 1 Only)	ACDATA[8]
Socket C V_{PP}EN1 Control Signal (Mode 1 Only)	ACDATA[9]
Socket D V_{PP}EN0 Control Signal (Mode 1 Only)	ACDATA[10]
Socket D V_{PP}EN1 Control Signal (Mode 1 Only)	ACDATA[11]
Socket C V_{CC}3V Control Signal (Mode 1 Only)	ACDATA[12]
Socket C V_{CC}5V Control Signal (Mode 1 Only)	ACDATA[13]
Socket D V_{CC}3V Control Signal (Mode 1 Only)	ACDATA[14]
Socket D V_{CC}5V Control Signal (Mode 1 Only)	ACDATA[15]

Note that in Mode 1, the ACDATA bus is renamed to CDATA.

The value of the V_{CC}3V and V_{CC}5V signals and V_{PP}EN[1:0] are determined by the control bits in the Power Control register for each socket.

The External Power Control Latch is updated by placing the values of the power control signals onto the corresponding ACDATA lines, and pulsing the PWRWR# signal low. This operation takes place after each of the following conditions:

- A write cycle to any of the Power Control registers
- A high Card Detect pin on a socket having Auto Power enabled
- Rising edge of PCIRST#
- PC Card Reset
- Detection of a PC Card insertion via one of the xINSERT# pins

The PPEC allows any cycle currently taking place on the PCMCIA bus or PCI bus to complete before performing an external power control write cycle, then initiates a write cycle to the external latch. The power control write operation cycle time is 6 PCICLKs, and consists of placing the Power Control bits for all sockets on the ACDATA bus, pulsing the PWRWR# signal for 5 PCICLKs, and holding the data for one additional clock to guarantee hold time. If a PCMCIA-targeted PCI cycle is initiated during an external power control write cycle, the PCI cycle is held in wait-states until the PCMCIA cycle can be executed.

5

1.4.2.3 Hardware-Initiated Power On Sequence

When the INSERT# pins are used to detect card insertions, the PCMCIA bus is tri-stated until power has been applied to all PC Cards. A condition could exist where a PCI Master device other than the CPU tries to access the PCMCIA bus while the PCMCIA bus is tri-stated, causing the PCI Master to be retried. This would effectively lock out the CPU from the PPEC and prevent the CPU from applying power to the PC Card. To avoid this situation, the PPEC must perform a hardware-initiated power-on sequence to the PC Card whenever the PCMCIA bus is tri-stated due to a PC Card insertion.

The following sequence of events describes a hardware-initiated power-on sequence:

1. INSERT# is detected active for one of the PCMCIA sockets.
2. Line buffers are flushed and the PCMCIA bus is tri-stated.
3. CD1# and CD2# are detected active.
4. The Power Control Register is updated according to the values of VS1/VS2.
5. An external power control write cycle is performed.
6. The PPEC is held for 256 PCICLKs (7.68 μs minimum) to allow V_{CC} voltage to stabilize.
7. The PCMCIA bus and The PPEC resume normal operation.

Any subsequent writes to the Power Control Registers by software override the V_{CC} Control bits set by hardware. If VS1/VS2 indicate the presence of an X.X-only PC Card, the PPEC does not apply power to the card and the PCMCIA bus is held in a tri-state condition until the card is removed. This situation is indicated by HOLDLED# remaining active. If SERR# is enabled, the PPEC asserts SERR# for one PCICLK to alert the system that an error condition exists, and operation can not continue.

1.4.2.4 Auto Power Enable

The Auto Power function in the PPEC is intended to allow hardware to automatically power down a PCMCIA socket based on the Card Detect pins (CDx). With Auto Power enabled, the power control signals $V_{CC}5V$ and $V_{CC}3V$ are active only while both Card Detect inputs are low. As soon as one of the CDx lines goes high indicating that a card is being extracted, an external power control write cycle is initiated to negate the active power control pins. The V_{PP} control pins are automatically negated with the negation of the V_{CC} control pins, independent of the Auto Power function. Software is responsible for debouncing the CDx pins and disabling Auto Power whenever a card is extracted. When the Auto Power function is disabled, the power control signals are not qualified with the Card Detects. The sequence of steps for inserting and extracting a PC Card when using Auto Power is as follows:

1. The default is Auto Power Enable = 0.
2. Software receives a Card Status Change (CSC) interrupt as a result of a card being inserted.
3. Software waits for the CDx pins to become stable.
4. Software reads the VSx pins and sets the V_{CC}/V_{PP} control bits in the Power Control Register.
5. The PPEC issues a write cycle to the external power control latch (Maxim Power Switch or discrete latch).
6. Software waits for power to become stable (50 μs minimum).
7. Software enables the socket interface and sets Auto Power Enable = 1.

Normal operation takes place.

8. The PPEC detects a rising edge on either CDx pin.
9. The PPEC sends a CSC interrupt and issues a write cycle to the external power control latch to disable the socket power. A hardware flag is automatically set, disabling power to the socket.
10. Software waits for the CDx pins to become stable.
11. Software disables the socket interface and sets Auto Power Enable = 0.
12. The PPEC clears the flag that disables power to the socket as a result of step 11.

When the Auto Power bit of the Power Control register (bit 5) is 0, Auto Power is disabled and power is controlled directly from the power control bits without being qualified with CDx. The PPEC does ***not*** prevent software from powering a card to a voltage other than that indicated by the VSx pins. Table 2 summarizes the operation of the PPEC power control.

1.5 PC Card ATA Support

The PCMCIA specification defines a protocol for ATA PC Cards. No special requirements are needed for accesses to PC Card ATA. Accesses to all ATA registers are treated as normal I/O accesses.

1.6 PCI Interface

The PPEC is a PCI target-only device. Its PCI Interface conforms to the Peripheral Component Interconnect (PCI) specification, which should be referenced for an understanding of the interface.
Table 3 identifies the PCI commands that the PPEC supports, and their encoding on signal lines C/BE#[3:0]. The PCI bus signal descriptions in the following section further define the PCI operations that are supported by the PPEC.

1.6.1 PCI SUPPORT

The following sections describe PPEC PCI support. Note that the PPEC is a target-only device, and therefore does not support PCI functions that are defined for PCI masters.

1.6.1.1 Address Decoding

The PPEC uses only positive address decode. The PPEC's PCI-PCMCIA Bridge and PCI-IDE Interface functions both have SLOW DEVSEL# timing response.

Table 2. Power Control Operation

Power Control Register			PPEC Pins		Tri-state Outputs (See Note)	Interface Status Register
Output Enable (Bit 7)	V_{CC} Enable (Bit 4)	Auto Power Enable (Bit 5)	CD1#	CD2#		PC Card Power Active
X	0	X	X	X	OFF	0
0	1	0	0	0	OFF	1
1	1	0	0	0	ON	1
X	1	0	X	1	OFF	1
X	1	0	1	X	OFF	1
0	1	1	0	0	OFF	1
1	1	1	0	0	ON	1
X	1	1	X	1	OFF	0
X	1	1	1	X	OFF	0

NOTE:
For this table, the term Tri-state Outputs includes the PPEC outputs that are unique for a given PCMCIA socket. This includes all of the address, data, and control signals in Mode 0 (2-socket), but does not include the shared address, data, and control signals in Mode 1 (4-socket). The shared PCMCIA signals defined in Mode 1 are enabled and disabled as a function of the INSERT# signals as described in Section 1.4.1.

1.6.1.2 Configuration Cycles

The PPEC supports only Type 0 PCI configuration cycles. As a multifunctional device it supports access to functions numbered 0 and 1. It does not respond to a configuration cycle that accesses functions 2-7, even if the functions are selected with the IDSEL mechanism.

NOTE:
None of the PPEC internal registers or PCMCIA I/O or memory locations can be accessed after PCI reset until PCI Configuration Software (part of BIOS) configures the system resources properly.

1.6.1.3 Burst Transfer Support

The PPEC supports burst transfers to PCMCIA memory and to the IDE I/O Data Port with a post-write buffering mechanism when internal data buffering is enabled.

The PPEC supports only Linear Incrementing burst transfers. Attempts to access the PPEC using burst transfers in a mode other than Linear Incrementing results in subsequent target disconnects, and splitting of the burst cycle into multiple single data phase transfers.

1.6.1.4 Exclusive (Locked) Access Support

The PPEC can be locked as a resource by any PCI Initiator. In the context of locked cycles, the PPEC and the PCMCIA subsystem are considered a single resource. A locked access to any address within the PCMCIA subsystem locks the PPEC.

Note that write-posting and read-prefetch are disabled for PCI locked cycles. Any PCI Initiator access to the PPEC subsystem while it is locked results in retry.

Table 3. PCI Commands

C/BE# [3:0]	Command Type	Supported As Target
0000	Interrupt Acknowledge	No
0001	Special Cycle	No
0010	I/O Read	Yes
0011	I/O Write	Yes
0100	Reserved	N/A(3)
0101	Reserved	N/A(3)
0110	Memory Read	Yes
0111	Memory Write	Yes
1000	Reserved	N/A(3)
1001	Reserved	N/A(3)
1010	Configuration Read	Yes
1011	Configuration Write	Yes
1100	Memory Read Multiple	No(2)
1101	Reserved	N/A(3)
1110	Memory Read Line	No(2)
1111	Memory Write and Invalidate	No(1)

NOTES:
1. Treated as Memory Write.
2. Treated as Memory Read.
3. PPEC does not respond on these commands.

1.6.1.5 Transaction Termination

As a target, the PPEC terminates transactions for the following conditions.

Disconnect

The PPEC responds with a disconnect when it is the target of multiple data phase transactions that cannot be serviced by the internal buffers (i.e., posted for writes or supply prefetched data during read operations). During posting, the PPEC terminates the cycle using disconnect semantics as soon as all posted write buffers are occupied. Similarly, the PPEC disconnects during burst reads as soon as a miss is generated (prefetch data is not available). This is because the next data phases would exceed the 8 PCI clock incremental latency limit while the PCI is kept in wait-states for more that 8 PCI clocks until one of the post write buffers is emptied to its destination, or additional data is fetched from the PCMCIA card or IDE interface.

Retry

The PPEC retries memory write cycles when all post write buffers are full. It also retries any cycle when it is locked as a resource and a PCI master tries to access the PPEC without negating the PCILOCK# signal during the address phase.

Target Abort

The PPEC generates this type of termination during non-aligned Dword I/O transfers with illegal combinations of address and BEx.

1.6.1.6 Parity Generation And Checking

The PPEC supports parity generation and checking for both the address and data phases of cycles in which it positively decodes address. The PPEC asserts the PERR# signal when it recognizes a parity error during bus transactions in which it is involved. The PPEC asserts the SERR# o/d signal for one PCI clock when it detects an address phase parity error, or a PCMCIA interface system error (i.e., when a X.XV-only PC Card is inserted).

1.6.1.7 PCI Memory Cache Support

The PCI can provide basic cache coherency control with two optional PCI signals, SDONE and SBO#. The PPEC does not support those signals, so PCMCIA memory cannot be directly cached. However, alternative schemes can be used (including "shadowing" in main memory and caching locally) to speed access to the read-only PCMCIA memory. This can improve performance of the XIP (eXecute In Place) PCMCIA applications.

1.7 System Interface Functionality

The PPEC can connect to system interrupts in two different ways:

a. via 10 direct system interrupt signals.

b. via 2 PCI interrupt signal lines which require additional routing.

The direct system interrupt mode is provided so that PCMCIA software that requires "ISA compatibility" (i.e. non-shareable IRQ handlers which require specific IRQ level) can run without modifications on Intel Microprocessor Architecture based platforms with PPEC as a PCI-PCMCIA Bridge. In this case, PCI interrupts are not used but the PCMCIA Card interrupts (as well as the Card Status Change interrupt) are configured to connect directly to specific system IRQ lines. On the system side, the PPEC interrupt signals can support either edge (ISA like) or level (active low-EISA, Microchannel, PCI like) triggering. Multiple PC Cards in a system can conflict if they try to utilize the same Edge interrupt Level. By steering them to different interrupt lines, the conflicts can be eliminated.

Designs that are not dependent on this type of software "compatibility" can use PCI interrupts (i.e., interrupt is configured using mechanism "b").

The Global Control Register provides individual bits to enable Edge/Level Mode. The Interrupt and General Control Register contains bits for I/O card Interrupt Steering. The Card Status Change Interrupt Configuration register contains bits for Card Status Change Interrupts.

1.7.1 DIGITAL AUDIO SUPPORT—SPKROUT# SIGNAL

The PPEC supports PC Card digital audio SPKR# signals. These signals are passed through to the SPKROUT# line, which is an exclusive-OR of all SPKR# inputs (from all sockets). In motherboard designs, this output line can be connected to the system speaker driver directly. In add-in card implementations, a cable can be used to access the speaker, or a Piezo electric transducer can be provided on the add-in card.

2.0 PPEC SIGNALS

The signals described in this section are arranged in functional groups. The # symbol at the end of a signal name indicates that the signal's active or asserted state occurs when the signal is at a low voltage level. When # is not present after the signal name, the signal is asserted when it is at the high voltage level.

The terms *assertion* and *negation* are used extensively in this document to minimize confusion when a mixture of *active-low* and *active-high* signals are described. The terms *assert* and *assertion* indicate that a signal is active, independent of whether that level is represented by a high voltage or a low voltage. The terms *negate* and *negation* indicate that a signal is inactive.

The following notation is used to describe the PPEC signal types:

in:	A standard input-only signal.
in (ST)	A Schmitt Trigger input signal.
out	A totem pole output signal.
o/d	An open drain input/output signal.
t/s	A bi-directional tri-state signal.
t/s/o	A uni-directional, tri-state output signal.
s/t/s	A sustained tri-state signal. This is an active low tri-state signal that is owned and driven by one and only one agent at a time. The agent that drives a s/t/s signal low must drive it high for at least one clock before letting it float. A new agent can not start driving an s/t/s signal sooner than one clock after the previous owner tri-states it. An external pull-up must be provided by the central resource to sustain the inactive state until another agent drives the signal.

2.1 PCI Bus Interface Signals

Name	Type	Description
PCICLK	in	**PCI CLOCK:** Provides timing for all transactions on the PCI bus. All other PCI signals are sampled on the rising edge of PCICLK, and all timing parameters are defined with respect to this edge. PPEC design is optimized for 25 MHz and 33 MHz PCI bus frequency. **NOTE:** PCMCIA and IDE state machines use PCICLK as a clock reference. This allows simpler design with good granularity for optimized timing, but selection of a PCI clock frequency other than 25 or 33 MHz may impact PCMCIA and IDE interface performance.
PCIRST#	in	**PCI RESET:** Forces the entire PPEC component into a known state. All t/s and s/t/s signals are forced to a high impedance state, and the o/d signals are allowed to float high. All internal PPEC state machines are reset, and all registers are set to their default values. PCIRST# may be asynchronous to PCICLK when asserted or negated. Although asynchronous, negation must be with a clean, bounce-free edge. PCIRST# must be asserted for a minimum 1 ms, and PCICLK must be active during the last 100 μs of the PCIRST# pulse.
AD[31:0]	t/s	**ADDRESS AND DATA:** Address and data are multiplexed on the same PCI pins. During the first clock of a transaction, AD[31:0] transfer a physical address (32 bits). During following clocks, AD[31:0] transfer data. A bus transaction consists of an address phase, followed by one or more data phases. PCI supports write bursts. Little-endian byte ordering is used. AD[7:0] define the least significant byte (LSB), and AD[31:24] the most significant byte (MSB). The information contained in the two low order address bits varies by address space. In the I/O address space, AD[1:0] are used to provide full byte address. During memory space accesses, these two bits provide information on the type of burst ordering. During configuration space accesses, they identify the type of configuration access. When the PPEC is the target of a PCI cycle, AD[31:0] is input during the address phase of a transaction. During the following data phase(s), the PPEC asserts data on AD[31:0] if a PCI read, or accepts data if a PCI write.

2.1 PCI Bus Interface Signals (Continued)

Name	Type	Description
C/BE#[3:0]	in	**BUS COMMAND AND BYTE ENABLES:** These signals are multiplexed on the same PCI pins. During the address phase of a transaction, C/BE#[3:0] define the bus command for bus command definitions. During the data phase, C/BE#[3:0] are used as Byte Enables. The Byte Enables determine which byte lanes carry meaningful data. C/BE#[0] applies to byte 0, and C/BE#[3] to byte 3. C/BE[3:0]# are not used for address decoding.
FRAME#	in	**CYCLE FRAME:** Driven by the current initiator to indicate the beginning and duration of an access. FRAME# is asserted to indicate that a bus transaction is beginning. Data transfers continue while FRAME# is asserted. When FRAME# is negated, the transaction is in the final data phase.
TRDY#	s/t/s	**TARGET READY:** Asserted by the PPEC as a target to indicate completion of the current data phase. TRDY# is used in conjunction with IRDY#. A data phase is completed on any clock during which both TRDY# and IRDY# are sampled asserted. When the PPEC is the target during a read cycle, TRDY# indicates that the PPEC has valid data asserted on AD[31:0]. When it is a target during a write cycle, it indicates that the PPEC is prepared to latch data.
IRDY#	in	**INITIATOR READY:** IRDY# as an input indicates that the current cycle initiator is able to complete the current data phase of the transaction. It is used in conjunction with TRDY#. A data phase is completed on any clock during which both IRDY# and TRDY# are sampled asserted. When the PPEC is the target of a write cycle, IRDY# indicates that valid data is present on AD[31:0]. During a read, it indicates that the initiator is prepared to latch data.
STOP#	s/t/s	**STOP:** Indicates that the PPEC, as a target of an PCI cycle, is requesting a master to stop the current transaction. Different semantics of the STOP# signal are defined in the context of other handshake signals (TRDY# and DEVSEL#).
PCILOCK#	in	**PCI LOCK:** Indicates an atomic operation that may require multiple transactions to complete. When PCILOCK# is sampled negated during the address phase of a transaction in which the PPEC is involved, the PPEC's interface becomes a locked resource until it samples PCILOCK# and FRAME# negated. When other masters attempt accesses to PPEC while it is locked, the PPEC responds with a RETRY termination.
IDSEL	in	**INITIALIZATION DEVICE SELECT:** Used as a chip select during configuration read and write transactions. It is sampled during the address phase of a transaction. If the PPEC samples IDSEL active during configuration read or write and address AD[1:0] = 00, it will respond by asserting DEVSEL# on the next cycle.
DEVSEL#	s/t/s	**DEVICE SELECT:** The PPEC asserts DEVSEL# to claim a PCI transaction as a result of positive decode, and when it samples IDSEL active and address AD[1:0] = 00 during configuration cycles to the PPEC configuration registers.
PAR	t/s	**PARITY:** Parity is even across AD[31:0] and C/BE#[3:0]. The PPEC drives PAR during read data phases when it is a target of a PCI cycle. This signal is an input in all other cases. During an address phase or write data phase in which the PPEC is a target, the PPEC samples this signal to compare it with internally generated parity. Note that PAR signal driving and tri-stating is always one clock delayed from the corresponding AD[31:0] signal driving and tri-stating.

2.1 PCI Bus Interface Signals (Continued)

Name	Type	Description
INTA#	o/d	**PCI INTERRUPT REQUEST A:** This is a level sensitive, active low signal that is used to signal interrupts from the PPEC's PCI-PCMCIA functional block. It is enabled in the PPIRR register. The connection of INTA# to the system interrupt controller is system specific. Note that this signal typically requires an external pull-up resistor.
INTB#	o/d	**PCI INTERRUPT REQUEST B:** This is a level sensitive, active low signal that is used to signal interrupts from the PPEC's PCI-IDE Interface functional block. It is enabled in the PPIRR register. The connection of INTB# to the system interrupt controller is system specific. Note that this signal typically requires an external pull-up resistor.
PERR#	s/t/s	**PARITY ERROR:** This is a sustained tri-state signal that is used to report data parity errors during all transactions for which the PPEC positively decodes address. It is typically used by the system logic to generate an NMI. SERR# is pure open drain, and is actively driven for a single PCI clock. The assertion of SERR# is synchronous with the clock and meets the setup and hold times of all bused signals. However, the restoring of SERR# to the negated state is accomplished by a weak pull-up resistor (same value as for s/t/s) which is provided by the system design, and not by the signaling agent or central resource. This pull-up resistor may take two to three clock periods to fully restore SERR#.
SERR#	o/d	**SYSTEM ERROR:** An open-drain signal that is used to report address parity errors during all transactions in which PPEC is involved. It is typically used by the system logic to generate an NMI. SERR# is pure open drain and is actively driven for a single PCI clock. The assertion of SERR# is synchronous to the clock and meets the setup and hold times of all bused signals. However, the restoring of SERR# to the negated state is accomplished by a weak pull-up resistor (same value as used for s/t/s) that is provided by the system, and not by the signaling agent or central resource. This pull-up may take two to three clock periods to fully restore SERR#.

2.2 System Interrupt Signals

Name	Type	Description
IRQ3	t/s/o or o/d	**SYSTEM INTERRUPT REQUEST IRQ3:** Depending on the PPEC's configuration, this signal can be selected as an active high t/s/o or active low o/d driven signal according to the specific signaling requirement for shared/non-shared ISA/EISA/ Microchannel or other system bus interrupt environments. Note that this signal may require an external pull-up resistor depending on the system configuration and usage.
IRQ4	t/s/o or o/d	**SYSTEM INTERRUPT REQUEST IRQ4:** Depending on the PPEC's configuration, this signal can be selected as an active high t/s/o or active low o/d driven signal according to the specific signaling requirement for shared/non-shared ISA/EISA/ Microchannel or other system bus interrupt environments. Note that this signal may require an external pull-up resistor depending on the system configuration and usage.

2.2 System Interrupt Signals (Continued)

Name	Type	Description
IRQ5	t/s/o or o/d	**SYSTEM INTERRUPT REQUEST IRQ5:** Depending on the PPEC's configuration, this signal can be selected as an active high t/s/o or active low o/d driven signal according to the specific signaling requirement for shared/non-shared ISA/EISA/ Microchannel or other system bus interrupt environments. Note that this signal may require an external pull-up resistor depending on the system configuration and usage.
IRQ7	t/s/o or o/d	**SYSTEM INTERRUPT REQUEST IRQ7:** Depending on the PPEC's configuration, this signal can be selected as an active high t/s/o or active low o/d driven signal according to the specific signaling requirement for shared/non-shared ISA/EISA/ Microchannel or other system bus interrupt environments. Note that this signal may require an external pull-up resistor depending on the system configuration and usage.
IRQ9	t/s/o or o/d	**SYSTEM INTERRUPT REQUEST IRQ9:** Depending on the PPEC's configuration, this signal can be selected as an active high t/s/o or active low o/d driven signal according to the specific signaling requirement for shared/non-shared ISA/EISA/ Microchannel or other system bus interrupt environments. Note that this signal may require an external pull-up resistor depending on the system configuration and usage.
IRQ10	t/s/o or o/d	**SYSTEM INTERRUPT REQUEST IRQ10:** Depending on the PPEC's configuration, this signal can be selected as an active high t/s/o or active low o/d driven signal according to the specific signaling requirement for shared/non-shared ISA/EISA/ Microchannel or other system bus interrupt environments. Note that this signal may require an external pull-up resistor depending on the system configuration and usage.
IRQ11	t/s/o or o/d	**SYSTEM INTERRUPT REQUEST IRQ11:** Depending on the PPEC's configuration, this signal can be selected as an active high t/s/o or active low o/d driven signal according to the specific signaling requirement for shared/non-shared ISA/EISA/ Microchannel or other system bus interrupt environments. Note that this signal may require an external pull-up resistor depending on the system configuration and usage.
IRQ12	t/s/o or o/d	**SYSTEM INTERRUPT REQUEST IRQ12:** Depending on the PPEC's configuration, this signal can be selected as an active high t/s/o or active low o/d driven signal according to the specific signaling requirement for shared/non-shared ISA/EISA/ Microchannel or other system bus interrupt environments. Note that this signal may require an external pull-up resistor depending on the system configuration and usage.
IRQ14	t/s/o or o/d	**SYSTEM INTERRUPT REQUEST IRQ14:** Depending on the PPEC's configuration, this signal can be selected as an active high t/s/o or active low o/d driven signal according to the specific signaling requirement for shared/non-shared ISA/EISA/ Microchannel or other system bus interrupt environments. Note that this signal may require an external pull-up resistor depending on the system configuration and usage.
IRQ15	t/s/o or o/d	**SYSTEM INTERRUPT REQUEST IRQ15:** Depending on the PPEC's configuration, this signal can be selected as an active high t/s/o or active low o/d driven signal according to the specific signaling requirement for shared/non-shared ISA/EISA/ Microchannel or other system bus interrupt requirements. Note that this signal may require an external pull-up resistor depending on the system configuration and usage.

2.3 Audio Interface And Configuration Mode Selection Signal

Name	Type	Description
SPKROUT# /MODE	t/s or o/d	**SPEAKER OUTPUT OR PCMCIA INTERFACE CONFIGURATION MODE:** SPKROUT# /MODE is configured as an input during reset. The user selects 2-Socket Mode with a weak pull-down resistor (10K) on the signal pin, and selects 4-Socket with a weak pull-up resistor. The PPEC automatically reconfigures the pin as an output for use as SPKROUT# after reading and storing the state of the SPKROUT# /MODE pin at the end of the reset sequence. This signal MUST be connected to an external pull-up or pull-down resistor according to the desired operating mode.

2.4 External Power Control Signal

Name	Type	Description
PWRWR#	out	**POWER CONTROL WRITE:** This signal is a write strobe for the PCMCIA Socket Power Control Logic. It is used to latch V_{CC} and V_{PP} power-control information that is transferred via the PCMCIA Socket A data lines. This signal is active during I/O write access to the PPEC's internal PCMCIA Power Control Registers, and during automatic Power-Control write sequences when reset is active or when the PCMCIA auto-power function is activated.

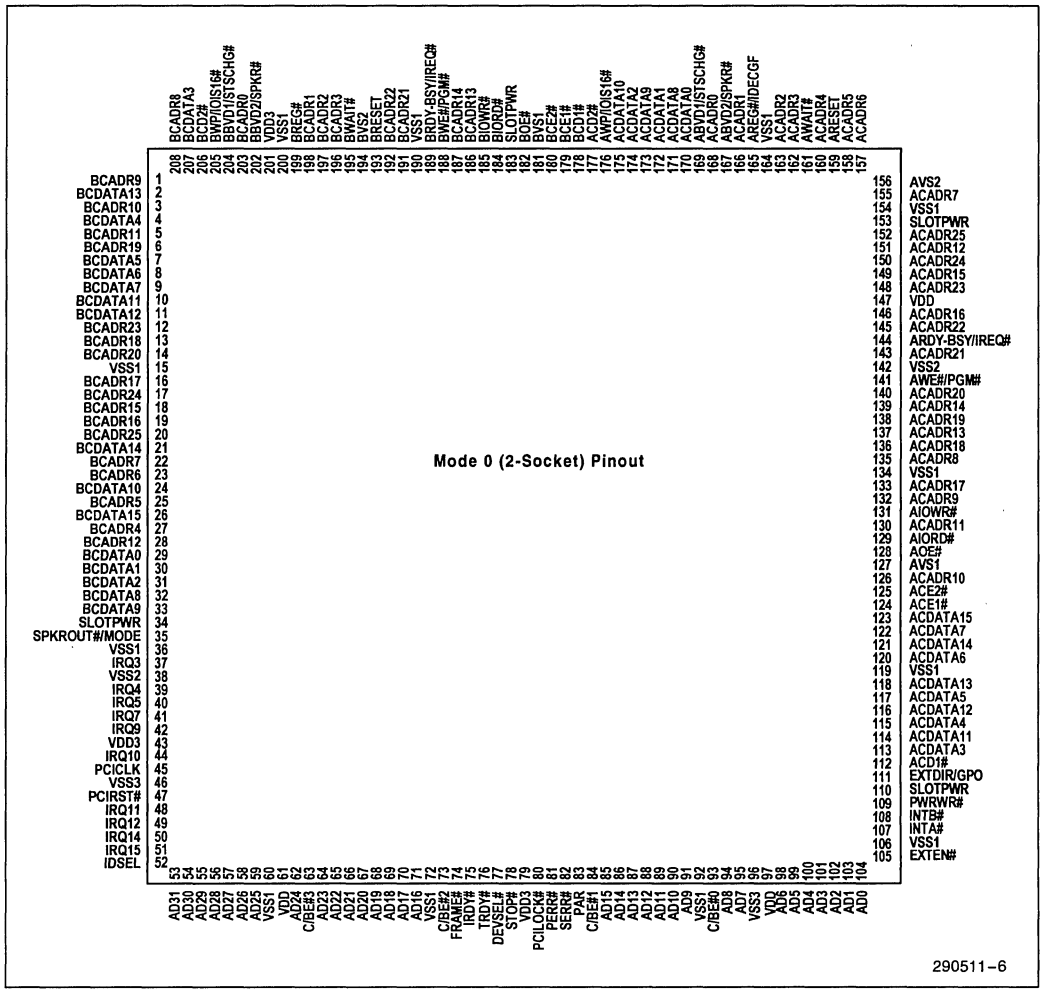
2.5 PCMCIA Interface Signals

The PPEC supports two PCMCIA socket configuration modes: Mode 0 and Mode 1. The number of sockets supported differs with each mode. The PPEC signals therefore change according to the operating mode.

All t/s and t/s/o PCMCIA signals are implemented using 5V/3.3V configurable buffers.

2.5.1 MODE 0 (TWO-SOCKET) CONFIGURATION MODE SIGNALS

Mode 0 configures the PPEC for two PCMCIA sockets: Socket A and Socket B. PCMCIA function signals are identical for both sockets. However, Socket A signals may be multiplexed with IDE interface signals (see Section 2.6) while Socket B signals are not multiplexed, and the Socket A AREG#/IDECFG signal has two functions while the Socket B REG# signal has one function. Figure 4 shows the PPEC signal pinout for Mode 0.



290511-6

Figure 4. Mode 0 Pinout

2.6 Socket A and B PCMCIA Signals

Name	Type	Description
ACDATA[15:0] BCDATA[15:0]	t/s	**SOCKET A AND SOCKET B DATA BUS SIGNALS [15:0]:** This is a 16-bit data bus that is used for data transfer between the PPEC and the PCMCIA card.
ACADR[25:0] BCADR[25:0]	t/s	**SOCKET A AND SOCKET B ADDRESS BUS SIGNALS [25:0]:** This is a 26-bit address bus that is used for addressing memory locations and attribute memory within a 64 MByte PCMCIA address range, and for addressing I/O within a 64 KByte PCMCIA address range.
AIORD# BIORD#	t/s/o	**SOCKET A AND SOCKET B I/O READ:** The PPEC uses this active low signal and the REG# signal to gate I/O Read data from the PC Card. When low, IORD# gates I/O Read data from a memory PC Card only when the REG# signal is also asserted.
AIOWR# BIOWR#	t/s/o	**SOCKET A AND SOCKET B I/O WRITE:** The PPEC uses this active low signal and the REG# signal to gate I/O Write data to the PC Card. When low, IOWR# gates the I/O Write data to the PC Card only when the REG# signal is also asserted.
AREG#/ IDECFG	t/s/o in	**SOCKET A ATTRIBUTE MEMORY SELECT AND IDE CONFIGURATION:** This signal has two functions. During reset it is configured as an input with the IDE Configuration function. The signal level is sampled at the end of the reset sequence. If it is sampled low, all of the Socket A pins are used for PCMCIA Socket-A interface signals; if it is sampled high, some of Socket A pins are used for IDE interface signals. The status of IDECFG pin when sampled is stored in the PPEC IDE Configuration Register so that it can be used to support BIOS software with enhanced auto-configuration capabilities. If it is sampled low at the end of the reset sequence, the signal is configured after reset as an output with the Attribute Memory Select function. The signal is inactive (high) during all normal accesses to main memory of PC Cards. I/O PC Cards do not respond to IORD# or IOWR# assertion when the AREG# signal is inactive. When it is active (low), access is limited to Attribute Memory when WE# or OE# are active, and to I/O ports when IORD# or IOWR# are active. Configurable memory PC Cards and I/O PC Cards contain configuration and status registers in the Attribute Memory Space. This signal pin MUST be either externally pulled-up (IDE interface) with a weak pull-up resistor (10K Ohm) or pulled down (no IDE interface), depending on the system configuration.
BREG#	t/s/o	**SOCKET B ATTRIBUTE MEMORY SELECT:** This signal is inactive (high) during all normal accesses to what is known as Main Memory of the PC Card. I/O PC Cards do not respond to IORD# or IOWR# when the BREG# signal is inactive. When this signal is active (low), access is limited to Attribute Memory when WE# or OE# are active, and to I/O ports when IORD# or IOWR# are active. Configurable memory PC Cards and I/O PC Cards contain configuration and status registers in the Attribute Memory Space.
AOE# BOE#	t/s/o	**SOCKET A AND SOCKET B OUTPUT ENABLE:** This is an active low signal that gates Memory Read data from memory PC Cards.

2.6 Socket A and B PCMCIA Signals (Continued)

Name	Type	Description
ACE[2:1]# BCE[2:1]#	t/s/o	**SOCKET A AND SOCKET B CHIP ENABLE [2:1]:** These are active low signals that are driven by the PPEC when the socket is enabled. CE1# enables even bytes; CE2# enables odd bytes.
AWE#/PGM# BWE#/PGM#	t/s/o	**SOCKET A AND SOCKET B WRITE ENABLE/PROGRAM:** This signal has a single function with two semantics. In WE# semantics it is used by the host to gate Memory Write data. In PGM# semantics it is used for memory PC Cards that employ programmable memory technologies.
ARDY-BSY#/ IREQ# BRDY-BSY#/ IREQ#	in (ST)	**SOCKET A AND SOCKET B READY/BUSY OR INTERRUPT REQUEST:** This signal has two functions. When a memory card is in use, it has the *Ready/Busy* function. The signal is driven low by the memory PC Card to indicate that the card circuits are busy processing a previous write command. READY/BUSY# is set high when the memory PC Card is ready to accept a new data transfer command. When an I/O card is in use, it has the */IREQ#* interrupt request function. The card asserts IREQ# to indicate to the host that a device on the I/O PC Card requires service by the host software. The signal is held at the inactive level when no interrupt is requested. The status of this signal is stored in the Interface Status Register.
AWP/IOIS16# BWP/IOIS16#	in	**SOCKET A AND SOCKET B WRITE PROTECT OR CARD IS 16-BIT PORT:** This signal has two functions. For memory PC Cards, it has the *Write Protect* function that reflects the status of the Write Protect switch on the cards. If a memory PC Card switch is present, this signal is asserted by the card when the switch is enabled (write protection desired), and negated when the switch is disabled. If the memory PC Card has no Write Protect switch, the card connects this line to ground or V_{CC}, depending on the condition of the card memory. If the memory PC Card can always be written, the signal pin is connected to ground. If the memory PC Card is permanently Write Protected, the pin is connected to V_{CC}. The status of WP is stored in the Interface Status Register. However, Memory Write Cycle is not blocked by the xWP signal unless the Write Protect bit is set to 1 in the card memory offset Address High Byte Register. For I/O PC Cards, it has the *"Card is 16-Bit Port"* (IOIS16#) function. The signal is asserted by the PC Card when the address on the bus corresponds to an address to which the PC Card responds, and the I/O Port that is addressed is capable of 16-bit accesses. This signal is used by the PPEC's data assembly/disassembly logic to determine the number of PCMCIA cycles required to complete data transfer (which from the PCI perspective can be 32-bits wide). If this signal is not asserted during a 16-bit I/O access, the PPEC generates two 8-bit data cycles to the even and odd bytes of the 16-bit word which is requested by the initial cycle. The status of this signal is stored in the Interface Status Register.
ARESET BRESET	t/s/o	**SOCKET A AND SOCKET B RESET:** This signal forces a hard reset to the PC card when asserted.

5

2.6 Socket A and B PCMCIA Signals (Continued)

Name	Type	Description
AWAIT# BWAIT#	in	**SOCKET A AND SOCKET B BUS CYCLE WAIT:** This signal is driven by the PC card to delay completion of a memory or I/O cycle that is in progress.
ABVD1/ STSCHG# BBVD1/ STSCHG#-	in	**SOCKET A AND SOCKET B BATTERY VOLTAGE DETECT 1/STATUS CHANGE-RING INDICATE:** This signal has three functions. As Battery Voltage Detect 1, it is driven by a memory PC Card that has a battery to indicate the condition of the battery as follows: • When both BVD1 and BVD2 are asserted (high) the battery is in good condition. • When BVD2 is negated while BVD1 is still asserted, the battery is in a warning condition and should be replaced, although data integrity on the memory PC Card is still assured. When BVD1 is negated (low) with BVD2 either asserted or negated, the battery is no longer serviceable and data is lost. As CHANGED STATUS (STSCHG#), it is held high the when the *Signal on Change* bit or the *Changed* bit in the Card Status Register on the PC Card is set to zero. When both the bits are one, the signal is held low. The Changed bit is the logical OR of the bits CVBAT1, CVBAT2, CWP and CBSYRDY in the Pin Replacement Register on the PC Card. The signal status is stored in the Interface Status Register.
ABVD2/ SPKR# BBVD2/ SPKR#	in	**SOCKET A AND SOCKET B BATTERY VOLTAGE DETECT OR DIGITAL AUDIO:** This signal has two functions. As Battery Voltage Detect, it is driven by a memory PC Card that has a battery to indicate the condition of the battery as follows: • When both BVD1 and BVD2 are asserted (high), the battery is in good condition. • When BVD2 is negated while BVD1 is still asserted, the battery is in a warning condition and should be replaced, although data integrity on the memory PC Card is still assured. When BVD1 is negated (low) with BVD2 either asserted or negated, the battery is no longer serviceable and data is lost. As Digital Audio, it is the input for a single amplitude (digital) audio waveform that is intended to drive the system's speaker via the SPKROUT# output pin on the system (host) interface. The status of the Battery Voltage Detect signal is stored in the Interface Status Register.

2.6 Socket A and B PCMCIA Signals (Continued)

Name	Type	Description
AVS1 BVS1	in	**SOCKET A AND SOCKET B CARD VOLTAGE CAPABILITY SENSE #1:** The PPEC samples this signal and the corresponding xVS2 signal before applying V_{CC} power to the socket to determine the PC card input voltage capability. The signal must be connected to an external pull-up resistor. The status of this signal is stored in the Socket Power Configuration Register
AVS2 BVS2	in	**SOCKET A AND SOCKET B CARD VOLTAGE CAPABILITY SENSE #1:** The PPEC samples this signal before configuring V_{CC} power to the socket to determine the PC card input voltage capability. If it is sampled high, 5V or 3.3V can be applied to the PC Card depending on the state of VS1. If it is sampled low, the PC Card required voltage is less than 3.3V, which the PPEC does not support. The status of this signal is available in the Socket Power Configuration Register.
ACD[2:1]# BCD[2:1]#	in (ST)	**SOCKET A AND SOCKET B CARD DETECT [2:1]:** These are two Card Detect signals that allow verification of proper card insertion. The signals are positioned at opposite ends of the connector to facilitate the detection process, and are connected to ground in the PC Card. The signals are therefore forced low whenever a card is placed in the socket. The status of the signals are available in the Interface Status Register. These signals have internal pull-up resistors.
EXTEN#	out	**EXTEN#:** This is used only in Mode 0, and only when the IDE Interface is not enabled. When the IDE is enabled in Mode 0, the EXTEN# signal pin is reserved. When low, EXTEN# enables external transceivers that de-couple the PCMCIA power latch from the Socket A CDATA[7:0] data bus signals to allow "hot" insertion and removal.
EXTDIR or GPO	out	**EXTERNAL TRANSCEIVER DIRECTION CONTROL OR GENERAL PURPOSE OUTPUT:** This signal has two functions, depending on whether the IDE Interface is enabled, as determined by the state of IDECFG at the end of the reset sequence. If the IDE Interface is disabled, the EXTDIR signal provides direction control for external transceivers. It is high during read cycles, and low during write cycles. If the IDE Interface is enabled, the GPO provides a general-purpose output. The GPO signal is directly controlled by bit 1 of the Global Security Control Register, and can be used to provide global PC Card LOCK function by controlling an external socket locking mechanism.

NOTE:
The PCMCIA Input Acknowledge (INPACK) signal is not supported by the PPEC.

2.6.1 MODE 1 (FOUR-SOCKET) CONFIGURATION SIGNALS

In Configuration Mode 0, two PC Card sockets are available. In Configuration Mode 1, four PC Card sockets are available, requiring more signals than Mode 0. Figure 5 shows the PPEC signal pinout for Mode 1.

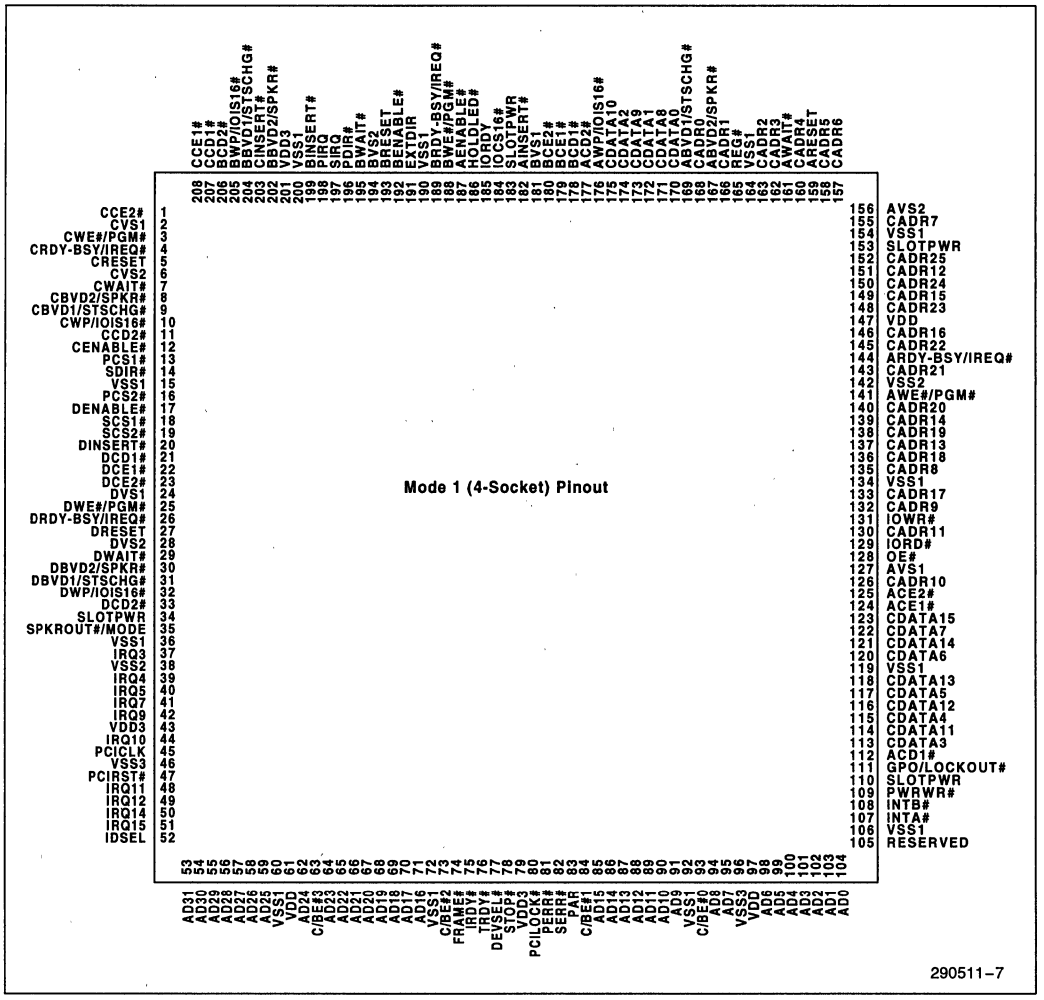


Figure 5. Mode 1 Pinout

Several signals and buses that are dedicated to Socket A in Mode 0 are common to all four sockets in Mode 1, as shown in Table 4.

Table 4. Mode 1 Common Signals

Mode 0 Socket A Signals	Mode 1 Common Socket Signals
ACDATA[15:0]	CDATA [15:0]
ACADR[25:0]	CADR[25:0]
AREG#	REG#
AOE#	OE#
AIORD#	IORD#
AIOWR#	IOWR#

This use of common signals significantly reduces the number of signals (and package pins) that would be required if each socket had it's own dedicated set of signals. It also releases the corresponding Socket B signal pins (BCDATA[15:0], BCADR[25:0], etc.) for use by the following socket-specific signals:

xCE1#	xVS1	xENABLE	xRDY-BUSY#/IREQ#
xCE2#	xVS2	xBVD1	xINSERT
xCD1#	xWP/IOIS16#	xBVD2	xWAIT#
xCD2#	xWE#/PGM#	xRESET	xINSERT#

XINSERT# and HOLDLED# (not listed) are Mode 1 signals that are not used in Mode 0.

All of the signals used in Mode 1 are described in the following table. The sockets have separate, but functionally identical sets of signals. Table 5 lists the PPEC pinout for the various configuration modes.

2.7 Socket A, B, C, and D PCMCIA Signals

Pin Name	Type	Description
CDATA[15:0]	t/s	**COMMON DATA BUS SIGNALS [15:0]:** This is a 16-bit data bus that is used for data transfer between the PPEC and the PCMCIA cards, and the PPEC and IDE drives when the IDE Interface is used. This bus is common to all of the sockets.
CADR[25:0]	t/s	**COMMON ADDRESS BUS SIGNALS [25:0]:** This is a 26-bit address bus that is used for addressing memory locations and attribute memory within a 64 MByte PCMCIA address range, and for addressing I/O within a 64 KByte PCMCIA address range. This bus is common to all of the sockets. If the IDE interface is used, CADR[2:0] transfer IDE register addresses during IDE accesses.
IORD#	t/s/o	**COMMON I/O READ:** The PPEC uses this active low signal and the REG# signal to gate I/O Read data from the PC Card. When low, IORD# gates I/O Read data from a memory PC Card only when the REG# signal is also asserted. This signal is common to all of the sockets.
IOWR#	t/s/o	**COMMON I/O WRITE:** The PPEC uses this active low signal and the REG# signal to gate I/O Write data to the PC Card. When low, IOWR# gates the I/O Write data to the PC Card only when the REG# signal is also asserted. This signal is common to all of the sockets.

2.7 Socket A, B, C, and D PCMCIA Signals (Continued)

Pin Name	Type	Description
REG	t/s/o	**COMMON ATTRIBUTE MEMORY SELECT:** This signal is inactive (high) during all normal accesses to main memory of PC Cards. I/O PC Cards do not respond to IORD# or IOWR# assertion when the AREG# signal is inactive. When it is active (low), access is limited to Attribute Memory when WE# or OE# are active, and to I/O ports when IORD# or IOWR# are active. Configurable memory PC Cards and I/O PC Cards contain configuration and status registers in the Attribute Memory Space. This signal is common to all of the sockets, and requires an external pull-up resistor to prevent the signal from floating during reset.
OE#	t/s/o	**COMMON OUTPUT ENABLE:** This is an active low signal that gates Memory Read data from memory PC Cards. It is common to all of the sockets.
EXTDIR	out	**EXTERNAL TRANSCEIVER DIRECTION CONTROL:** This signal is high during reads, and low during writes, and is used for both high bytes and low bytes. It defaults to write (low) at power-up. EXTDIR is used to control the drive direction of only the PCMCIA transceivers. Separate transceiver controls are provided for the IDE Interface.
ACE#[2:1] BCE#[2:1] CCE#[2:1] DCE#[2:1]	t/s/o	**SOCKET A-D CHIP ENABLE [2:1]:** These are active low signals that are driven by the PPEC when the socket is enabled. CE1# enables even bytes; CE2# enables odd bytes.
AWE#/PGM# BWE#/PGM# CWE#/PGM# DWE#/PGM#	t/s/o	**SOCKET A-D WRITE ENABLE/PROGRAM:** This signal has single function, but with two semantics. In WE# semantics, this signal is used by the host to gate Memory Write data. In PGM# semantics this signal is used for memory PC Cards that employ programmable memory technologies.
ARDY-BSY#/ IREQ BRDY-BSY#/ IREQ# CRDY-BSY#/ IREQ# DRDY-BSY#/ IREQ#	in(ST)	**SOCKET A-D READY/BUSY OR INTERRUPT REQUEST:** This signal has two functions, depending on the card in the socket. When a memory card is in use, it has the *Ready/Busy* function. The signal is driven low by the memory PC Card to indicate that the card circuits are busy processing a previous write command. READY/BUSY# is set high when memory PC Cards are ready to accept a new data transfer command. When an I/O card is in use, it has the */REQ#* interrupt request function. The card asserts IREQ# to indicate to the host that a device on the I/O PC Card requires service by the host software. The signal is held at the inactive level when no interrupt is requested. The status of this signal is available in the Interface Status Register.

2.7 Socket A, B, C, and D PCMCIA Signals (Continued)

Pin Name	Type	Description
AWP/IOIS16# BWP/IOIS16# CWP/IOIS16# DWP/IOIS16#	in	**SOCKET A-D WRITE PROTECT OR CARD IS 16-BIT PORT:** This signal has two functions, depending on the type of card in the socket. For memory PC Cards, it has the *Write Protect* function that reflects the status of the Write Protect switch on the cards. If a memory PC Card switch is present, this signal is asserted by the card when the switch is enabled (write protection desired), and negated when the switch is disabled. If the memory PC Card has no Write Protect switch, the card connects this line to ground or to V_{CC}, depending on the condition of the card memory. If the memory PC Card can always be written, the signal pin is connected to ground. If the memory PC Card is permanently Write Protected, the pin is connected to V_{CC}. The status of WP is available in the Interface Status Register. However, Memory Write Cycle is not blocked by the WP signal unless the Write Protect bit is set to 1 in the card memory offset Address High Byte Register. For I/O PC Cards, it has the *"Card is 16-Bit Port"* (IOIS16#) function. The signal is asserted by the PC Card when the address on the bus corresponds to an address to which the PC Card responds, and the I/O Port that is addressed is capable of 16-bit accesses. This signal is used by the PPEC's data assembly/disassembly logic to determine the number of PCMCIA cycles required to complete data transfers (which from the PCI perspective can be 32-bits wide). If this signal is not asserted during a 16-bit I/O access, the PPEC generates two 8-bit data cycles to the even and odd bytes of the 16-bit word which is requested by the initial cycle. The status of this signal is available in the Interface Status Register.
ARESET BRESET CRESET DRESET	t/s/o	**SOCKET A-D RESET:** This signal forces a hard reset to the PC card when asserted.
AWAIT# BWAIT# CWAIT# DWAIT#	in	**SOCKET A-D BUS CYCLE WAIT:** This signal is driven by the PC card to delay completion of a memory or I/O cycle that is in progress.

2.7 Socket A, B, C, and D PCMCIA Signals (Continued)

Name	Type	Description
ABVD1/ STSCHG# BBVD1/ STSCHG# CBVD1/ STSCHG# DBVD1/ STSCHG#	in	**SOCKET A-D BATTERY VOLTAGE DETECT 1/STATUS CHANGE:** This signal has three functions. As Battery Voltage Detect 1, it is driven by memory a PC Card that has a battery to indicate the condition of the battery, and is used with BVD2 as follows: • When both BVD1 and BVD2 are asserted (high), the battery is in good condition. • When BVD2 is negated while BVD1 is still asserted, the battery is in a warning condition and should be replaced, although data integrity on the memory PC Card is still assured. When BVD1 is negated (low) with BVD2 either asserted or negated, the battery is no longer serviceable and data is lost. As CHANGED STATUS (STSCHG#), it is held high the when the *Signal on Change* bit or the *Changed* bit in the Card Status Register on the PC Card is set to zero. When both the bits are one, the signal is held low. The Changed bit is the logical OR of the bits CVBAT1, CVBAT2, CWP and CBSYRDY in the Pin Replacement Register on the PC Card. The signal status is stored in the Interface Status Register.
ABVD2/ SPKR# BBVD2/ SPKR# CBVD2/ SPKR# DBVD2/ SPKR#	in	**SOCKET A-D BATTERY VOLTAGE DETECT OR DIGITAL AUDIO:** This signal has two functions. As Battery Voltage Detect, it is driven by a memory PC Card that has a battery to indicate the condition of the battery as follows: • When both BVD1 and BVD2 are asserted (high), the battery is in good condition. • When BVD2 is negated while BVD1 is still asserted, the battery is in a warning condition and should be replaced, although data integrity on the memory PC Card is still assured. When BVD1 is negated (low) with BVD2 either asserted or negated, the battery is no longer serviceable and data is lost. As Digital Audio, it is the input for a single amplitude (digital) audio waveform that is intended to drive the system's speaker via the SPKROUT# output pin on the system (host) interface. The status of the Battery Voltage Detect signal is stored in the Interface Status Register.
AVS1 BVS1 CVS1 DVS1	in	**SOCKET A-D CARD VOLTAGE CAPABILITY SENSE #1:** In Mode 1 partially buffered and non-buffered implementations, the PPEC samples this signal and the corresponding xVS2 signal before applying V_{CC} power to the socket to determine the PC card input voltage capability. The signal must be connected to an external pull-up resistor. The status of this signal is stored in the Socket Power Configuration Register.

2.7 Socket A, B, C, and D PCMCIA Signals (Continued)

Pin Name	Type	Description
AVS2 BVS2 CVS2 DVS2	in	**SOCKET A-D CARD VOLTAGE CAPABILITY SENSE #2:** In Mode 1 partially buffered and non-buffered implementations, the PPEC samples this signal and the corresponding xVS1 signal before applying V_{CC} power to the socket to determine the PC card input voltage capability. The signal must be connected to an external pull-up resistor. The status of this signal is stored in the Socket Power Configuration Register.
ACD[2:1]# BCD[2:1]# CCD[2:1]# DCD[2:1]#	in (ST)	**SOCKET A-D CARD DETECT [2:1]:** These are two Card Detect signals that allow verification of proper card insertion. The signals are positioned at opposite ends of the connector to facilitate the detection process, and are connected to ground in the PC Card. The signals are therefore forced low whenever a card is placed in the host socket. The status of the signals are stored in the Interface Status Register. ADC[2:1]# and BCD[2:1]# have internal pull-up resistors; CCD[2:1]# and DCD[2:1]# require external pull-up resistors.
AENABLE# BENABLE# CENABLE# DENABLE#	out	**SOCKET A-D BUFFER ENABLE:** This signal enables the PC Card socket buffers/ transceivers, and is controlled by the corresponding socket Power Control Register.
AINSERT# BINSERT# CINSERT# DINSERT#	in	**SOCKET A-D CARD INSERTION DETECT:** These signals are used in non-buffered or partially-buffered Mode 1 implementations to provide the PPEC with an early indication that a PC Card is being inserted into a PCMCIA socket. xINSERT# requires an external pull-up resistor, and is connected directly to one of the socket pins which is used normally (in fully buffered configuration) for V_{SS} connection. See the "PPEC Design Guide" for a detailed discussion of the use of these pins.
GPO/ LOCKOUT#	out	**GENERAL PURPOSE OUTPUT/LOCKOUT:** When bit 2 of the Global Security Control Register (GSCTRL) is set to 1, the GPO signal is enabled. When bit 2 of the GSTCTRL register is set to 0, the LOCKOUT# signal is enabled. The default value of bit 2 of the GSCTRL register is 0, enabling the LOCKOUT# signal. The GPO signal is directly controlled by bit 1 of the Global Security Control Register, and can be used to provide global PC Card LOCK function by controlling an external socket locking mechanism. LOCKOUT# is intended for systems that require a locking mechanism for the PCMCIA sockets. When the Lockout feature is selected and LOCKOUT# is active, the CD2# for each socket is treated as an Eject Request pin, and CD1# is internally routed to the circuitry that uses CD2#. This feature does not affect the routing of the CDx# signals to the interrupt generator or to the status registers, so a rising edge on either card detect still generates a Card Status Change interrupt. All other internal circuitry that uses CD1# and CD2#, such as the circuitry that tri-states the PCMCIA bus during a card removal, uses CD1# exclusively while the socket is locked.
HOLDLED#	out	**HOLD LED:** This signal is used to indicate a HOLD condition on the PCMCIA Bus during PC Card insertions and removals in non-buffered or partially-buffered Mode 1 implementations.

2.8 IDE Interface Signal

The PPEC supports two Fast Local Bus IDE interfaces (2 connectors with a total of 4 IDE drives) in both PCMCIA interface configuration modes. For clarity, the IDE interface pins are named and defined in this section independently of the basic signals with which they are multiplexed.

Most of the IDE interface signals are multiplexed with Socket A signals in Mode 0. Table 5 lists the signal pinout for Mode 0 with and without IDE selection.

In Mode 1, the IDE interface is independent of the PCMCIA interface with the exception of the data, address, write, and read strobes which are used for both interfaces as follows:

IDE Signal	PCMCIA Signal
DD[15:0]	CDATA[15:0]
DA[2:0]	CADR[2:0]
DIOW#	IOWR#
DIOR#	IORD#

Figure 2 shows the Mode 1 IDE signals.

2.9 IDE Interface Signals

Name	Type	Description
DD[15:0]	t/s	**DRIVE DATA BUS [15:0]:** This is an 8-bit or 16-bit bi-directional data bus that is located between the IDE interface controller, and the drive. The lower 8 bits are used for 8-bit transfers (registers, ECC bytes and, if the drive supports the Features Register, for 8-bit-only data transfers that may be selected). Data signals DD[7:0] can be used to support the IDE Hardware Configuration feature.
DA[2:0]	out	**DRIVE ADDRESS [2:0]:** This is a 3-bit binary coded address asserted by the host to access a drive register or the drive data port.
DIOR#	out	**DRIVE I/O READ STROBE:** The falling edge of DIOR# enables data from a register or from the drive's data port onto the IDE interface data bus. The rising edge of DIOR# latches data into the PPEC.
DIOW#	out	**DRIVE I/O WRITE STROBE:** The rising edge of DIOW# clocks data from the IDE interface data bus into a register or into the drive's data port.
IOCS16#	in	**I/O CHIP-SELECT 16-BIT:** The assertion of IOCS16# indicates to the PPEC that 16-bit data port has been addressed, and that the drive is prepared to send or receive a 16-bit data word.
IORDY	in	**I/O CHANNEL READY:** This signal is driven low by the currently accessed drive to extend the IDE transfer cycle when the drive is not ready to respond to a data transfer request. When IORDY is not negated, it is in a high impedance state from the perspective of the IDE drive(s). An external pull-up resistor must be provided on the cable side of the IDE Interface.
PCS1#	out	**PRIMARY IDE CHIP SELECT FOR DATA/COMMAND I/O ADDRESS RANGE:** This signal can be asserted only when IDEEN# is asserted (low) during IDE cycles to the appropriate address range as defined by the IDE Address Configuration Registers. It selects Command Block Registers of the drive(s) on the Primary IDE Interface.

2.9 IDE Interface Signals (Continued)

Name	Type	Description
PCS2#	out	**PRIMARY IDE CHIP SELECT FOR CONTROL/STATUS I/O ADDRESS RANGE:** This signal can be asserted only when IDEEN# is asserted (low) during IDE cycles to the appropriate address range as defined by the IDE Address Configuration Registers. It selects Control Block Registers of the drive(s) on the Primary IDE Interface.
SCS1#	out	**SECONDARY IDE CHIP SELECT FOR DATA/COMMAND I/O ADDRESS RANGE:** This signal can be asserted only when IDEEN# is asserted (low) during IDE cycles to the appropriate address range as defined by the IDE Address Configuration Registers. It selects Command Block Registers of the drive(s) on the Secondary IDE Interface.
SCS2#	out	**SECONDARY IDE CHIP SELECT FOR CONTROL/STATUS I/O ADDRESS RANGE:** This signal can be asserted only when IDEEN# is asserted (low) during IDE cycles to the appropriate address range as defined by the IDE Address Configuration Registers. It selects Control Block Registers of the drive(s) on the Secondary IDE Interface.
PIRQ	in	**PRIMARY IDE INTERFACE INTERRUPT REQUEST:** This signal is used by the IDE drives on the Primary connector to interrupt the host processor. It can be routed to any system interrupt, and can therefore be internally OR-ed (within the PPEC) with Secondary IDE interrupt.
SIRQ	in	**SECONDARY IDE INTERFACE INTERRUPT REQUEST:** This signal is used by the IDE drives on the Secondary connector to interrupt the host processor. It can be routed to any system interrupt, and can therefore be internally OR-ed (within the PPEC) with Primary IDE interrupt.
PDIR#	out	**PRIMARY IDE TRANSCEIVER DIRECTION CONTROL:** This signal controls the drive direction of the Primary IDE transceivers. The signal is asserted low only during read access to the Primary IDE registers. It is driven high all other times (during writes to Primary IDE registers, and when the Primary IDE is not accessed.
SDIR#	out	**SECONDARY IDE TRANSCEIVER DIRECTION CONTROL:** This signal controls the drive direction of the Secondary IDE transceivers. The signal is asserted low only during read access to the Secondary IDE registers. It is driven high all other times (during writes to Secondary IDE registers, and when the Secondary IDE is not accessed).

2.10 Pin Cross-Reference List

Table 5 lists the PPEC signals in package pin order according to mode. Note that the signal pinout changes in Mode 0 depending on whether the IDE interface is enabled. In Mode 1, the IDE interface is separate from the socket interfaces with the exception of the IDE DD[15:0] data bus, DA[2:0] address bus, DIOR# read strobe, and DIOW# write strobe which are multiplexed with the CDATA[15:0], CADR[2:0], IORD#, and IOWR# common socket signals.

Table 5. Pin Cross-Reference

Pin	Mode 1 Pin Name (Four Sockets)	Mode 0 Pin Name (Two Sockets)	Mode 0 Pin Name (One Socket + IDE)
1	CCE2#	BCADR9	BCADR9
2	CVS1	BCDATA13	BCDATA13
3	CWE#/PGM#	BCADR10	BCADR10
4	CRDY-BSY#/IREQ#	BCDATA4	BCDATA4
5	CRESET	BCADR11	BCADR11
6	CVS2	BCADR19	BCADR19
7	CWAIT#	BCDATA5	BCDATA5
8	CBVD2/SPKR#	BCDATA6	BCDATA6
9	CBVD1/STSCHG#	BCDATA7	BCDATA7
10	CWP/IOIS16#	BCDATA11	BCDATA11
11	CCD2#	BCDATA12	BCDATA12
12	CENABLE#	BCADR23	BCADR23
13	PCS1#	BCADR18	BCADR18
14	SDIR#	BCADR20	BCADR20
15	V_{SS} (OUTPUTS)	V_{SS} (OUTPUTS)	V_{SS} (OUTPUTS)
16	PCS2#	BCADR17	BCADR17
17	DENABLE#	BCADR24	BCADR24
18	SCS1#	BCADR15	BCADR15
19	SCS2#	BCADR16	BCADR16
20	DINSERT#	BCADR25	BCADR25
21	DCD1#	BCDATA14	BCDATA14
22	DCE1#	BCADR7	BCADR7
23	DCE2#	BCADR6	BCADR6
24	DVS1	BCDATA10	BCDATA10
25	DWE#/PGM#	BCADR5	BCADR5
26	DRDY-BSY#/IREQ#	BCDATA15	BCDATA15
27	DRESET	BCADR4	BCADR4
28	DVS2	BCADR12	BCADR12

Table 5. Pin Cross-Reference (Continued)

Pin	Mode 1 Pin Name (Four Sockets)	Mode 0 Pin Name (Two Sockets)	Mode 0 Pin Name (One Socket + IDE)
29	DWAIT#	BCDATA0	BCDATA0
30	DBVD2/SPKR#	BCDATA1	BCDATA1
31	DBVD1/STSCHG#	BCDATA2	BCDATA2
32	DWP/IOIS16#	BCDATA8	BCDATA8
33	DCD2#	BCDATA9	BCDATA9
34	SLOTPWR (5V)	SLOTPWR (5V)	SLOTPWR (5V)
35	SPKROUT#/MODE	SPKROUT#/MODE	SPKROUT#/MODE
36	V_{SS} (OUTPUTS)	V_{SS} (OUTPUTS)	V_{SS} (OUTPUTS)
37	IRQ3	IRQ3	IRQ3
38	V_{SS} (CORE)	V_{SS} (CORE)	V_{SS} (CORE)
39	IRQ4	IRQ4	IRQ4
40	IRQ5	IRQ5	IRQ5
41	IRQ7	IRQ7	IRQ7
42	IRQ9	IRQ9	IRQ9
43	VDD (5V)	VDD (5V)	VDD (5V)
44	IRQ10	IRQ10	IRQ10
45	PCICLK	PCICLK	PCICLK
46	V_{SS} (INPUTS)	V_{SS} (INPUTS)	V_{SS} (INPUTS)
47	PCIRST#	PCIRST#	PCIRST#
48	IRQ11	IRQ11	IRQ11
49	IRQ12	IRQ12	IRQ12
50	IRQ14	IRQ14	IRQ14
51	IRQ15	IRQ15	IRQ15
52	IDSEL	IDSEL	IDSEL
53	AD31	AD31	AD31
54	AD30	AD30	AD30
55	AD29	AD29	AD29
56	AD28	AD28	AD28
57	AD27	AD27	AD27
58	AD26	AD26	AD26
59	AD25	AD25	AD25
60	V_{SS} (OUTPUTS)	V_{SS} (OUTPUTS)	V_{SS} (OUTPUTS)

Table 5. Pin Cross-Reference (Continued)

Pin	Mode 1 Pin Name (Four Sockets)	Mode 0 Pin Name (Two Sockets)	Mode 0 Pin Name (One Socket + IDE)
61	VDD (5V)	VDD (5V)	VDD (5V)
62	AD24	AD24	AD24
63	C/BE3	C/BE3	C/BE3
64	AD23	AD23	AD23
65	AD22	AD22	AD22
66	AD21	AD21	AD21
67	AD20	AD20	AD20
68	AD19	AD19	AD19
69	AD18	AD18	AD18
70	AD17	AD17	AD17
71	AD16	AD16	AD16
72	V_{SS} (OUTPUTS)	V_{SS} (OUTPUTS)	V_{SS} (OUTPUTS)
73	C/BE2	C/BE2	C/BE2
74	FRAME#	FRAME#	FRAME#
75	IRDY#	IRDY#	IRDY#
76	TRDY#	TRDY#	TRDY#
77	DEVSEL#	DEVSEL#	DEVSEL#
78	STOP#	STOP#	STOP#
79	VDD (5V)	VDD (5V)	VDD (5V)
80	PCILOCK#	PCILOCK#	PCILOCK#
81	PERR#	PERR#	PERR#
82	SERR#	SERR#	SERR#
83	PAR	PAR	PAR
84	C/BE1	C/BE1	C/BE1
85	AD15	AD15	AD15
86	AD14	AD14	AD14
87	AD13	AD13	AD13
88	AD12	AD12	AD12
89	AD11	AD11	AD11
90	AD10	AD10	AD10
91	AD9	AD9	AD9
92	V_{SS} (OUTPUTS)	V_{SS} (OUTPUTS)	V_{SS} (OUTPUTS)
93	C/BE0	C/BE0	C/BE0

Table 5. Pin Cross-Reference (Continued)

Pin	Mode 1 Pin Name (Four Sockets)	Mode 0 Pin Name (Two Sockets)	Mode 0 Pin Name (One Socket + IDE)
94	AD8	AD8	AD8
95	AD7	AD7	AD7
96	V_{SS} (INPUTS)	V_{SS} (INPUTS)	V_{SS} (INPUTS)
97	VDD (5V)	VDD (5V)	VDD (5V)
98	AD6	AD6	AD6
99	AD5	AD5	AD5
100	AD4	AD4	AD4
101	AD3	AD3	AD3
102	AD2	AD2	AD2
103	AD1	AD1	AD1
104	AD0	AD0	AD0
105	RESERVED	EXTEN#	RESERVED
106	V_{SS} (OUTPUTS)	V_{SS} (OUTPUTS)	V_{SS} (OUTPUTS)
107	INTA#	INTA#	INTA#
108	INTB#	INTB#	INTB#
109	PWRWR#	PWRWR#	PWRWR#
110	SLOTPWR (5V)	SLOTPWR (5V)	SLOTPWR (5V)
111	GPO/LOCKOUT#	EXTDIR	GPO
112	ACD1#	ACD1#	RESERVED
113	CDATA3	ACDATA3	DD3
114	CDATA11	ACDATA11	DD11
115	CDATA4	ACDATA4	DD4
116	CDATA12	ACDATA12	DD12
117	CDATA5	ACDATA5	DD5
118	CDATA13	ACDATA13	DD13
119	V_{SS} (OUTPUTS)	V_{SS} (OUTPUTS)	V_{SS} (OUTPUTS)
120	CDATA6	ACDATA6	DD6
121	CDATA14	ACDATA14	DD14
122	CDATA7	ACDATA7	DD7
123	CDATA15	ACDATA15	DD15
124	ACE1#	ACE1#	RESERVED
125	ACE2#	ACE2#	RESERVED
126	CADR10	ACADR10	PCS2#
127	AVS1	AVS1	RESERVED

Table 5. Pin Cross-Reference (Continued)

Pin	Mode 1 Pin Name (Four Sockets)	Mode 0 Pin Name (Two Sockets)	Mode 0 Pin Name (One Socket + IDE)
128	OE#	AOE#	PDIR#
129	IORD#	AIORD#	DIOR#
130	CADR11	ACADR11	PCS1#
131	IOWR#	AIOWR#	DIOW#
132	CADR9	ACADR9	RESERVED
133	CADR17	ACADR17	RESERVED
134	V_{SS} (OUTPUTS)	V_{SS} (OUTPUTS)	V_{SS} (OUTPUTS)
135	CADR8	ACADR8	RESERVED
136	CADR18	ACADR18	RESERVED
137	CADR13	ACADR13	SCS1#
138	CADR19	ACADR19	RESERVED
139	CADR14	ACADR14	RESERVED
140	CADR20	ACADR20	RESERVED
141	AWE#/PGM#	AWE#/PGM#	RESERVED
142	V_{SS} (CORE)	V_{SS} (CORE)	V_{SS} (CORE)
143	CADR21	ACADR21	RESERVED
144	ARDY-BSY#/IREQ#	ARDY-BSY#/IREQ#	IORDY
145	CADR22	ACADR22	RESERVED
146	CADR16	ACADR16	RESERVED
147	VDD (5V)	VDD (5V)	VDD (5V)
148	CADR23	ACADR23	RESERVED
149	CADR15	ACADR15	SDIR#
150	CADR24	ACADR24	PIRQ
151	CADR12	ACADR12	SCS2#
152	CADR25	ACADR25	SIRQ
153	SLOTPWR (5V)	SLOTPWR (5V)	SLOTPWR (5V)
154	V_{SS} (OUTPUTS)	V_{SS} (OUTPUTS)	V_{SS} (OUTPUTS)
155	CADR7	ACADR7	RESERVED
156	AVS2	AVS2	RESERVED
157	CADR6	ACADR6	RESERVED
158	CADR5	ACADR5	RESERVED
159	ARESET	ARESET	RESERVED
160	CADR4	ACADR4	RESERVED

Table 5. Pin Cross-Reference (Continued)

Pin	Mode 1 Pin Name (Four Sockets)	Mode 0 Pin Name (Two Sockets)	Mode 0 Pin Name (One Socket + IDE)
161	AWAIT#	AWAIT#	RESERVED
162	CADR3	ACADR3	RESERVED
163	CADR2	ACADR2	DA2
164	V_{SS} (OUTPUTS)	V_{SS} (OUTPUTS)	V_{SS} (OUTPUTS)
165	REG#	AREG#/IDECFG	AREG#/IDECFG
166	CADR1	ACADR1	DA1
167	ABVD2/SPKR#	ABVD2/SPKR#	RESERVED
168	CADR0	ACADR0	DA0
169	ABVD1/STSCHG#	ABVD1/STSCHG#	RESERVED
170	CDATA0	ACDATA0	DD0
171	CDATA8	ACDATA8	DD8
172	CDATA1	ACDATA1	DD1
173	CDATA9	ACDATA9	DD9
174	CDATA2	ACDATA2	DD2
175	CDATA10	ACDATA10	DD10
176	AWP/IOIS16#	AWP/IOIS16#	IOCS16#
177	ACD2#	ACD2#	RESERVED
178	BCD1#	BCD1#	BCD1#
179	BCE1#	BCE1#	BCE1#
180	BCE2#	BCE2#	BCE2#
181	BVS1	BVS1	BVS1
182	AINSERT#	BOE#	BOE#
183	SLOTPWR (5V)	SLOTPWR (5V)	SLOTPWR (5V)
184	IOCS16#	BIORD#	BIORD#
185	IORDY	BIOWR#	BIOWR#
186	HOLDLED#	BCADR13	BCADR13
187	AENABLE#	BCADR14	BCADR14
188	BWE#/PGM#	BWE#/PGM#	BWE#/PGM#
189	BRDY-BSY#/IREQ#	BRDY-BSY#/IREQ#	BRDY-BSY#/IREQ#
190	V_{SS} (OUTPUTS)	V_{SS} (OUTPUTS)	V_{SS} (OUTPUTS)
191	EXTDIR	BCADR21	BCADR21
192	BENABLE#	BCADR22	BCADR22
193	BRESET	BRESET	BRESET

Table 5. Pin Cross-Reference (Continued)

Pin	Mode 1 Pin Name (Four Sockets)	Mode 0 Pin Name (Two Sockets)	Mode 0 Pin Name (One Socket + IDE)
194	BVS2	BVS2	BVS2
195	BWAIT#	BWAIT#	BWAIT#
196	PDIR#	BCADR3	BCADR3
197	SIRQ	BCADR2	BCADR2
198	PIRQ	BCADR1	BCADR1
199	BINSERT#	BREG#	BREG#
200	V_{SS} (INPUTS)	V_{SS} (INPUTS)	V_{SS} (INPUTS)
201	VDD (5V)	VDD (5V)	VDD (5V)
202	BBVD2/SPKR#	BBVD2/SPKR#	BBVD2/SPKR#
203	CINSERT#	BCADR0	BCADR0
204	BBVD1/STSCHG#	BBVD1/STSCHG#	BBVD1/STSCHG#
205	BWP/IOIS16#	BWP/IOIS16#	BWP/IOIS16#
206	BCD2#	BCD2#	BCD2#
207	CCD1#	BCDATA3	BCDATA3
208	CCE[1]#	BCADR[8]	BCADR[8]

3.0 PPEC REGISTER DESCRIPTIONS

The PPEC registers consist of PCI configuration registers, and PCMCIA Socket Configuration registers. The PCI configuration registers are classified as PCI-PCMCIA Bridge PCI Configuration Registers, or PCI-IDE Interface PCI Configuration Registers. The PCMCIA Socket Configuration registers are classified as General Setup Registers, Interrupt Registers, I/O Mapping Control Registers, or Memory Mapping Control Registers.

Several PPEC registers contain reserved bits or fields labeled "Reserved" that must be handled correctly by software. During reads, software must mask the reserved bits as undefined. During writes, software must ensure that the values of the reserved bits are preserved by first reading the reserved bits, merging the reserved bit values with the new values of the non-reserved bits, then writing the data.

Several bits are described as "not implemented" in the register descriptions. These bits correspond to PCI-defined functions that are not implemented in the PPEC, but are described to facilitate tracking of PCI-supported features.

3.1 PCI Configuration Registers

The PPEC supports PCI-PCMCIA Bridge and PCI-IDE Interface functions. These functions can be configured independently with two sets of PCI configuration registers in compliance with the PCI Local Bus Specification Revision 2.0. The two sets of configuration registers are accessed through a mechanism defined for multi-functional PCI devices. The PCI-PCMCIA Bridge configuration registers are addressed as a function #0 with AD[10:8] as shown in the following table, and the PCI-IDE Interface configuration registers are addressed as a function #1. Attempted access of a register in the 2–7 function range results in no response by the PPEC and a PCI-master abort.

Functions are accessed by AD[10:8] during the address phase of the configuration cycle as follows:

AD[10:8]	PPEC PCI Function Addressed
000	#0: PCI-PCMCIA Bridge
001	#1: PCI-IDE Interface
010 through 111	none (Reserved)

Note that the control bits for certain PCI functions that are defined in the PCI Specification but not used in the PPEC are shown in the configuration registers, but are described as "not implemented".

Table 6. PCI-PCMCIA Bridge PCI Configuration Registers

Address Offset	Mnemonic	Register Name	Access
00–01h	VENID	Vendor ID	RO
02–03h	DEVID	Device ID	RO
04–05h	PCICMD	PCI Command	R/W
06–07h	PCISTS	PCI Status	RO
08h	REVID	Revision ID	RO
09–0Bh	CCODE	Class Code (CCPIB and CCCB)	RO
0C–0Dh		Reserved	
0Eh	HTYPE	Header Type	RO
0Fh		Reserved	
10–13h	PBBA	PCI-PCMCIA Bridge Base Address	R/W
14–3Bh		Reserved	
3Ch	INTLIN	Interrupt Line	R/W
3Dh	INTPIN	Interrupt Pin	R/W
3E–3Fh		Reserved	
40h	PCICON	PCI Configuration Control	R/W
41–4Fh		Reserved	
50h	PPIRR	PCMCIA-PCI Interrupt Routing Register	
51–FFh		Reserved	

3.1.1 PPEC FUNCTION #0—PCI-PCMCIA BRIDGE PCI CONFIGURATION REGISTERS

The PCI-PCMCIA Bridge PCI Configuration Registers, listed in Table 6, are 8-bit, 16-bit, and 32-bit registers. Particular bytes within 16-bit and 32-bit register are selected with byte enables. Reserved registers and bits are reserved for future use, and writing to them has no effect. When writing to a register with reserved bits, the reserved bits should be read first, then properly masked and written back to prevent future software incompatibility.

3.1.1.1 VENID—Vendor ID

Register Offset: 00h
Default Value: 8086h
Access: Read Only
Size: 16 bits

This is a unique 16 bit value assigned to a vendor that, together with the Device ID, uniquely identifies each PCI device. Writes to this register have no effect.

Bits[15:0]: Vendor Identification
This is a 16-bit value assigned to Intel.

3.1.1.2 DEVID—Device ID

Register Offset: 02h
Default Value: 1221h
Access: Read Only
Size: 16 bits

This is a unique 16 bit value that is assigned to the PCI-PCMCIA Bridge function. The Device ID, together with the Vendor ID, uniquely identifies each PCI device. Writes to this register have no effect.

Bits[15:0]: Device Identification
This value identifies the PCI-PCMCIA Bridge function.

3.1.1.3 PCICMD—PCI Command

Register Offset: 04h
Default Value: 0000h
Access: Read/Write
Size: 16 bits

This 16 bit register contains PCI control information.

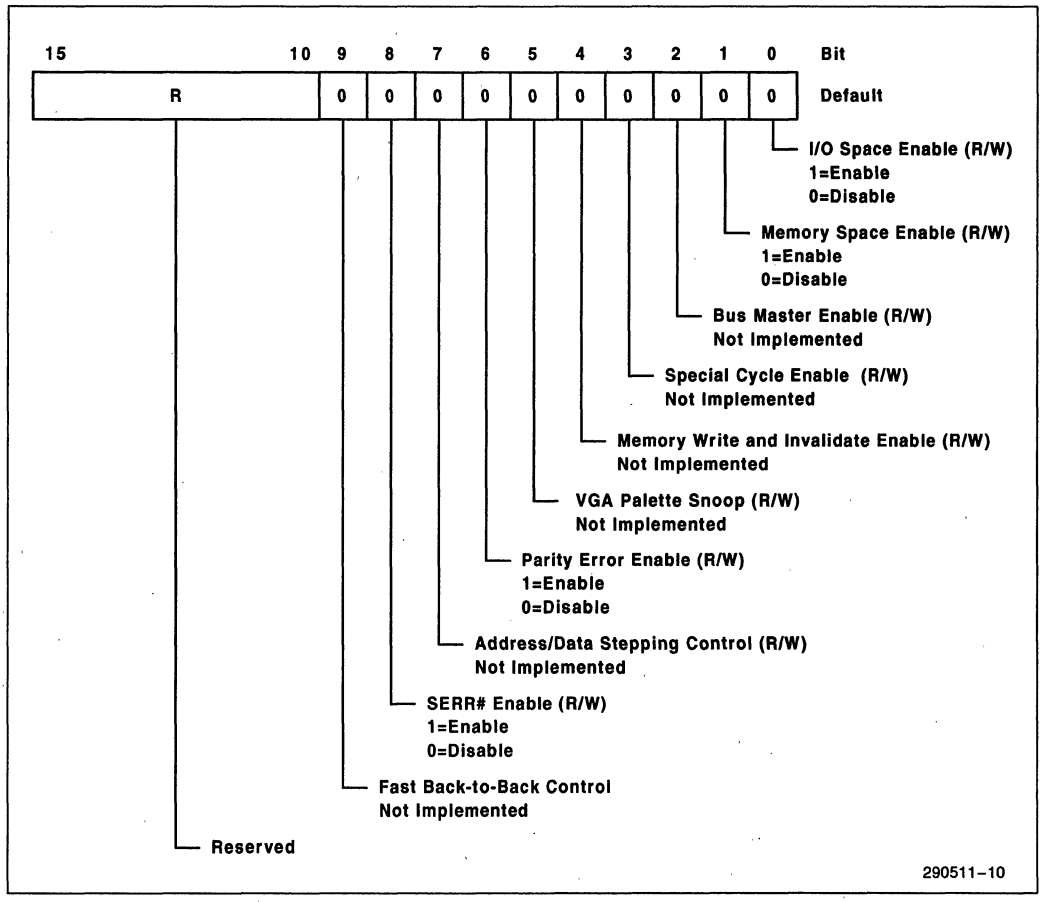


Figure 6. PCI Command Register

Bits[15:10]: Reserved

Bit 9: Fast Back-to-Back Control
This control function can be used only by a PCI master and is therefore not implemented in the PPEC. The bit always reads "0" (disabled).

Bit 8: SERR# Enable
When this bit is set to 1, the PPEC asserts SERR# when it detects a parity error during an address phase, and when it detects an X.X PC Card in Mode 1, partially-buffered implementations. When this bit is set to 0, SERR# is not asserted for any reason. Reset sets the bit to 0.

Bit 7: Address/Data Stepping Control
This control function can be used only by a PCI master and is therefore not implemented in the PPEC. The bit always reads "0" (disabled).

Bit 6: Parity Error Enable
This bit controls PPEC response to PCI data parity errors. When this bit is set to 1, the PPEC activates PERR# when it detects a parity error during a data phase. When this bit is set to 0, the PPEC ignores parity errors. Reset sets the bit to 0 and disables data parity checking.

Bit 5: VGA Palette Snoop
This bit is intended only for specific control of PCI-based VGA devices, and is not applicable to the PPEC. The bit is not implemented, and always reads "0".

Bit 4: Memory Write and Invalidate Enable
This control function can be used only by the PCI master, and is therefore not implemented in the PPEC. The bit always reads "0" (disabled).

Bit 3: Special Cycle Enable
This bit is intended to enable response to supported special cycles. Since the PPEC does not respond to any special cycle, the bit is not implemented and always reads "0".

Bit 2: Bus Master Enable
This bit is intended to enable mastership of the PCI. Since the PPEC cannot be a PCI master, the bit is not implemented and always reads "0".

Bit 1: Memory Space Enable
This bit enables the PPEC to accept PCI-originated memory cycles. When the bit is set to 0, the PPEC does not respond to PCI memory cycles to PCMCIA cards, and the PPEC DEVSEL# logic is inhibited during the memory cycles.

Bit 0: I/O Space Enable
This bit enables the PCI-PCMCIA Bridge to accept PCI-originated I/O cycles. When the bit is set to 0, the PPEC does not respond to PCI master I/O cycles, and the PPEC DEVSEL# logic is inhibited during the I/O cycles.

3.1.1.4 PCISTS—PCI Status

Register Offset:	06h
Default Value:	0480h
Access:	Read Only (*see description)
Size:	16-bits

This 16-bit register is used to record status information for PCI bus-related events. Reads to this register behave normally. Writing bits 11, 14, and 15 to 1 set the bits to 0. The other register bits cannot be written.

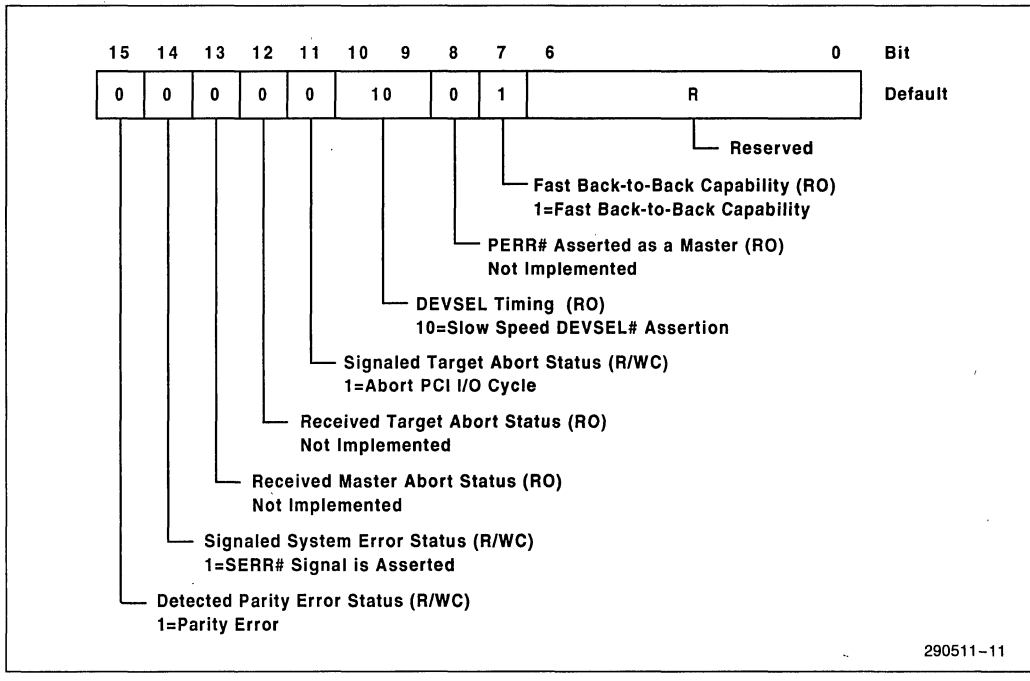


Figure 7. PCI Status Register

5

Bits 15: Detected Parity Error Status
This bit is to be set whenever the 82092AA as a target detects a parity error during the data phase, even if parity error handling is disabled (as controlled by bit 6 in the *PCI Command* register). DEFAULT = 0.

Bit 14: Signaled System Error Status
This bit is used to indicate that PCI device asserted SERR# signal. The 82092AA will assert this bit whenever it generates address phase parity error via SERR# pin. DEFAULT = 0.

Bit 13: Received Master Abort Status (not applicable)
Since this control function can be used only by the PCI master it is not implemented and the bit is always read as a "0" (disabled).

Bit 12: Received Target Abort Status (not applicable)
Received Target Abort Sataus by the PCI master. Since this control function can be used only by the PCI master it is not implemented and the bit is always read as a "0" (disabled).

Bit 11: Signaled Target Abort Status
This bit is to be set by PCI target devices when they generate a Target Abort.

The 82092AA generates Target Abort in the case when it is the target of the PCI I/O cycle and address/byte-enable combination is invalid. More details are provided in Section 1.6.1.5, Transaction Termination.

Bit[10:9]: DEVSEL Timing
These read-only bits encode the timing of DEVSEL# when 82092AA responds as a Target. PCI Specification defines three allowable timings for assertion of DEVSEL#: 00b = fast, 01b = medium, and 10b = slow (11b is reserved). These bits are Read-Only and they indicate the slowest time that a device asserts DEVSEL# for any bus command except *Configuration Read* and *Configuration Write*. The 82092AA PCI-IDE function implements medium speed DEVSEL# timing and therefore these bits contain value 01b.

Bit 8: PERR# Asserted as a Master
This control function can be used only by a PCI master. Therefore this control function is not implemented and the bit will be always read as a "0" (disabled).

Bit 7: Fast Back-to-Back Capability
This read-only bit indicates PCI target capability to support fast back-to-back cycles. The 82092AA can support this type of cycle originated by any PCI master and therefore this bit is set to "1". DEFAULT = 1.

Bit[6:0]: Reserved

3.1.1.5 REVID—Revision ID

Register Offset:	08h
Default Value:	01h
Access:	Read Only
Size:	8 bits

This 8 bit register contains device revision information. Writes to this register have no effect.

Bits[7:0]: Revision Identification
This is the revision level of the PPEC. The initial PPEC revision level is 01h.

3.1.1.6 CCPIB—Class Code-Programming Interface Byte

Register Offset:	09h
Default Value:	00h
Access:	Read Only
Size:	8 bits

This 8 bit register contains device Programming Interface information related to the Class Code bytes located at 0Ah offset. Writes to this register have no effect.

Bits[7:0]: Programming Interface
There are no specific register-level programming interfaces defined for this Class Code (indicated by register CCCB). Therefore, the value of this field is 0.

3.1.1.7 CCCB—Class Code-Class Code Bytes

Register Offset:	0Ah
Default Value:	0605h
Access:	Read Only
Size:	16 bits

This 16-bit register contains device Class Code bytes in the following format: [BASE CLASS][SUBCLASS]. Writes to this register have no effect.

Bits[15:8]: Base Class
The value 06h in this field identifies the function class as a *bus bridge*.

Bits[7:0]: Sub-Class
The value 05h in this field identifies the function sub-class as a *PCMCIA bridge*.

3.1.1.8 HTYPE—Header Type

Register Offset: 0Eh
Default Value: 80h
Access: Read Only
Size: 8 bits

This register indicates whether or not the device contains multiple functions, and identifies the layout of bytes 10h through 3Fh in configuration space. Bit 7 indicates a multi-functional device when set to 1. Bits[6:0] specify layout of bytes 10h-3Fh. The PPEC uses layout type #0 as defined in the PCI specification. Writes to this register have no effect.

Bit 7: Multifunction Indicator
This bit is set to 1 to indicate that the PPEC is a multifunctional device.

Bits[6:0]: Byte Layout
This field specifies layout type "0" for bytes 10-3Fh, as defined in the PCI specification.

3.1.1.9 PBBA—PCI-PCMCIA Bridge Base Address

Register Offset: 10h
Default value: 0000 0001h
Access: Read/Write
Size: 32 bits

This register determines the starting address of the PCMCIA Index/Data Socket Configuration registers mapped in the system I/O space. PCMCIA Index/Data Socket Configuration registers allow indirect access to the block of PCMCIA socket control registers which consists of 256 8-bit locations divided into four sub-blocks, each containing 64 8-bit configuration registers that control the operations of particular PCMCIA socket.

PCMCIA Index/Data Registers can be mapped anywhere in 4 GByte I/O space on a Dword boundary. They provide 82365SL-compatible windowing access to the PCMCIA socket control/configuration registers. The 82365SL-compatible PCMCIA configuration registers access via Index/Data registers implies that Index is an 8-bit I/O port located at [BASEADDRESS], and Data is an 8-bit I/O port located at [BASEADDRESS] + 1.

Bits[31:2]: Base Address
This value determines the starting address of the PCMCIA Index/Data Socket Configuration registers mapped in the system I/O space.

Bit 1: Reserved

Bit 0: I/O Space Indicator
This bit is set to 1 to indicate I/O space.

NOTE:
Accesses to locations BASEADDRESS + 2 and BASEADDRESS + 3 are not permitted. These accesses may cause errors that are not reported to the system.

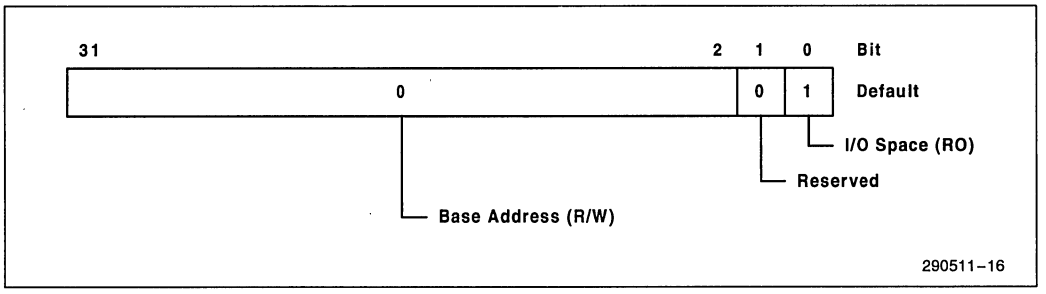


Figure 8. PCI-PCMCIA Bridge Base Address Register

3.1.1.10 INTLIN—Interrupt Line

Register Offset: 3Ch
Default Value: 00h
Access: Read/Write
Size: 8 bits

This register is used to communicate interrupt line routing information. BIOS software must initialize this register during system configuration. The value in this register identifies the interrupt request level of the system interrupt controller(s) to which the PPEC interrupt pin is connected. Device drivers and the operating system can use this information to determine priority and vector information. The value in this register is system architecture specific.

Bits[7:0]: Interrupt Line Identification
The value in this field identifies the interrupt request level of the system interrupt controller(s) to which the PPEC interrupt pin is connected.

NOTE:
The PCI-PCMCIA Bridge can connect to system interrupt controllers in two different ways:

a. via a single PCI interrupt signal line INTA# that requires additional routing (system specific).
b. via 10 direct system interrupt signals.

Mode (b) is provided to allow PCMCIA software that requires "ISA compatibility" (i.e., non-shareable IRQ handlers that require specific IRQ level) to run without modifications on Intel-architecture platforms with the PPEC as a PCI-PCMCIA Bridge. In this case, the PCI interrupt scheme is not used, but the PCMCIA Card interrupts (as well as the Card Status Change interrupt) are configured to connect directly to specific system IRQ lines. The PCI Interrupt Line and the PCMCIA-PCI Interrupt Routing Register (PPIRR) remain in default state 0. Designs that are not dependent on this type of software "compatibility" can use PCI interrupt scheme using mechanism (a).

3.1.1.11 INTPIN—Interrupt Pin

Register Offset: 3Dh
Default Value: 01h
Access: Read Only
Size: 8 bits

This register is used to indicate that the PCI-PCMCIA Bridge uses the INTA# PCI Interrupt Pin for signaling PC Card interrupts (Card Status Change and/or I/O Interrupts). The PCMCIA-PCI Interrupt Routing Register (PPIRR, located at offset 50h) is used to enable (per PC Card) signaling of interrupts using the PCI interrupt scheme.

Bits[7:0]: Interrupt Pin Selection
The value in this field, 01h, identifies the interrupt pin used by the PPEC's PCI-PCMCIA Bridge function as INTA#.

3.1.1.12 PCICON—PCI Configuration Control

Register Offset: 40h
Default Value: XXh
Access: Read/Write
Size: 8 bits

The default is determined by the PCMCIA and IDE configurations as defined in the PCMCIA Interface description in this document.

Bit 0: PCICLK Configuration
This read/write bit defines the system PCICLK frequency. It must be initialized by the system software to provide optimized timing for 25 MHz or 33 MHz. as follows:

1 = 25 MHz PCICLK
0 = 33 MHz PCICLK

Bits[7:6]: Reserved

Bit 5: Enhanced PCMCIA Timing Mode Enable
When set to 1, this bit enables enhanced PCMCIA timing mode. When set to 0, the PPEC timing is 82365SL compatible manner as far as timing control based on the SMSTH0 PCMCIA memory window control register is concerned. The slowest 365 timing is selected and all writes to the enhanced timing mode bits are ignored. When set to 1, the enhanced timing control is enabled.

Bit 4: Global PCMCIA Read-Prefetch Buffering Enable
This bit globally enables PCI to PCMCIA data buffering for Prefetch Read operations when set to 1. When set to 0, buffered operations are disabled.

Bit 3: Global PCMCIA Post-Write Buffering Enable
This bit globally enables PCI to PCMCIA data buffering for Post Write operations when set to 1. When set to 0, buffered operations are disabled.

This register provides read-only configuration information and data buffering enable/disable function.

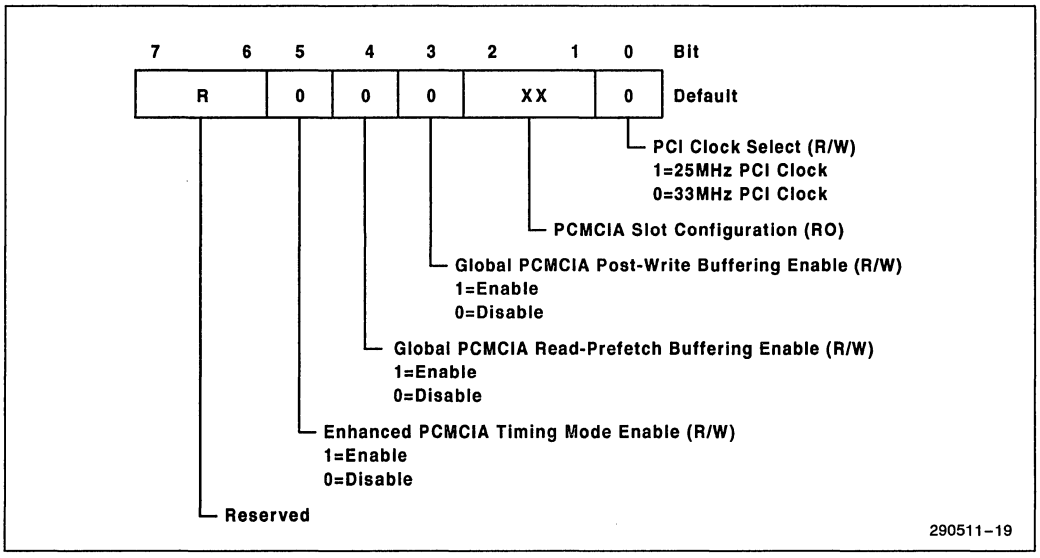


Figure 9. PCI Configuration Control Register

Bits[2:1]: PCMCIA Socket Configuration
These two read-only bits define the PCMCIA socket configuration (i.e., the number of supported sockets based on PCMCIA interface and IDE interface pin configuration) as follows:

Bit2	Bit1	Configuration
0	0	2 PCMCIA Sockets (2-Socket Mode)
0	1	1 PCMCIA Socket And IDE (2-Socket Mode)
1	X	4 PCMCIA Sockets (4-Socket Mode)

3.1.1.13 PPIRR—PCMCIA-PCI Interrupt Routing Register

Register Offset: 50h
Default Value: 00h
Access: Read/Write
Size: 8 bits

This register allows mapping of PCMCIA interrupts to either ISA interrupts, or PCI interrupts. Two interrupts, Card Status Interrupt and Card I/O Interrupt, can be generated and independently routed for each socket. When a register bit is set to 0, the corresponding interrupt is routed via one of the 10 system interrupt lines (ISA mechanism) as specified in the Interrupt and General Control Register and the Card Status Change Interrupt Configuration Register. When a bit is set to 1, the corresponding interrupt is routed via INTA# (PCI mechanism).

Bit 7: Socket D Card I/O IRQ Interrupt Routing
This bit selects Card I/O Interrupt routing via the PCI mechanism, or the ISA mechanism.

Bit 6: Socket D Card Status Change Interrupt Routing
This bit selects Card Status Change Interrupt routing via the PCI mechanism, or the ISA mechanism.

Bit 5: Socket C Card I/O IRQ Interrupt Routing
This bit selects Card I/O Interrupt routing via the PCI mechanism, or the ISA mechanism.

Bit 4: Socket C Card Status Change Interrupt Routing
This bit selects Card Status Change Interrupt routing via the PCI mechanism, or the ISA mechanism.

Bit 3: Socket B Card I/O IRQ Interrupt Routing
This bit selects Card I/O Interrupt routing via the PCI mechanism, or the ISA mechanism.

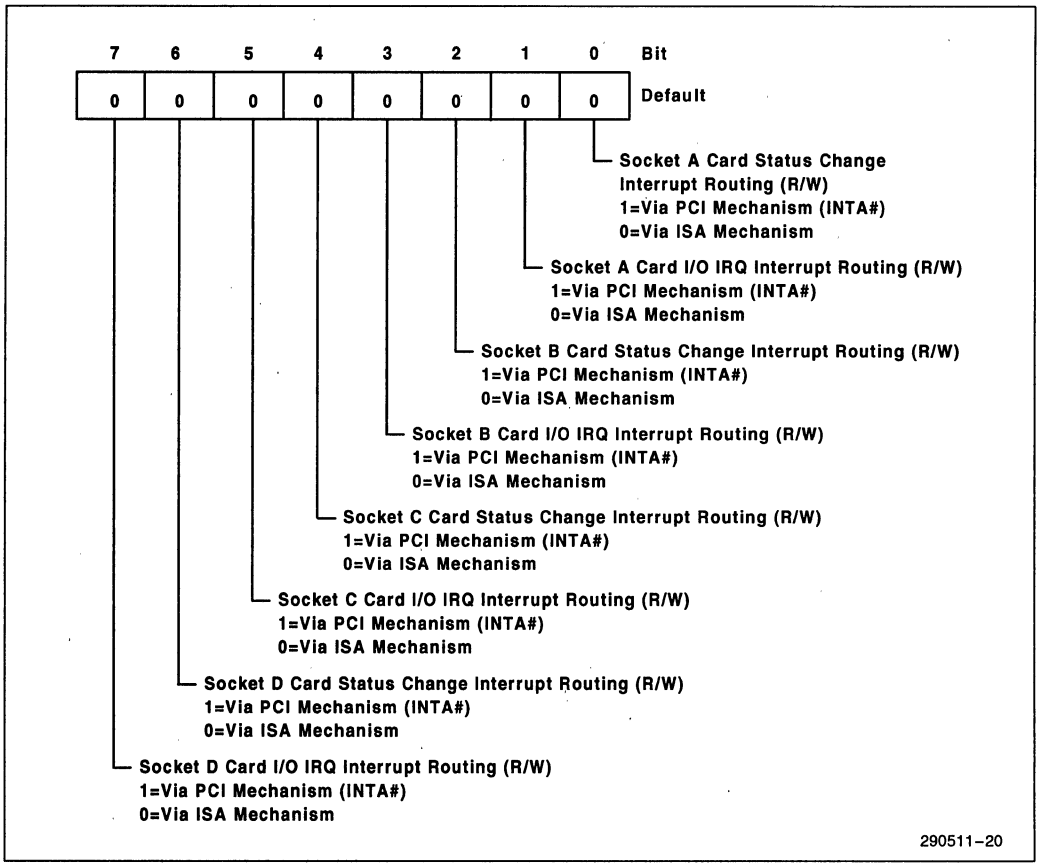


Figure 10. PCMCIA Interrupt Routing Register

Bit 2: Socket B Card Status Change Interrupt Routing
This bit selects Card Status Change Interrupt routing via the PCI mechanism, or the ISA mechanism.

Bit 1: Socket A Card I/O IRQ Interrupt Routing
This bit selects Card I/O Interrupt routing via the PCI mechanism, or the ISA mechanism.

Bit 0: Socket A Card Status Change Interrupt Routing
This bit selects Card Status Change Interrupt routing via the PCI mechanism, or the ISA mechanism.

3.1.2 PPEC Function #1—PCI-IDE Interface PCI Configuration Registers

The PCI-PCMCIA Bridge PCI Configuration Registers, listed in Table 7, are 8-bit, 16-bit, and 32-bit registers. Particular bytes within 16-bit and 32-bit registers are selected with byte enables. Reserved registers and bits are reserved for future use, and writing to them has no effect. When writing to a register with reserved bits, the reserved bits should be read first, then properly masked and written back to prevent future software incompatibility.

Table 7. PCI-IDE Interface PCI Configuration Registers

Address Offset	Mnemonic	Register Name	Access
00-01h	VENID	Vendor ID	RO
02-03h	DEVID	Device ID	RO
04-05h	PCICMD	PCI Command	R/W
06-07h	PCISTS	PCI Status	RO
08h	REVID	Revision ID	RO
09-0Bh	CCODE	Class Code (CCPIB and CCCB)	RO
0C-0Dh		Reserved	
0Eh	HTYPE	Header Type	RO
0Fh		Reserved	
10-13h	PDCBA	IDE Base Address #0—Primary IDE Data/Command	R/W
14-17h	PCSBA	IDE Base Address #1—Primary IDE Control/Status	R/W
18-1Bh	SDCBA	IDE Base Address #2—Secondary IDE Data/Command	R/W
1C-1Fh	SCSBA	IDE Base Address #3—Secondary IDE Control/Status	R/W
20-3Bh		Reserved	
3Ch	INTLIN	Interrupt Line	R/W
3Dh	INTPIN	Interrupt Pin	R/W
3E-3Fh		Reserved	
40h	PCICON	PCI Configuration Control	R/W
41-43h		Reserved	
44h	PIDECFG	Power-On IDE Configuration	R/W
45-47h		Reserved	
48-49h	PIDETC	Primary IDE Timing Control	R/W
4A-4Bh	SIDETC	Secondary IDE Timing Control	R/W
4Ch	IIIRR	IDE-ISA Interrupt Routing Register	R/W
4Dh	IDEICS	IDE Interrupt Configuration and Status Register	R/W
50h	PCIRR	IDE-PCI Interrupt Routing Register	R/W
51-FFh		Reserved	

3.1.2.1 VENID—Vendor ID

Register Offset: 00h
Default Value: 8086h
Access: Read Only
Size: 16 bits

This is a unique 16-bit value assigned to a vendor that, together with the Device ID, uniquely identifies each PCI device. Writes to this register have no effect.

Bits[15:0]: Vendor Identification
This is a 16-bit value assigned to Intel.

3.1.2.2 DEVID—Device ID

Register Offset: 02h
Default Value: 1222h
Access: Read Only
Size: 16 bits

This is a unique 16-bit value that is assigned to the PCI-IDE Interface function. Writes to this register have no effect.

Bits[15:0]: Device Identification
This value identifies the PCI-IDE Interface function.

3.1.2.3 PCICMD—PCI Command

Register Offset: 04h
Default Value: 0000h
Access: Read/Write
Size: 16 bits

This 16 bit register contains PCI control information.

Bits[15:10]: Reserved

Bit 9: Fast Back-to-Back Control
This control function can be used only by a PCI master and is therefore not implemented in the PPEC. The bit always reads "0" (disabled).

Bit 8: SERR# Enable
When this bit is set to 1, the PPEC asserts SERR# when it detects a parity error during an address phase, and when it detects an X.X PC Card in Mode 1, partially-buffered implementations. When this bit is set to 0, SERR# is not asserted for any reason. Reset sets the bit to 0.

Bit 7: Address/Data Stepping Control
This control function can be used only by a PCI master and is therefore not implemented in the PPEC. The bit always reads "0" (disabled).

Bit 6: Parity Error Enable
This bit controls PPEC response to PCI data parity errors. When this bit is set to 1, the PPEC activates PERR# when it detects a parity error during a data phase. When this bit is set to 0, the PPEC ignores parity errors. Reset sets the bit to 0 and disables data parity checking.

Bit 5: VGA Palette Snoop
This bit is intended only for specific control of PCI-based VGA devices, and is not applicable to the PPEC. The bit is not implemented, and always reads "0".

Bit 4: Memory Write and Invalidate Enable
This control function can be used only by the PCI master, and is therefore not implemented in the PPEC. The bit always reads "0" (disabled).

Bit 3: Special Cycle Enable
This bit is intended to enable response to supported special cycles. Since the PPEC does not respond to any special cycle, the bit is not implemented and always reads "0".

Bit 2: Bus Master Enable
This bit is intended to enable mastership of the PCI. Since the PPEC cannot be a PCI master, the bit is not implemented and always reads "0".

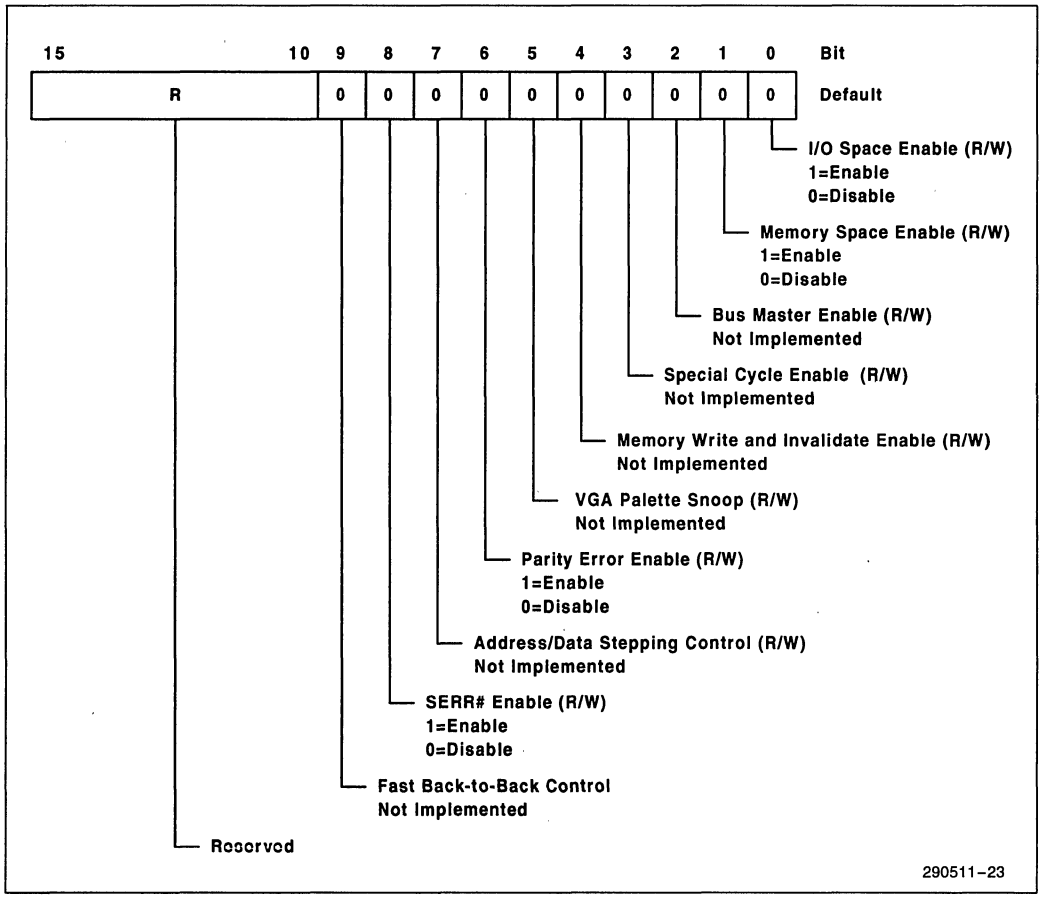


Figure 11. PCI Command Register

5

Bit 1: Memory Space Enable
This bit is intended to enable acceptance of PCI-originated memory cycles. Since the PCI-IDE Interface does not use memory cycles, the bit is not implemented and always reads "0".

Bit 0: I/O Space Enable
This bit enables the PCI-IDE Interface to accept PCI-originated I/O cycles. When the bit is set to 0, the interface does not respond to PCI master I/O cycles, and the PPEC's PCI-IDE DEVSEL# logic is inhibited during the I/O cycles.

This 16 bit register is used to record status information for PCI bus-related events. Reads to this register behave normally. Writing bits 11, 14, and 15 to 1 set the bits to 0. The other register bits cannot be written.

Bit 15: Detected Parity Error Status
The PPEC sets this bit to 1 when, as a target, it detects a parity error during a data phase, even if parity error handling is disabled by bit 6 and bit 8 in the PCI Command Register.

3.1.2.4 PCISTS—PCI Status

Register Offset: 06h
Default Value: 0280h
Access: Read Only (see register description)
Size: 16 bits

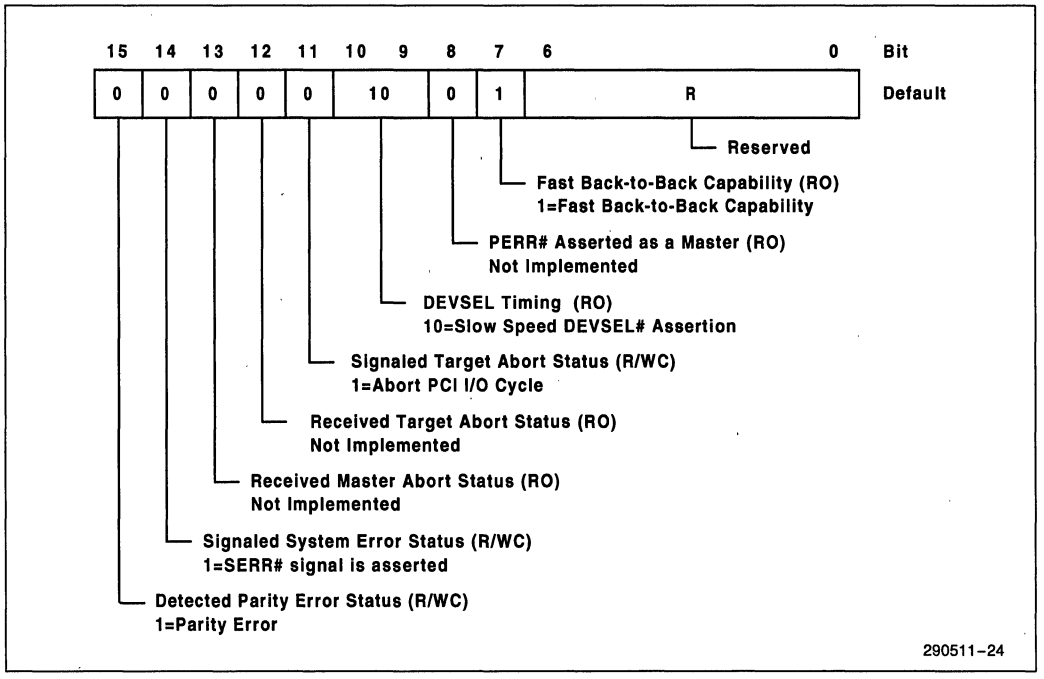


Figure 12. PCI Status Register

Bit 14: Signaled System Error Status
The PPEC sets this bit to 1 whenever it signals an address phase parity error by asserting SERR#.

Bit 13: Received Master Abort Status
This control function can be used only by a PCI master, and is therefore not implemented in the PPEC. The bit always reads "0" (disabled).

Bit 12: Received Target Abort Status
This control function can be used only by a PCI master, and is therefore not implemented in the PPEC. The bit always reads "0" (disabled).

Bit 11: Signaled Target Abort Status
The PPEC sets this bit to 1 when it is the target of a PCI I/O cycle, and the address/byte-enable combination is invalid.

Bits[10:9]: DEVSEL Timing
These read-only bits identify the slowest DEVSEL# response time for all bus commands except *configuration read* and *configuration write*, as defined in the PCI Specification. The PPEC implements medium speed DEVSEL# timing for PCI-IDE functions, and the bits are therefore 10b.

Bit 8: PERR# Asserted as a Master
This control function can be used only by a PCI master and is therefore not implemented in the PPEC. The bit always reads "0" (disabled).

Bit 7: Fast Back-to-Back Capability
This read-only bit is set to 1 to indicate that the PPEC can support fast back-to-back cycles originated by a PCI master.

Bits[6:0]: Reserved

3.1.2.5 REVID—Revision ID

Register Offset: 08h
Default Value: 01h
Access: Read Only
Size: 8 bits

This 8-bit register contains device revision information. Writes to this register have no effect.

Bits[7:0]: Revision Identification
This is the revision level of the PPEC. The initial PPEC revision level is 01h.

3.1.2.6 CCPIB—Class Code-Programming Interface Byte

Register Offset: 09h
Default Value: 00h
Access: Read Only
Size: 8 bits

This 8-bit register contains device Programming Interface information related to the Class Code register located at 0Ah offset. Writes to this register have no effect.

Bits[7:0]: Programming Interface
There are no specific register-level programming interfaces defined for this Class Code (indicated by register CCCB). Therefore, the value of this field is 0.

3.1.2.7 CCCB—Class Code-Class Code Bytes

Register Offset: 0Ah
Default Value: 0101h
Access: Read Only
Size: 16 bits

This 16-bit register contains device Class Code information in the following format: [BASE CLASS][SUB-CLASS]. Writes to this register have no effect.

Bits[15:8]: Base Class
The value 01h in this field identifies the function class as a *mass storage controller*.

Bits[7:0]: Sub-Class
The value 01h in this field identifies the function subclass as an *IDE Controller*.

3.1.2.8 HTYPE—Header Type

Register Offset: 0Eh
Default Value: 80h
Access: Read Only
Size: 8 bits

This register indicates whether or not the device contains multiple functions, and identifies the layout of bytes 10h through 3Fh in configuration space. Bit 7 indicates s multifunctional device when set to 1. Bits[6:0] specify layout of bytes 10h-3Fh. The PPEC uses layout type #0 as defined in the PCI specification. Writes to this register have no effect.

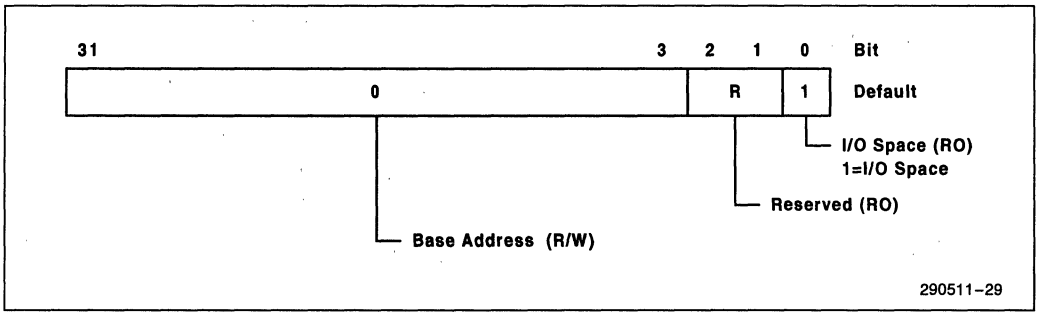


Figure 13. IDE Base Address Register 0

Bit 7: Multifunction Indicator
This bit is set to 1 to indicate that the PPEC is a multifunctional device.

Bits[6:0]: Byte Layout
This field specifies layout type "0" for bytes 10-3Fh, as defined in the PCI Specification.

3.1.2.9 PDCBA—IDE Base Address #0-Primary IDE Data/Command Address Range

Register Offset: 10h
Default value: 0000 0001h
Access: Read/Write
Size: 32 bits

This register determines the starting address of the primary IDE I/O address range for the Data/Command register block. It can be mapped anywhere in 4 GByte space on an 8-byte boundary.

The address range is defined as follows:

BaseAddress ≤ address ≤ BaseAddress + 7.

Bits[31:3]: Base Address
This field holds the programmable base address of the primary IDE I/O address range for the Data/Command register block.

Bits[2:1]: Reserved

Bit 0: I/O Space
This read only bit is set to 1 to indicate I/O space.

NOTE:
The PCI-compliant Base Address mechanism can be used for motherboard PPEC/IDE applications with PCI-customized BIOS. For applications that must use IDE functions in an ISA-compatible manner (e.g. PPEC add-in card with system BIOS that does not support PPEC-IDE), the address ranges defined with the four IDE BASE registers can be disabled by selecting the IDE hardware configuration feature in the IDE Power-On Configuration register. This feature allows configuration of IDE addresses for the compatible ranges, and selection of enhanced IDE timing mode. The compatible ranges are:

- Primary Data/Command Ports: 1F0-1F7h
- Primary Control/Status Ports: 3F6h
- Secondary Data/Command Ports: 170-177h
- Secondary Control/Status Ports: 376h

This PPEC feature eliminates the need for custom software for the IDE function, and applies to all four IDE Base Address Registers.

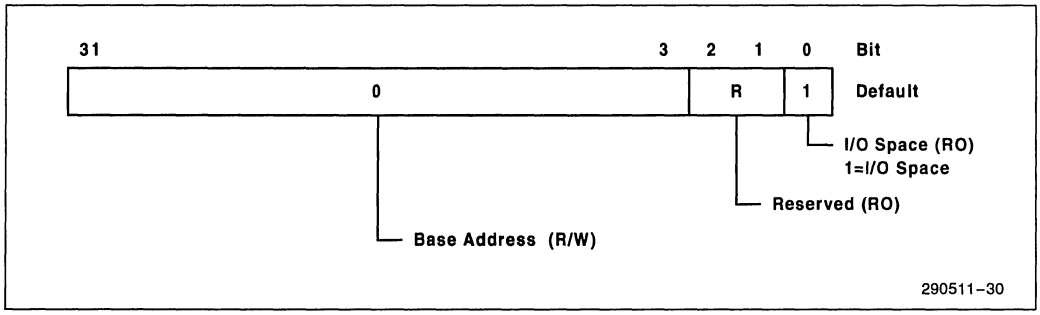


Figure 14. IDE Base Address Register 1

3.1.2.10 PCSBA—IDE Base Address #1-Primary IDE Control/Status Address Range

Register Offset: 14h
Default value: 0000 0001h
Access: Read/Write
Size: 32 bits

This register determines the starting address of the primary IDE I/O address range for the Control/Status register block. It can be mapped anywhere in 4 GByte I/O space on an 8-byte boundary.

The Primary address range is defined as follows:

BaseAddress ≤ address ≤ BaseAddress + 7.

Bits[31:3]: Base Address
This field holds the programmable base address of the primary IDE I/O address range for the Control/Status register block. These bits are read/write which indicates that the size of the required I/O address range is 8 bytes. In practical applications, only a subset of this range (1 byte) is used to access IDE registers.

Bits[2:1]: Reserved

Bit 0: I/O Space
This read only bit is set to 1 to indicate I/O space.

NOTE:
When accessing Control/Status Registers, proper offset must be used – 6h to access address xxx6h (Alternate Status Register).

3.1.2.11 SDCBA—IDE Base Address #2-Secondary IDE Data/Command Address Range

Register Offset: 18h
Default value: 0000 0001h
Access: Read/Write
Size: 32 bits

This register determines the starting address of the secondary IDE I/O address range for the Data/Command register block. It can be mapped anywhere in 4 GByte space on an 8-byte boundary.

5

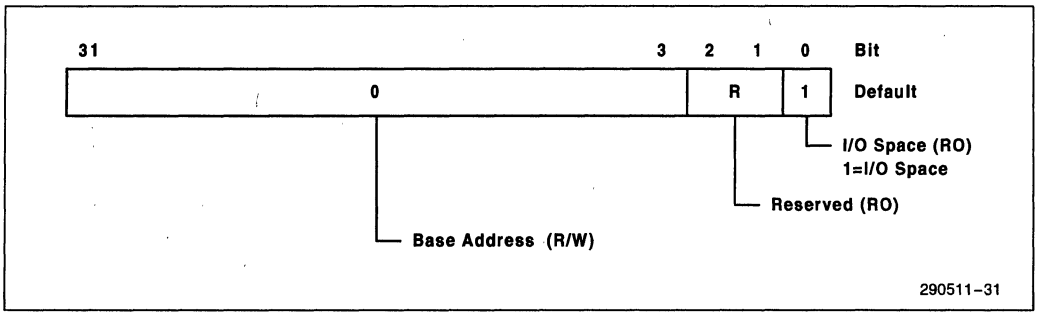


Figure 15. IDE Base Address Register 2

The address range is defined as follows:

BaseAddress ≤ address ≤ BaseAddress + 7.

Bits[31:3]: Base Address
This field holds the programmable base address of the secondary IDE I/O address range for the Data/ Command register block.

Bits[2:1]: Reserved

Bit 0: I/O Space
This read only bit is set to 1 to indicate I/O space.

3.1.2.12 SCSBA—IDE Base Address #3- Secondary IDE Control/Status Address Range

Register Offset: 1Ch
Default value: 0000 0001h
Access: Read/Write
Size: 32 bits

This register determines the starting address of the secondary IDE I/O address range for the Control/ Status register block. It can be mapped anywhere in 4 GByte I/O space on an 8-byte boundary.

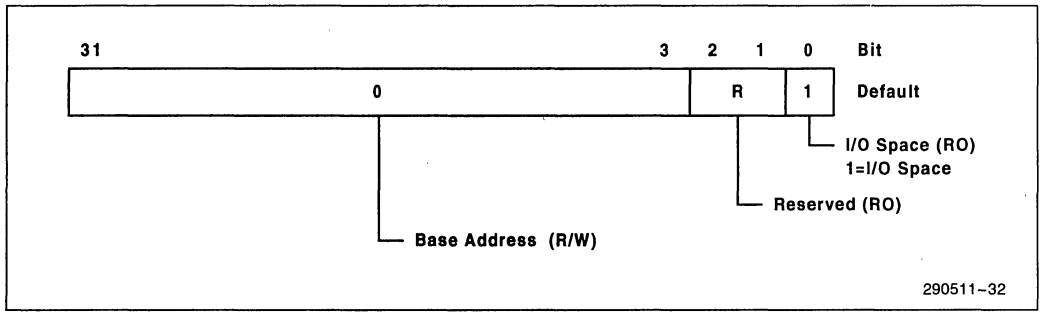


Figure 16. IDE Base Address Register 3

Secondary IDE Control/Status address range is defined as follows:

BaseAddress ≤ address ≤ BaseAddress + 7.

Bits[31:3]: Base Address
This field holds the programmable base address of the secondary IDE I/O address range for the Control/Status register block. These bits are read/write, which indicates that the size of the required I/O address range is 8 bytes. In practical applications, only a subset of this range (2 bytes) is used to access IDE registers.

Bits[2:1]: Reserved

Bit 0: I/O Space
This read only bit is set to 1 to indicate I/O space.

NOTE:
When accessing Control/Status Registers, proper offset must be used–6h to access address xxx6h (Alternate Status Register).

3.1.2.13 INTLIN—Interrupt Line

Register Offset:	3Ch
Default Value:	00h
Access:	Read/Write
Size:	8 bits

This register is used to communicate interrupt line routing information. BIOS software must initialize this register during system configuration. The value in this register identifies the interrupt request level of the system interrupt controller(s) to which the PPEC interrupt pin is connected. Device drivers and the operating system can use this information to determine priority and vector information. The value in this register is system architecture specific.

Bits[7:0]: Interrupt Line Identification
The value in this field identifies the interrupt request level of the system interrupt controller(s) to which the PPEC interrupt pin is connected.

NOTE:
The IDE Interface can connect to system interrupts in two different ways:

a. via a single PCI interrupt signal line (INTB#) that requires additional routing (system specific).

b. via 10 direct system interrupt signals (ISA mechanism).

Mode (b) is provided so that 'ISA compatible' IDE BIOS software can be used without modifications on X86/Pentium™-based platforms with the PPEC as a PCI-IDE Interface Controller. In this case, PCI interrupts are not used. IDE interrupts are configured to connect directly to a specific system IRQ line as specified in the IDE-ISA Interrupt Routing Register (IIIRR). The PCI Interrupt Line Register (INTLN) and the IDE-PCI Interrupt Routing Register (PCIRR) must remain in default state 0. Designs that do not require this software compatibility can use the PCI interrupt mechanism (a).

3.1.2.14 INTPIN—Interrupt Pin

Register Offset: 3Dh
Default Value: 02h
Access: Read Only
Size: 8 bits

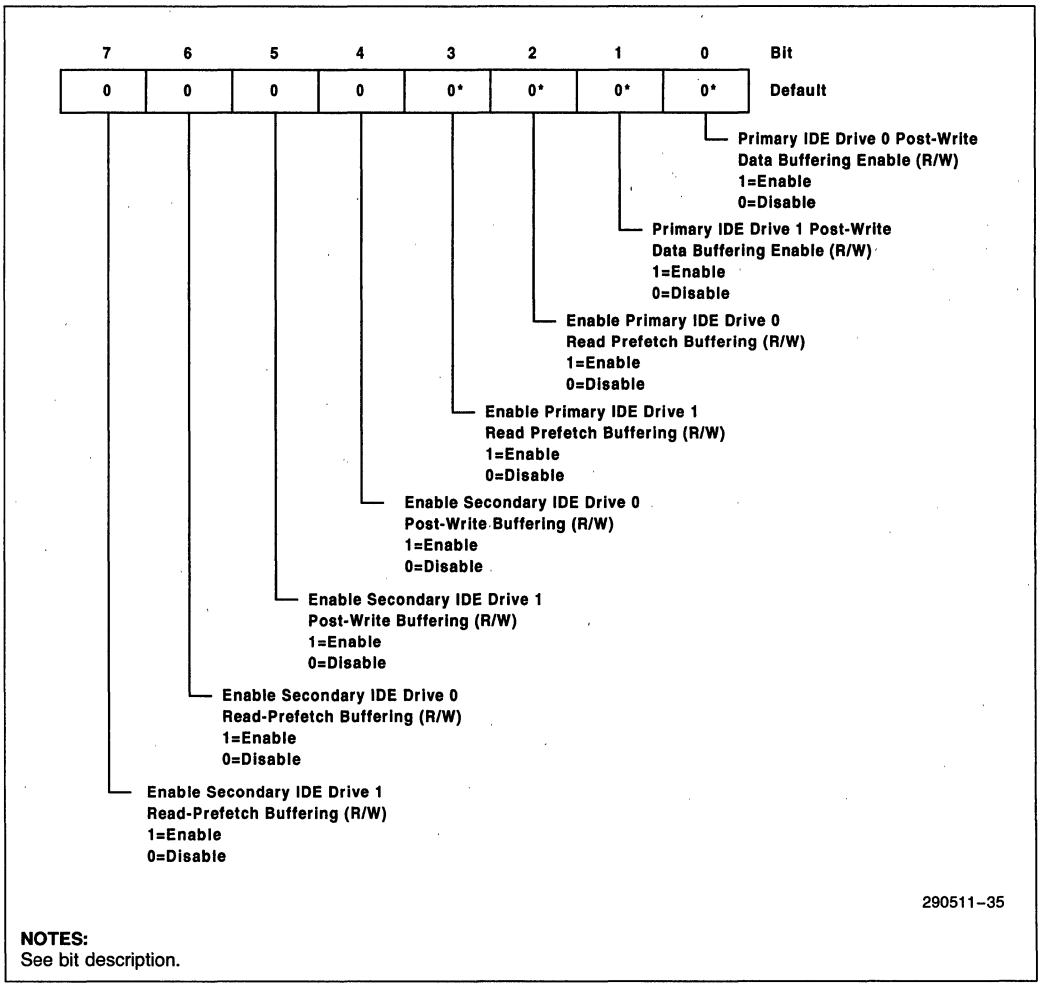
The value in this register, 02h, identifies the interrupt pin used by the PCI-IDE Interface for signaling IDE interrupts (Primary and/or Secondary IRQs) as INTB#. The IDE-PCI Interrupt Routing Register (PCIRR, located at offset 50h) is used to enable (for each IDE interface) interrupt signaling using the PCI interrupt scheme.

Bits[7:0]: Interrupt Pin Selection
The value in this field, 02h, identifies the PCI interrupt pin that is used by the PCI-IDE Interface device function as INTB#.

3.1.2.15 PCICON—PCI Configuration Control

Register Offset: 40h
Default Value: 00h
Access: Read/Write
Size: 8 bits

This register provides control of PCI-IDE data buffering.



NOTES:
See bit description.

Figure 17. PCI Configuration Control Register

Bit 7: Enable Secondary IDE Drive 1 Read-Prefetch Buffering
When this bit is set to 1, PCI to secondary IDE Drive 1 data buffering is enabled for Read Prefetch operations. When the bit is set to 0, the buffering is disabled.

Bit 6: Enable Secondary IDE Drive 0 Read-Prefetch Buffering
When this bit is set to 1, PCI to secondary IDE Drive 0 data buffering is enabled for Read Prefetch operations. When the bit is set to 0, the buffering is disabled.

Bit 5: Enable Secondary IDE Drive 1 Post-Write Buffering
When this bit is set to 1, PCI to secondary IDE Drive 1 data buffering is enabled for Post-Write operations. When the bit is set to 0, the buffering is disabled.

Bit 4: Enable Secondary IDE Drive 0 Post-Write Buffering
When this bit is set to 1, PCI to secondary IDE Drive 0 data buffering is enabled for Post-Write operations. When the bit is set to 0, the buffering is disabled.

Bit 3: Enable Primary IDE Drive 1 Read Prefetch Buffering
When this bit is set to 1, PCI to secondary IDE Drive 1 data buffering is enabled for Read Prefetch operations. When the bit is set to 0, the buffering is disabled. The default value of this bit is dependent on the IDE Hardware Configuration feature. If IDE Hardware Configuration is enabled (bit 0 of the PIDECFG register is set to 1), the bit defaults to the value of bit 7 of the PIDECFG register. If IDE Hardware Configuration is not enabled, the bit defaults to 0.

Bit 2: Enable Primary IDE Drive 0 Read Prefetch Buffering
When this bit is set to 1, PCI to primary IDE Drive 0 data buffering is enabled for Read Prefetch operations. When the bit is set to 0, the buffering is disabled. The default value of this bit is dependent on the IDE Hardware Configuration feature. If IDE Hardware Configuration is enabled (bit 0 of the PIDECFG register is set to 1), the bit defaults to the value of bit 7 of the PIDECFG register. If IDE Hardware Configuration is not enabled, the bit defaults to 0.

Bit 1: Enable Primary IDE Drive 1 Post-Write Buffering
When this bit is 1, PCI to secondary IDE Drive 1 Post-Write data buffering is enabled. When this bit is 0, buffering is disabled. The default value of this bit is dependent on the IDE Hardware Configuration feature. If IDE Hardware Configuration is enabled (bit 0 of the PIDECFG register is set to 1), the bit defaults to the value of bit 7 of the PIDECFG register. If IDE Hardware Configuration is not enabled, the bit defaults to 0.

Bit 0: Primary IDE Drive 0 Post-Write Buffering
When this bit is 1, PCI to primary IDE Drive 0 Post-Write data buffering is enabled. When this bit is 0, buffering is disabled. The default value of this bit is dependent on the IDE Hardware Configuration feature. If IDE Hardware Configuration is enabled (bit 0 of the PIDECFG register is set to 1), the bit defaults to the value of bit 7 of the PIDECFG register. If IDE Hardware Configuration is not enabled, the bit defaults to 0.

3.1.2.16 PIDECFG—Power-On IDE Configuration

Register Offset:	44h
Default Value:	xxh
Access:	Read/Write
Size:	8 bits

This register reports the status of the IDE Hardware Configuration signals that are multiplexed on PCMCIA Socket A data lines ACDATA[7:0] when the PPEC operates in Mode 0 (2-socket mode), and on common data lines CDATA[7:0] when the PPEC operates in Mode 1 (4-socket mode). The status of these signals is latched in this register during reset if this feature is enabled by the global IDE Hardware Configuration enable pin. These bits control the preload default value in the IDE timing control registers.

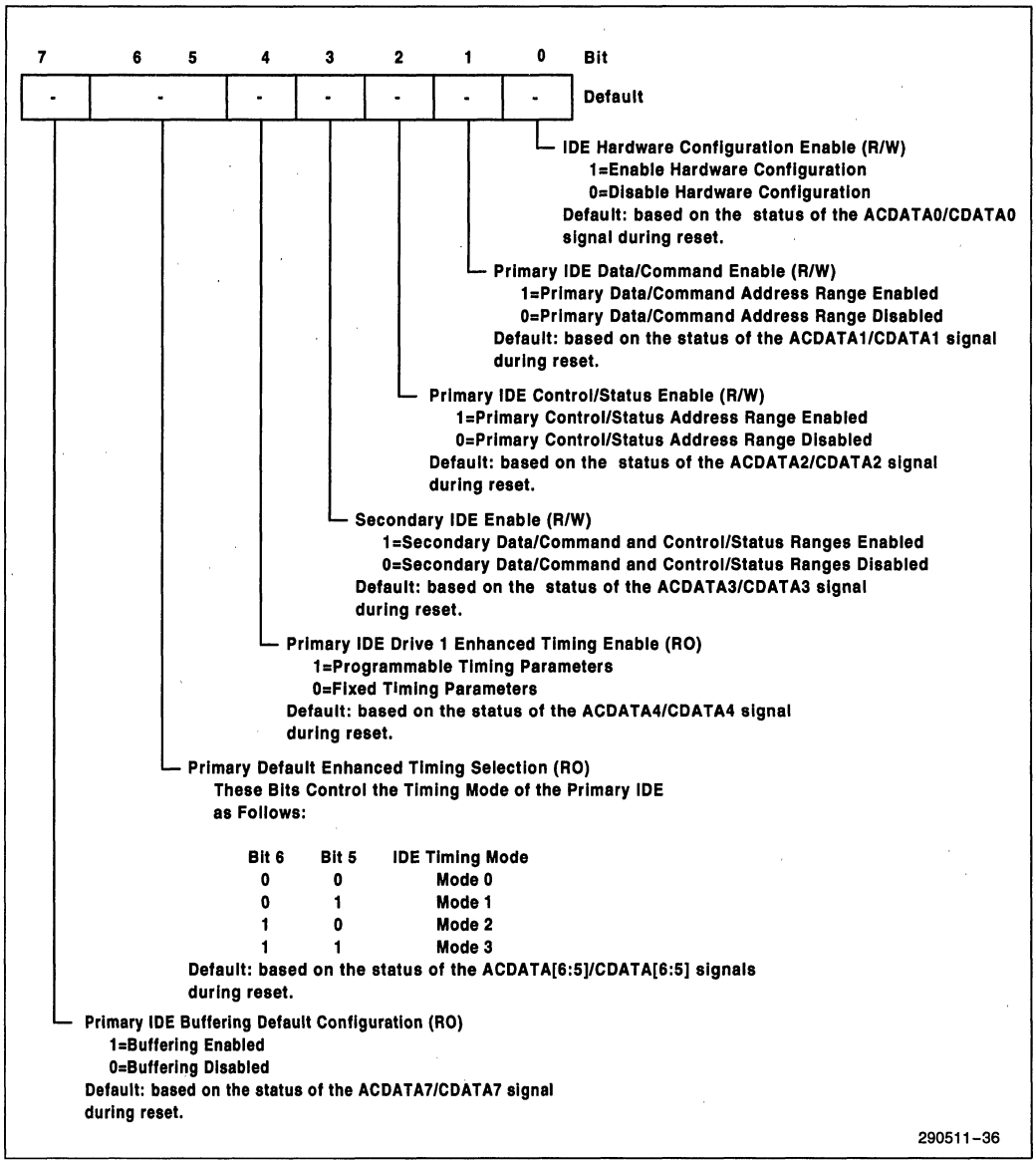


Figure 18. Power-On IDE Configuration Control Register

Bit 7: Primary IDE Buffering Default Configuration
This bit enables IDE data port buffering when set to 1 at the end of the reset sequence if ACDATA0 is externally pulled high, and disables IDE data port buffering when set to 0. If ACDATA0 is externally pulled low at the end of the reset sequence, this bit is not used to configure the IDE interface. After reset, this bit indicates the state of the ACDATA7/ CDATA7 signal at the end of the reset sequence.

Bits[6:5]: Primary Default Enhanced Timing Selection
These bits control the timing mode of the primary IDE interface at the end of the reset sequence if ACDATA0 is externally pulled high, as follows:

Bit 6	Bit 5	IDE Mode Timing
0	0	Mode 0
0	1	Mode 1
1	0	Mode 2
1	1	Mode 3

If ACDATA0 is externally pulled low at the end of the reset sequence, these bits are not used to configure the IDE interface. After reset, these bits indicate the states of the ACDATA[6:5]/CDATA[6:5] signals at the end of the reset sequence.

Note that these bits directly define the timing mode for Primary IDE drive 0, and that the same timing is applied to the Primary IDE drive 1 if bit 4 of this register is set to 1.

Bit 4: Primary Drive #1 Enhanced Timing Enable
When this bit is set to 1 at the end of the reset sequence with ACDATA0 externally pulled high, primary drive #1 data port timing is based on programmable timing parameters. When the bit is set to 0 at the end of the reset sequence, the timing is fixed, Mode 0 compatible timing. If ACDATA0 is externally pulled low at the end of the reset sequence, this bit is not used to configure the IDE interface. After reset, this bit indicates the state of the ACDATA4/ CDATA4 signal at the end of the reset sequence.

Bit 3: Secondary IDE Enable
When this bit is 0, all Secondary IDE registers are disabled regardless of whether they are selected from the preset compatible range (170-177h, 376h), or IDE Base Address #2 and #3. When this bit is 1, the Secondary IDE registers are enabled. The default value of this bit is 0 if IDE Hardware Configuration is disabled (bit 0 is sampled low). The default value is the value of ACDATA3 sampled at the end of the reset sequence if IDE Hardware Configuration is enabled (bit 0 is sampled high).

Bit 2: Primary IDE Control/Status Enable
When this bit is 0, the Primary IDE Control/Status register is disabled regardless of whether it is selected from the preset compatible range (3F6h), or IDE Base Address #1. When this bit is 1, the Primary IDE Data/ Command register is enabled. The default value of this bit is 0 if IDE Hardware Configuration is disabled (bit 0 is sampled low). The default value is the value of ACDATA2 sampled at the end of the reset sequence if IDE Hardware Configuration is enabled (bit 0 is sampled high).

Bit 1: Primary IDE Data/Command Enable
When this bit 0, the Primary IDE Data/Command registers are disabled regardless of whether they are selected from the preset compatible range (1F0-1F7h), or IDE Base Address #0. When this bit is 1, the Primary IDE Data/Command registers are enabled. This bit defaults to 0 if IDE Hardware Configuration is disabled (bit 0 is sampled low). The default value is the value of ACDATA1 sampled at the end of the reset sequence if IDE Hardware Configuration is enabled (bit 0 is sampled high).

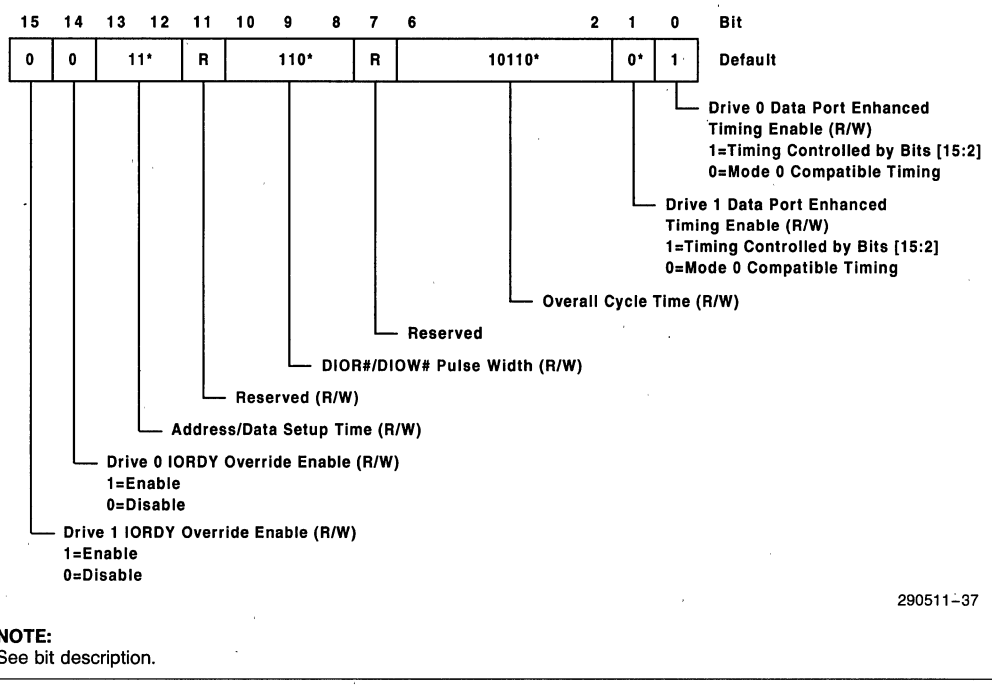
Bit 0: IDE Hardware Configuration Enable
The default value of this bit is determined by the value of the ACDATA0 signal at reset. If ACDATA0 is sampled high at the end of the reset sequence, the default values of hardware configuration bits[7:1] are determined by the values of ACDATA[7:1], the fixed IDE compatible ranges are selected (subject to enable bits[3:1]), and PCI-IDE space defaults to enabled. If ACDATA0 is sampled low, hardware configuration bits[7:1] default to 0, the IDE Base Address registers are selected (subject to enable bits[3:1]), and PCI-IDE I/O space defaults to disabled. A software write to this bit selects between IDE compatible ranges (1) and IDE Base Address registers (0), but does not affect the PCI-IDE I/O space enable.

NOTE:
The Secondary IDE timing and Data Buffering control are not IDE Hardware configurable. They default to the slowest timing mode (Mode 0), and data buffering disabled.

3.1.2.17 PIDETC—Primary IDE Timing Control

Register Offset: 48h
Default value: xxxxh
Access: Read/Write
Size: 16 bits

This register determines the timing characteristics and IORDY control of the Primary IDE interface.



NOTE:
See bit description.

Figure 19. Primary IDE Timing Control Register

Bit 15: Drive #1 IORDY Override Enable
When this bit is set to 1, the external IORDY signal is overridden for Primary Drive 1 (i.e., the IORDY signal is asserted internally regardless of the state of the external IORDY signal). When this bit is 0, the external IORDY signal is used to control completion of primary IDE drive #1 accesses.

Bit 14: Drive #0 IORDY Override Enable
When this bit is set to 1, the external IORDY signal is overridden for Primary Drive 0 (i.e., the IORDY signal is asserted internally regardless of the state of the external IORDY signal). When this bit is 0, the external IORDY signal is used to control completion of primary IDE drive #0 accesses.

Bits[13:12]: Address/Data Setup Time
These bits define, in system clock (PCICLK) periods, the address/data setup time with respect to the write/read strobes. If IDE hardware configuration is enabled by bit 0 of the Power-On IDE Configuration Register (PIDECFG), the default value of the bits correspond to the IDE timing mode selected by bits[6:5] of the PIDECFG Register. If IDE hardware configuration is not enabled, the default is 11 = Mode 0.

Bit 11: Reserved

Bits[10:8]: DIOR#/DIOW# Pulse Width
These bits define the width of the Write and Read strobes in system clock (PCICLK) periods. If IDE hardware configuration is enabled by bit 0 of the Power-On IDE Configuration Register (PIDECFG), the default value of the bits correspond to the IDE timing mode selected by bits[6:5] of the PIDECFG Register. If IDE hardware configuration is not enabled, the default is 110 = Mode 0.

Bit 7: Reserved

Bits[6:2]: Overall Cycle Time
Defines the length of the IDE cycle in system clock (PCICLK) periods. If IDE hardware configuration is enabled by bit 0 of the Power-On IDE Configuration Register (PIDECFG), the default value of the bits correspond to the IDE timing mode selected by bits[6:5] of the PIDECFG Register. If IDE hardware is not enabled, the default is 10110 = Mode 0.

Bit 1: Drive #1 Data Port Enhanced Timing Enable
When this bit is set to 1, IDE Primary Drive 0 access timing is controlled by bits[15:2] of this register. Accesses to other ports is based on compatible timing as defined by Mode 0 in the ATA specification. If IDE hardware configuration is enabled by bit 0 of the Power-On IDE Configuration Register (PIDECFG), the default value of the bit is the value of bit 4 of the PIDECFG Register. If IDE hardware configuration is not enabled, the default is 0. When this bit is set to 0, accesses is based on compatible Mode #0 timing.

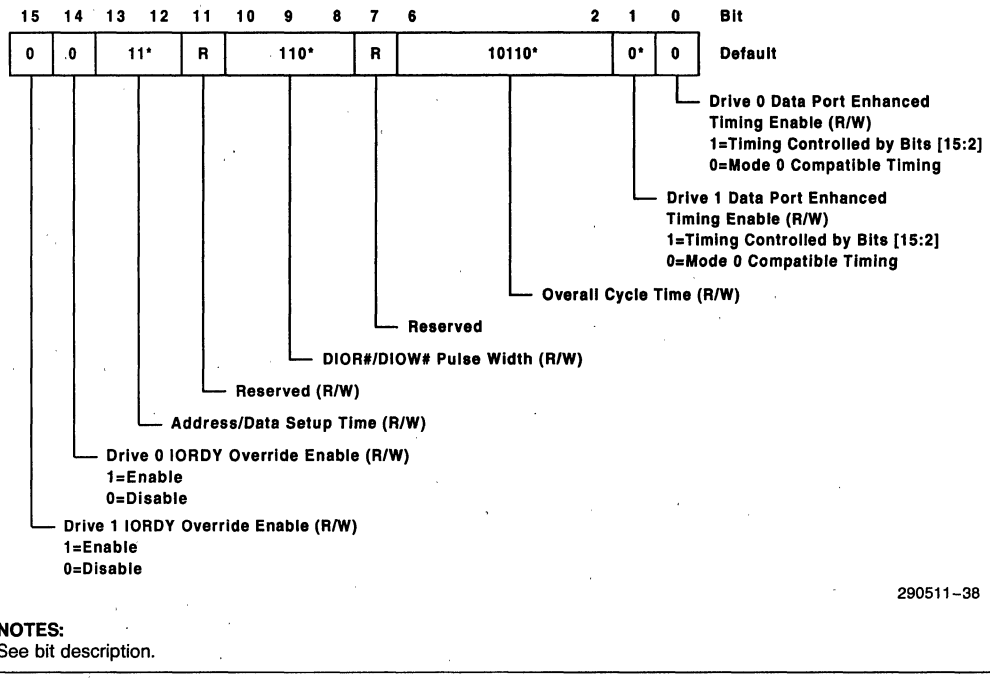
Bit 0: Drive #0 Data Port Enhanced Timing Enable
When this bit is set to 1, IDE Primary Drive 0 access timing is controlled by bits[15:2] of this register. Accesses to other ports is based on compatible timing as defined by Mode 0 in the ATA specification. When this bit is set to 0, accesses are based on compatible Mode #0 timing. The default value for this bit is 1.

5

3.1.2.18 SIDETC—Secondary IDE Timing Control

Register Offset:	4Ah
Default value:	3658h
Access:	Read/Write
Size:	16 bits

This register determines the timing characteristics and IORDY control of the Secondary IDE interface.



NOTES:
See bit description.

Figure 20. Secondary IDE Timing Control Register

Bits 15: Drive #1 IORDY Override Enable
When this bit is set to 1, the external IORDY signal is overridden for IDE Secondary Drive 1 (i.e., the IORDY signal is asserted internally regardless of the state of the external IORDY signal). When this bit is 0, the external IORDY signal is used to control completion of Secondary IDE drive #1 accesses.

Bit 14: Drive #0 IORDY Override Enable
When this bit is set to 1, the external IORDY signal is overridden for IDE Secondary Drive 0 (i.e., the IORDY signal is asserted internally regardless of the state of the external IORDY signal). When this bit is 0, the external IORDY signal is used to control completion of Secondary IDE drive #0 accesses.

Bits[13:12]: Address/Data Setup Time
These bits define, in system clock (PCICLK) periods, the address/data setup time with respect to the write/read strobes.

Bit 11: Reserved

Bits[10:8]: DIOR#/DIOW# Pulse Width.
These bits define the width of the Write and Read strobes in system clock (PCICLK) periods.

Bit 7: Reserved

Bits[6:2]: Overall Cycle Time
These bits define the length of the IDE cycle in system clock (PCICLK) periods.

Bit 1: Drive #1 Data Port Enhanced Timing Enable
When this bit is set to 1, IDE Secondary Drive 1 data port access timing is controlled by bits[15:2] of this register. Accesses to other ports are based on compatible timing as defined by Mode 0 in the ATA specification. When this bit is set to 0, accesses to IDE ports when Drive 0 is active is based on compatible Mode #0 timing.

Bit 0: Drive #0 Data Port Enhanced Timing Enable
When this bit is set to 1, IDE Secondary Drive 0 data port access timing is controlled by bits[15:2] of this register. Accesses to other ports are based on compatible timing as defined by Mode 0 in the ATA specification. When this bit is set to 0, accesses to IDE ports when Drive 0 is active is based on compatible Mode #0 timing.

NOTE:
The Secondary IDE timing mode is not IDE Hardware configurable. It defaults to Mode 0, the slowest timing mode.

3.1.2.19 IIIRR—IDE-ISA Interrupt Routing Register

Register Offset: 4Ch
Default Value: XXh
Access: Read/Write
Size: 8 bits

This register selects mapping of the Primary and Secondary IDE Interface interrupt requests to any of 10 ISA-compatible system interrupts.

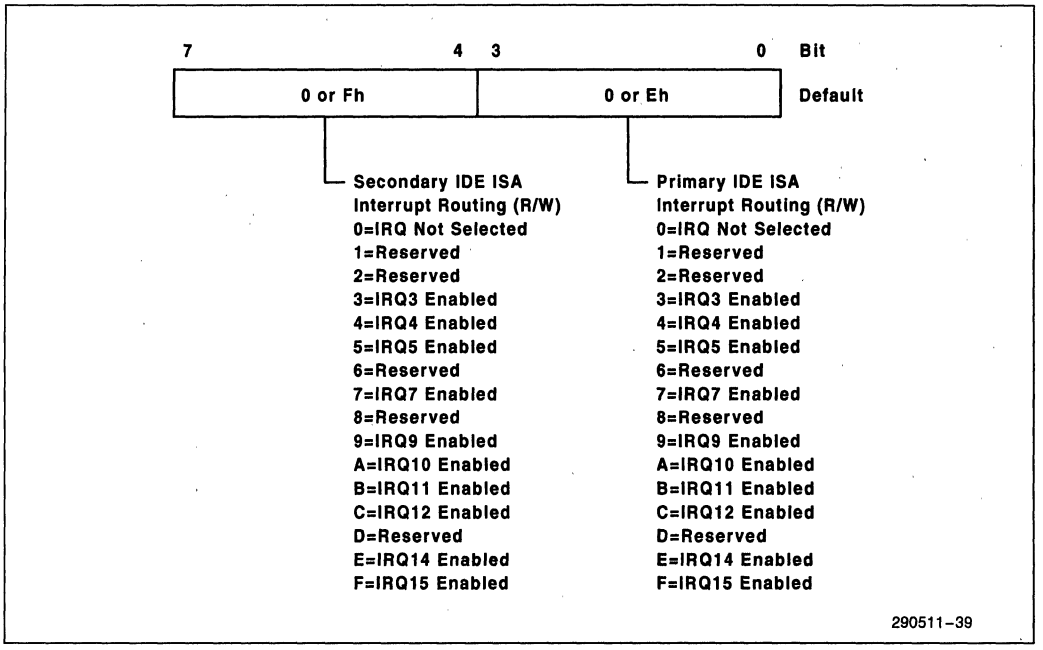


Figure 21. IDE-ISA Interrupt Routing Register

Bits[7:4]: Secondary IDE ISA Interrupt Mapping
This field selects the IRQ level for the Secondary IDE Interface. The default for this field is 0 if IDE Hardware Configuration is disabled (bit 0=0) in the Power-On IDE Configuration Control Register (PIDECFG). The default is Fh if IDE Hardware Configuration is enabled (bit 0=1) and the Secondary address range is enabled.

Bits[3:0]: Primary IDE ISA Interrupt Mapping
This field selects the IRQ level for the Primary IDE Interface. The default for this field is 0 if IDE Hardware Configuration is disabled (bit 0=0) in the Power-On IDE Configuration Control Register (PIDECFG). The default is Eh if IDE Hardware Configuration is enabled (bit 0=1) and the Primary Data/Command address range is enabled.

3.1.2.20 IDEICS—IDE Interrupt Configuration/ Status Register

Register Offset: 4Dh
Default Value: 00h
Access: Read/Write
Size: 8 bits

This register allows selection of edge or level mode for Primary and Secondary interrupts selected via the IIIRR Register, and provides non-latched read-only status of the physical PIRQ and SIRQ signal pins.

Bits[7:4]: Reserved

Bit 3: Secondary IDE System Interrupt Operation Mode
This bit provides the IRQ operation mode for the system interrupt signal selected to be used as a Secondary IDE Interrupt via the IIIRR Register as follows:

1=Level Mode Selected
0=Edge Mode Selected

Bit 2: Primary IDE System Interrupt Operation Mode
This bit provides the IRQ operation mode for the system interrupt signal selected to be used as a Primary IDE Interrupt via the IIIRR Register as follows:

1=Level Mode Selected
0=Edge Mode Selected

Bit 1: Secondary IDE Interrupt Status
This bit provides the status of the SIRQ Secondary IRQ signal as follows:

1=Secondary IDE IRQ active
0=Secondary IDE IRQ inactive

Bit 0: Primary IDE Interrupt Status
This bit provides the status of the SIRQ Primary IRQ signal as follows:

1=Primary IDE IRQ active
0=Primary IDE IRQ inactive

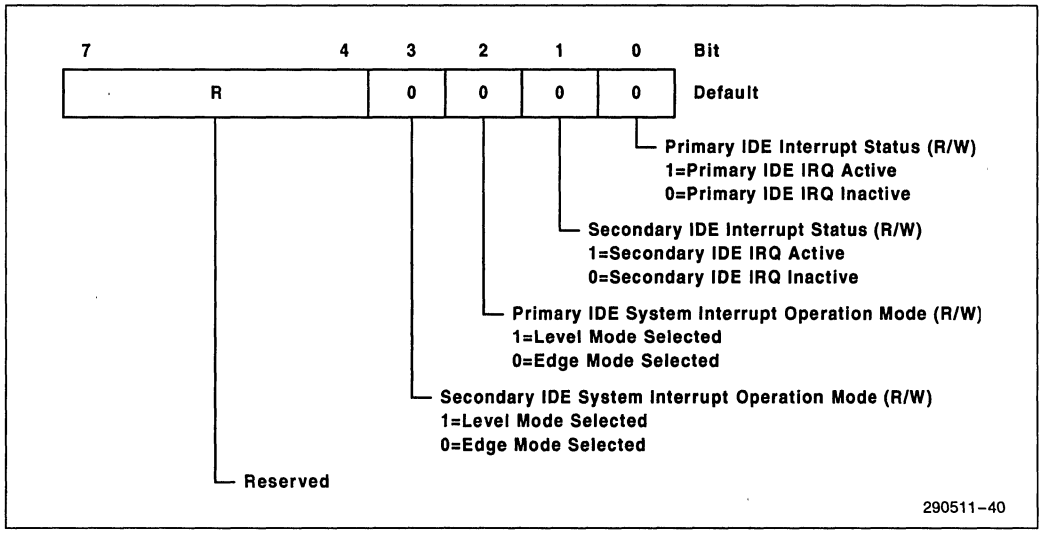


Figure 22. IDE Interrupt Configuration/Status Register

3.1.2.21 PCIRR—IDE-PCI Interrupt Routing Register

Register Offset: 50h
Default Value: 00h
Access: Read/Write
Size: 8 bits

This register allows mapping of IDE interrupts to either ISA interrupts, or PCI interrupts. One interrupt can be generated for the primary IDE interface, and another for the secondary IDE interface. Each of the interrupts can be routed independently. When bit 0 or 1 is set to 0, the corresponding interrupt is routed via one of 10 system interrupt lines (ISA mechanism) selected by the IDE ISA Interrupt Mapping Register. When set to 1, the corresponding interrupt is routed via INTB# (PCI mechanism).

Bits[7:2]: Reserved

Bit 1: Secondary IDE Interrupt Routing
This bit selects Secondary IDE Interface Interrupt routing via the PCI mechanism, or the ISA mechanism.

Bit 0: Primary IDE Interrupt Routing
This bit selects Primary IDE Interface Interrupt routing via the PCI mechanism, or the ISA mechanism.

3.2 PCMCIA Socket Configuration Registers

The PPEC has four identical sets of registers for controlling the four PCMCIA sockets, with each set controlling one socket. Each register set is comprised of four types of registers: General Setup Registers, Interrupt Registers, I/O Mapping Control Registers, and Memory Mapping Control Registers. One set of registers is described in the following sections, with the address offset for each socket shown in each description.

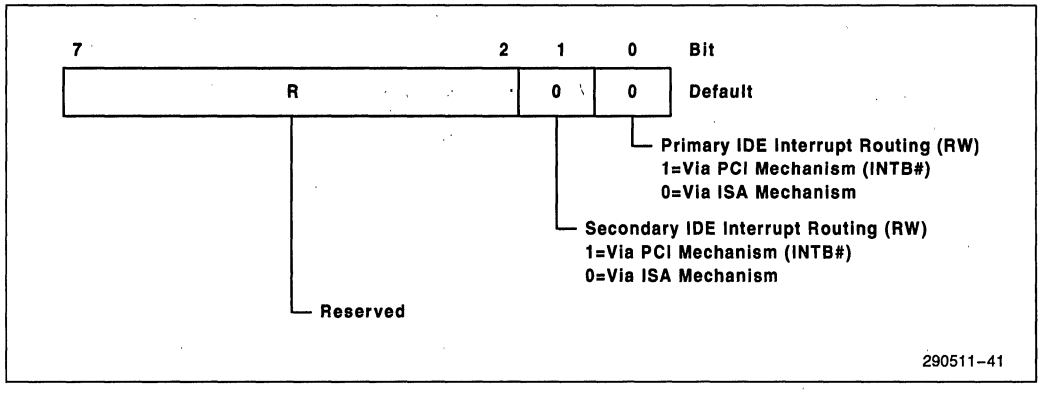


Figure 23. IDE Interrupt Routing Register

3.2.1 GENERAL SETUP REGISTERS

The General Setup Registers, listed in Table 8, are 8-bit registers. Writes to Read Only General Setup registers and register bits have no effect.

Table 8. General Setup Registers

Register Offset Sockets A:D				Mnemonic	Register Name	Access
A	B	C	D			
00	40	80	C0	IDREG	Identification	RO
01	41	81	C1	ISTAT	Interface Status	RO
02	42	82	C2	PCTRL	Power Control	R/W
04	44	84	C4	CSTCH	Card Status Change	R/W
06	46	86	C6	ADWEN	Address Window Enable	R/W
1E	5E	9E	DE	GCTRL	Global Control	R/W
2E	6E	AE	EE	CSCTRL	Global Security Control	R/W
16	56	96	D6	CDGEN	Card Detect and General Control	R/W
26	66	A6	E6	CPAGE	Card Memory Page	R/W

3.2.1.1 IDREG—Identification Register

Register Offset: Socket A—00h
Socket B—40h
Socket C—80h
Socket D—C0h
Default value: 84h
Access: Read/Write
Size: 8 bits

The Identification Register is used by the system software to determine the type of PC Cards supported by the socket.

Bits[7:6]: Interface Type
These bits indicate the type of PC Cards supported by the PPEC at the particular socket as follows:

00 = I/O Only
01 = Memory Only
10 = Memory and I/O
11 = Reserved

These bits do not identify the type of card that is present at the socket.

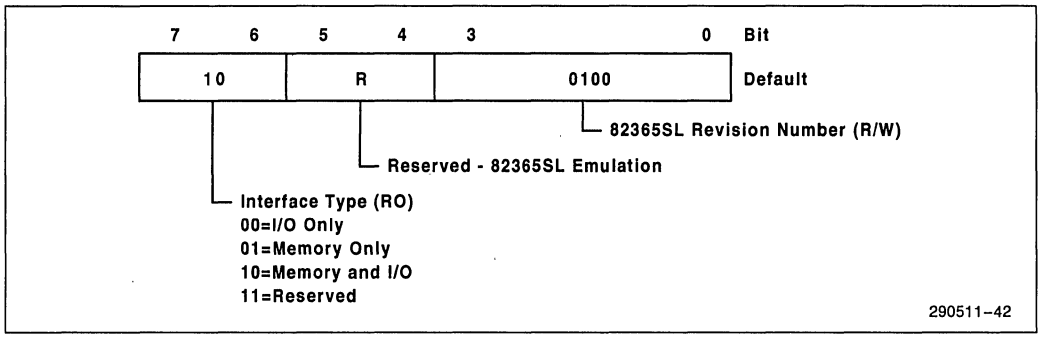


Figure 24. Identification Register

Bits[5:4]: Reserved
These are read/write bits that can be used to store 82365SL-specific information to allow 82365SL emulation.

Bits[3:0]: Reserved—82365SL Revision Information
This read/write field holds 82365SL revision information that allows the PPEC to emulate the 82365SL. Software checks this field before executing code written for the 82365SL.

3.2.1.2 ISTAT—Interface Status Register

Register Offset: Socket A—01h
Socket B—41h
Socket C—81h
Socket D—C1h
Default value: XX
Access: Read-only
Size: 8 bits

The Interface Status Register provides the current status of the PC Card socket interface signals.

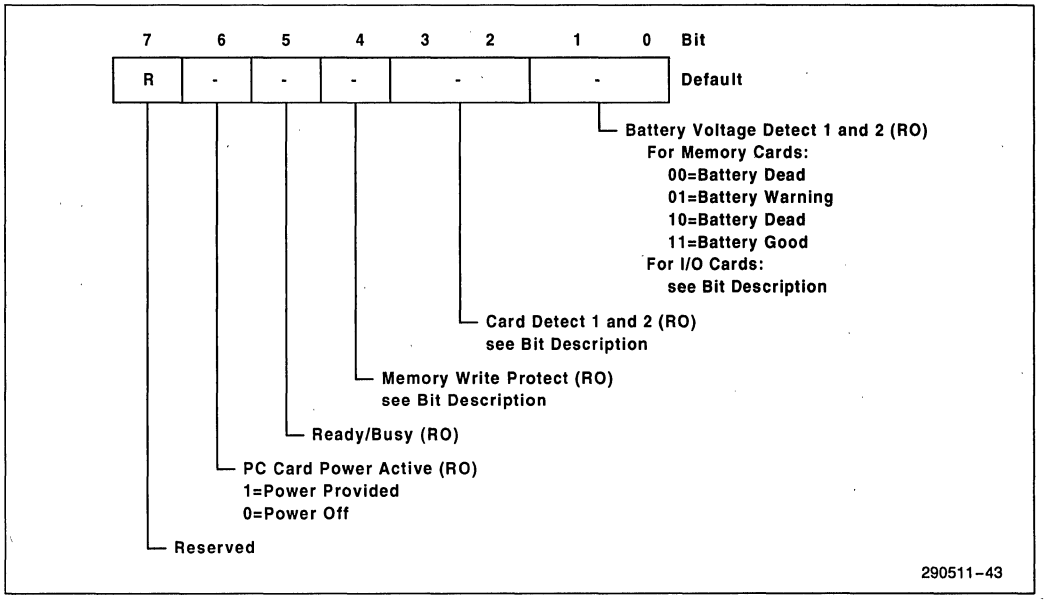


Figure 25. Interface Status Register

Bit 7: Reserved

Bit 6: PC Card Power Active
Indicates the current power status of the socket. If this bit is set to zero, power to the socket is turned off (V_{CC} and V_{PP} are not applied). If the bit is set to one, power is applied to the socket (V_{CC} is applied according to bits[4:3] of the Power Control Register, and V_{PP} is applied according to bits[1:0] of the Power Control Register).

Bit 5: Ready/Busy#
This bit is set to 1 to indicate that the PC Card is ready to accept a data transfer, and set to 0 to indicate that the card is busy completing an operation and cannot accept new data or commands.

Bit 4: Memory Write Protect
This bit indicates the logic level of the WP signal on the memory PC Card interface. However, memory write access to the socket is blocked only if the write protect bit in the associated Card Memory Offset Address Register High byte register is set to one.

Bits[3:2]: Card Detect 1 and 2
These bits indicate, when both are set to 1, that a card is present at the socket and is fully seated. Bit 2 is set to 1 if the CD1 signal on the PC Card interface is active, and bit 3 is set to 1 if the CD2 signal is active. Bits 2 and 3 are set to 0 if the corresponding CD1 and CD2 signals on the PC Card interface are inactive.

Bits[1:0]: Battery Voltage Detect 1 and 2
For memory cards, these bits indicate the status of the battery as follows:

BVD1	BVD2	Status
0	0	Battery Dead
0	1	Battery Dead
1	0	Warning
1	1	Battery Good

For I/O PC Cards, bit 0 indicates the current status of the STSCHG signal from the PC Card. For I/O PC Cards, bit 1 indicates the current state of SPKR signal from the PC Card. Refer to the Interrupt General Control Register bit 7 description for more details.

3.2.1.3 PCTRL—Power Control Register

Register Offset: Socket A—02h
Socket B—42h
Socket C—82h
Socket D—C2h
Default value: xx
Access: Read/Write
Size: 8 bits

This register controls power to the PC card. PCIRST# (PCI reset) clears all bits in this register. Output Enable should not be set until the register has been previously written setting the socket V_{PP} and V_{CC} Power Control bits.

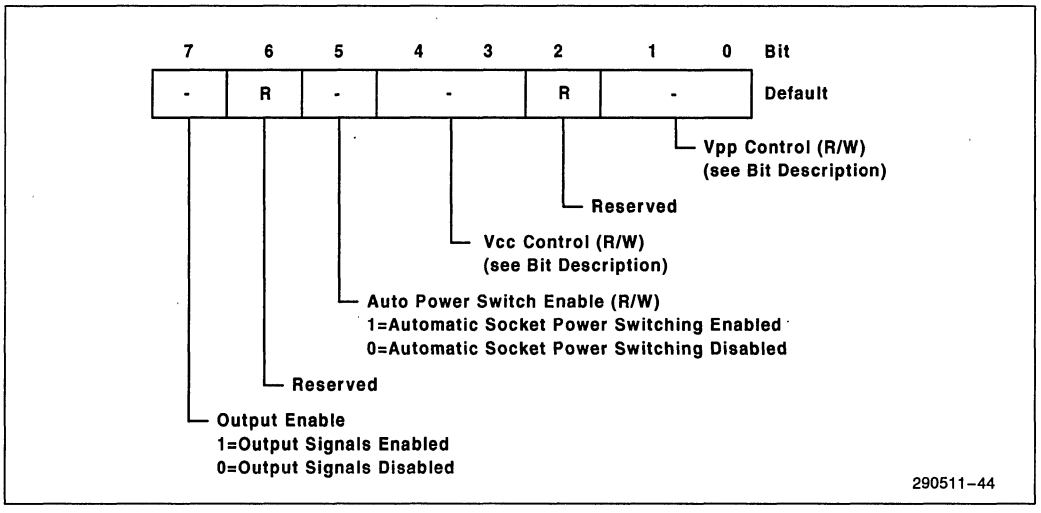


Figure 26. Power Control Register

5

Bit 7: Output Enable
When this bit is set to 0, the PPEC output signals that are directly connected to the socket (Mode 0) or are shared by other sockets (Mode 1) are tri-stated, and the ENABLE# signal to the socket is inactive. When the bit is set to 1, the signals are not tri-stated, and the ENABLE# signal is asserted. Output Enable should be set to 0 when the socket is not powered.

Bit 6: Reserved

Bit 5: Auto Power Switch Enable
When this bit is set to 0, automatic socket power switching based on card detects is disabled. When the bit is set to 1, automatic socket power switching is enabled. Automatic socket power switching function controls the V_{CC}xV and V_{PP}ENx power control bits. V_{CC} is 5V or 3.3V depending on the sampled states of the VS1 and VS2 signals provided in the Card Detect and General Control Register.

Bits[4:3]: V_{CC} Control
These bits control the power to the PC Card via the external V_{CC}-3V and V_{CC}-5V control logic (External Power Latch). The two bits are encoded as follows:

Bit 4	Bit 3	V_{CC}5V	V_{CC}3V	Description
0	0	0	0	No Connect
0	1	0	0	Reserved
1	0	1	0	5.0V
1	1	0	1	3.3V

Bit 2: Reserved

Bits[1:0]: V_{PP} Control
These bits switch V_{PP} power using the external Vpp control logic (External Power Latch). The two bits are encoded together with bit 4 to implement the following control functions:

Bit 4	Bit 1	Bit 0	V_{PP} EN1	V_{PP} EN0	Applied Voltage
1	0	0	0	0	No Connect
1	0	1	0	1	5.0V
1	1	0	1	0	12.0V
1	1	1	0	0	Reserved
0	x	x	0	0	No Connect

For more details on V_{CC}/V_{PP} control functions, see the Power Control description in Section 1.4.2 of this document.

3.2.1.4 CSTCH—Card Status Change Register

Register Offset:	Socket A—04h
	Socket B—44h
	Socket C—84h
	Socket D—C4h
Default value:	00h
Access:	Read/Write
Size:	8 bits

This register contains the status of the sources for the Card Status Change Interrupts. These sources can be enabled to generate a Card Status Change Interrupt by setting the corresponding bit in the Card Status Change Interrupt Configuration Register. The bits in this register read back as 0 when the corresponding status enable bits in the Card Status Change Interrupt Configuration Register are set to 0.

When the Explicit Write Back Card Status Change Acknowledge bit is set in the Global Control Register, the acknowledgment of sources for the Card Status Change Interrupt is performed by writing back 1 to the appropriate bit in the Card Status Change Register that was read as a 1. Once the interrupt source is acknowledged by writing a 1 to the bit, the bit reads back as 0. The interrupt signal responding to the card status change remains active, if enabled on a system IRQ line, until all of the bits in this register are zero.

When the Explicit Write Back Card Status Acknowledge bit is not set, the Card Status Change Interrupt remains active, if enabled on a system IRQ line, until the Card Status Change Register is read. The read operation to the Card Status Change Register resets all bits in the register.

If two or more Card Status Change Interrupts are pending or a Card Status Change Interrupt condition occurs while another is being serviced, the PPEC does not generate a second interrupt.

The Interrupt Service Routine must read the Card Status Change Register to ensure that all interrupt requests are serviced before exiting the service routines.

Asserted PCIRST# (PCI reset) clears all bits in this register.

In the following bit descriptions, bits names that are in parenthesis are valid when the interface is configured for I/O PC Cards.

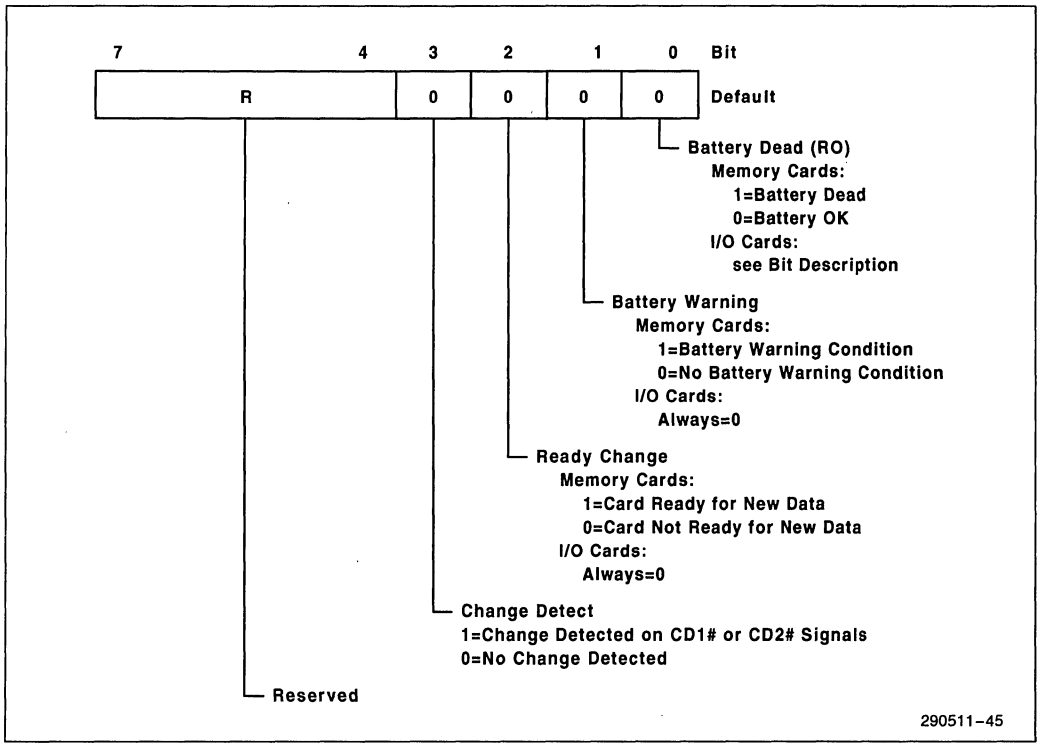


Figure 27. Card Status Change Register

5

Bits[7:4]: Reserved

Bit 3: Change Detect
This bit is set to 1 when a change occurs in either the CD1# or CD2# signal, or when a Software Interrupt is generated.

Bit 2: Ready Change
This bit is set to 1 when a low-to-high transition occurs on the RDY-BSY# signal, indicating that the memory PC Card is ready to accept a new data transfer. The bit reads 0 for I/O PC Cards.

Bit 1: Battery Warning
This bit is set to 1 when a battery warning condition is detected. The bit reads 0 for I/O PC Cards.

Bit 0: Battery Dead
For memory PC Cards, this bit is set to 1 when a battery dead condition has been detected. For I/O PC Cards, it is set to 1 when the STSCHG# signal from the I/O PC Card has been asserted low. The system software must then read the status change register in the PC Card to determine why the status change signal (STSCHG#) has been asserted.

3.2.1.5 ADWEN—Address Window Enable Register

Register Offset:	Socket A—06h Socket B—46h Socket C—86h Socket D—C6h
Default value:	00h
Access:	Read/Write
Size:	8 bits

This register controls enabling of the memory and I/O mapping windows to the PC Card memory or I/O space. All bits in this register are cleared after reset.

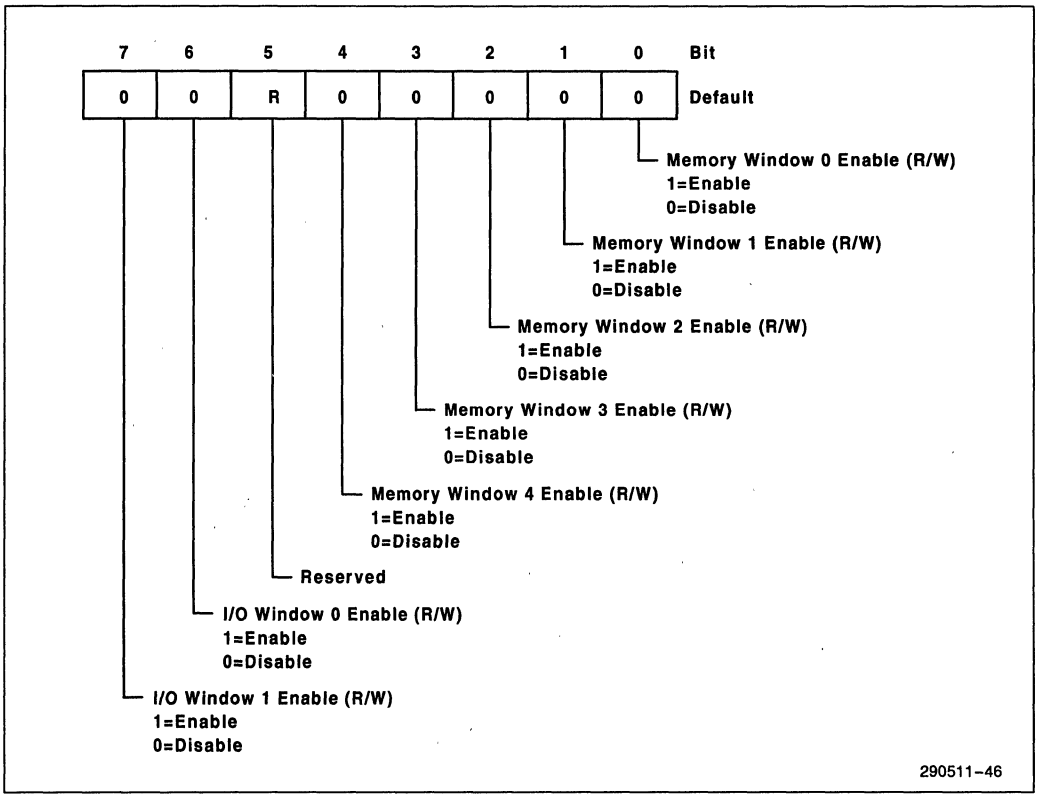


Figure 28. Address Window Enable Register

Bit 7–Bit 6: I/O Window 1 and Window 0 Enables
Bits[7:6] function identically and independently. Bit 7 applies to I/O Window 1; Bit 6 applies to I/O Window 0. When the bits are set to 0, the card enable signals to the PC cards that are accessed through the corresponding I/O windows are inhibited. When set to one, the card enable signals are not inhibited. I/O accesses pass addresses from the system bus directly through to the PC cards. The corresponding Start and Stop register pairs must all be set to the desired window values before setting either of the bits to one.

Bit 5: Reserved

Bit 4–Bit 0: Memory Window 4 - Memory Window 0 Enables
Bits[4:0] function identically and independently. Bit 4 applies to Memory Window 4, bit 3 applies to Memory Window 3, etc. When the bits are set to 0, the card enable signals to PC cards that are accessed through the corresponding memory windows are inhibited. When set to one, the card enable signals are not inhibited. The corresponding start, stop, and offset register pairs must all be set to the desired window values before setting any of the bits to 1. When one of the bits is set to 1 and the system address is within the corresponding window, the computed address will be generated for the accessed PC Card.

3.2.1.6 GCTRL—Global Control Register

Register Offset:	Socket A—1Eh
	Socket B—5Eh
	Socket C—9Eh
	Socket D—DEh
Default value:	00h
Access:	Read/Write
Size:	8 bits

This register is not duplicated for each socket, but can be accessed with the Socket A, B, C or D index. PCI reset clears all bits in this register.

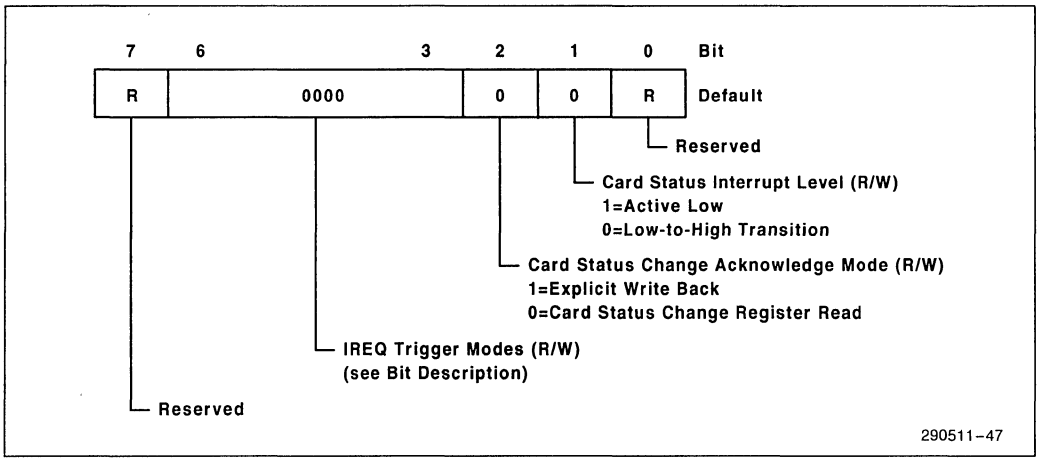


Figure 29. Global Control Register

Bit 7: Reserved

Bits[6:3]: IREQ Trigger Modes
When set to 1, these bits select level mode interrupts for IRQs generated by the particular PC card interrupts. When set to 0 (default), they select edge mode interrupts. Bit 3 is used for Socket A, Bit 4 for Socket B, Bit 5 for Socket C, and Bit 6 for Socket D

Bit 2: Card Status Change Acknowledge Mode
When this bit is set to 1, each Card Status Change Interrupt is acknowledged with an explicit write of 1 to the Card Status Change Register bit that identifies the interrupt. When this bit is set to 0 (default state), each Card Status Change Interrupt is acknowledged by reading the Card Status Change Register. Reading the Card Status Change Register clears the register.

Bit 1: Card Status Interrupt Level
When this bit is set to 1, the mode of the IRQ outputs used to signal the Card Status Change (CSC) Interrupt is active low level. In this mode, the IRQs remain tri-stated until there is a card status change condition, at which time the asserted IRQ output goes low. The IRQ remains low until the interrupt is acknowledged (serviced). Once serviced, the IRQ output will change from low to tri-state.

When this bit is set to its default state of 0, the CSC IRQ outputs are low-to-high edge triggered interrupts. In this mode, the IRQ signals remain tri-stated until enabled for CSC interrupt, at which time the IRQ outputs are asserted low. The outputs stay low until there is a card status change condition, which causes the appropriate IRQ output to transition to the high level. It will remain high until the interrupt is acknowledged (serviced), then transitions to the low state. When disabled, the IRQ signals are tri-stated.

Bit 0: Reserved

3.2.1.7 GSCTRL—Global Security Control Register

Register Offset:	Socket A—2Eh Socket B—6Eh Socket C—AEh Socket D—EEh
Default value:	00h
Access:	Read/Write
Size:	8 bits

This register is not duplicated for each socket, but can be accessed with the Socket A, B, C or D index. PCI reset clears all bits in this register.

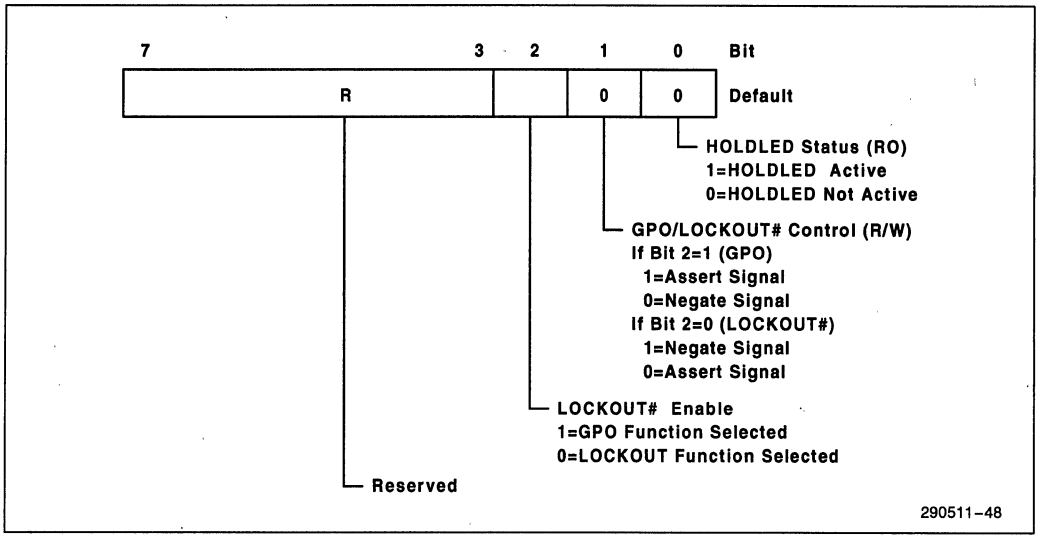


Figure 30. Global Security Control Register

Bits[7:3]: Reserved

Bit 2: Lockout# Enable
This bit configures bit 1 to function as GPO control when set to 1, and as LOCKOUT# control when set to 0.

Bit 1: General Purpose Output Control
When bit 2 is set to 1, this bit allows software control of the General Purpose Output control signal. When bit 2 is set to 0, this bit allows software control of the LOCKOUT# output signal. When bit 2 is 0 (LOCKOUT function) and this bit is 0, the LOCKOUT# signal is asserted, and the xCD2# signals function as "eject request". When bit 2 is 0 and this bit is 1, LOCKOUT# is negated, and the xCD2# signals retain their original function as Card Detect signals. See the PPEC Design Guide for details.

Bit 0: HOLDLED Status
This bit provides the status of the HOLDLED# output signal (pin), and is valid only in Mode 1.

3.2.1.8 CDGEN—Card Detect and General Control Register

Register Offset:	Socket A—16h
	Socket B—56h
	Socket C—96h
	Socket D—D6h
Default value:	00h
Access:	Read/Write
Size:	8 bits

This register is used to reset configuration registers and store voltage select signal status. It is necessary that the Configuration Reset Enable bit is set to 1 by the card detect change interrupt service routine only when a PC Card is inserted, and set to 0 when the card is removed.

Bit 7: VS2 Voltage Select Status
This bit indicates the status of the VS2 multifunctional signal, which is used with VS1 to select proper V_{CC} voltage (5V or 3.3V or disable) at a socket.

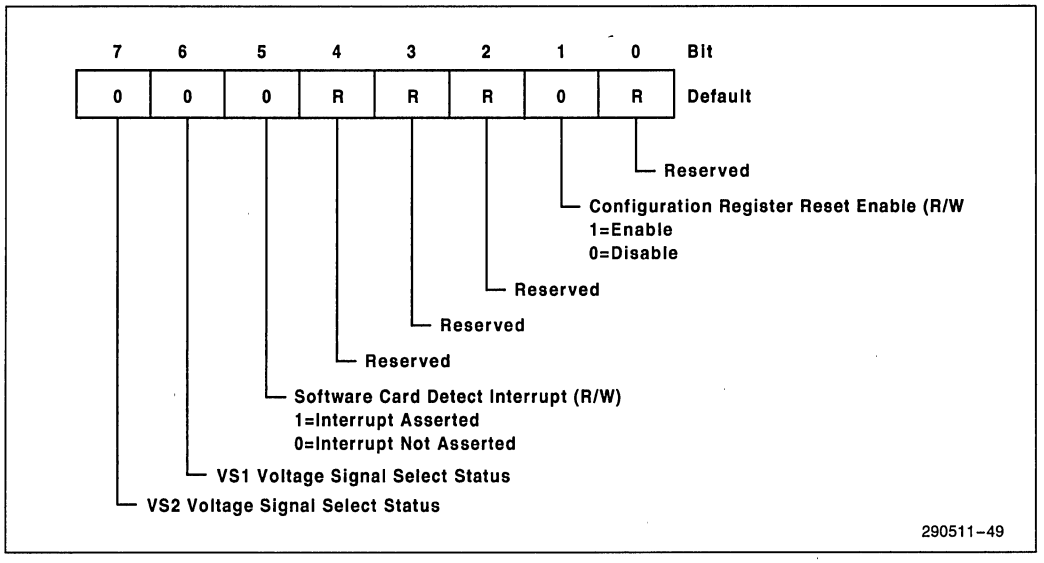


Figure 31. Card Detect and General Control Register

Bit 6: VS1 Voltage Select Status
This bit indicates the status of the VS1 voltage select signal, which is used to select proper V_{CC} voltage (5V or 3.3V or disable) at a socket.

Bit 5: Software Card Detect Interrupt
Setting this bit to 1 causes a Card Detect Card Status Change Interrupt for the associated socket if the Card Detect Enable bit is set to 1 in the Card Status Change Interrupt Configuration Register. The software interrupt functions and is acknowledged in the same manner as the hardware-generated interrupt.

The Hardware Card Detect Card Status Change Interrupt is not affected by the Software Card Detect Interrupt. The previous state of the CD1 and CD2 inputs are latched so that though a Card Detect Card Status Change Interrupt occurs and is serviced and the CD1 and CD2 inputs change from the previous state, a Hardware Card Detect Card Status Change Interrupt is still generated. If the Card Detect Enable bit is set to 0 in the Card Status Change Interrupt Configuration Register, writing a 1 to the Software Card Detect Interrupt bit has no effect.

The Software Card Detect Interrupt bit always reads back as a 0.

Bit 4: Reserved

Bit 3: Reserved

Bit 2: Reserved

Bit 1: Configuration Register Reset Enable
When this bit is set to 0, the configuration register reset function that is based on card detect is disabled. When it is set to 1, a reset pulse is generated to reset the configuration registers for the socket to their default state (zero's) when both the CD1 and CD2 inputs for the socket go high. There is one Configuration Register Reset Enable for each socket.

Bit 0: Reserved

5

3.2.1.9 CPAGE—Card Memory Page Address Register

Register Offset: Socket A—26h
Socket B—66h
Socket C—A6h
Socket D—E6h
Default value: 00h
Access: Read/Write
Size: 8 bits

This register holds an 8-bit page address that allows selection of a 16 MByte window page in the 4 GByte memory address space in which socket memory windows are mapped. Access to a window is allowed only when the page address in the corresponding Card Memory Page Address Register matches PCI memory address bits A[31:24], indicating a page hit. Reset clears all bits in this register, so that the default page is the first page (i.e., 0-16 MByte address range).

Bits[7:0]: Page Address
Page Address bits[7:0] correspond to system address lines A[31:24]. Access to one of the five memory windows is allowed only if a match between the page address and system address lines A[31:24] occurs.

3.2.2 INTERRUPT REGISTERS

The Interrupt Registers are listed in Table 9. The registers are 8-bit, read/write registers.

Table 9. Interrupt Registers

Register Offset Sockets A:D				Mnemonic	Register Name	Access
A	B	C	D			
03h	43h	83h	C3h	IGENC	Interrupt and General Control	R/W
05h	45h	85h	C5h	CSCICR	Card Status Change Interrupt Configuration	R/W

3.2.2.1 IGENC—Interrupt and General Control Register

Register Offset: Socket A—03h
Socket B—43h
Socket C—83h
Socket D—C3h
Default value: 00h
Access: Read/Write
Size: 8 bits

The Interrupt and General Control Register controls card type selection, card reset, and interrupt steering for the PC Card I/O interrupts.

Bit 7: Reserved

Bit 6: Card Reset
Setting this bit to 0 resets the PC Card by activating the RESET signal to the card. The RESET signal remains active until bit is set to 1.

Bit 5: Card Type
Setting this bit to 1 selects I/O PC Card, enabling the PC Card interface multiplexer to route PC Card I/O signals. Setting the bit to 0 selects Memory PC Card. When the bit is set to 1 (I/O PC Card), the STSCHG# signal from the I/O PC Card is used as the STSCHG status change signal. The current status of the signal is then available to be read from the Interface Status Register (01H), and the STSCHG signal can be configured as a source for the Card Status Change Interrupt.

Bit 4: Reserved

Bits[3:0]: IRQ Level Selection
This field selects interrupt routing for I/O PC Cards only.

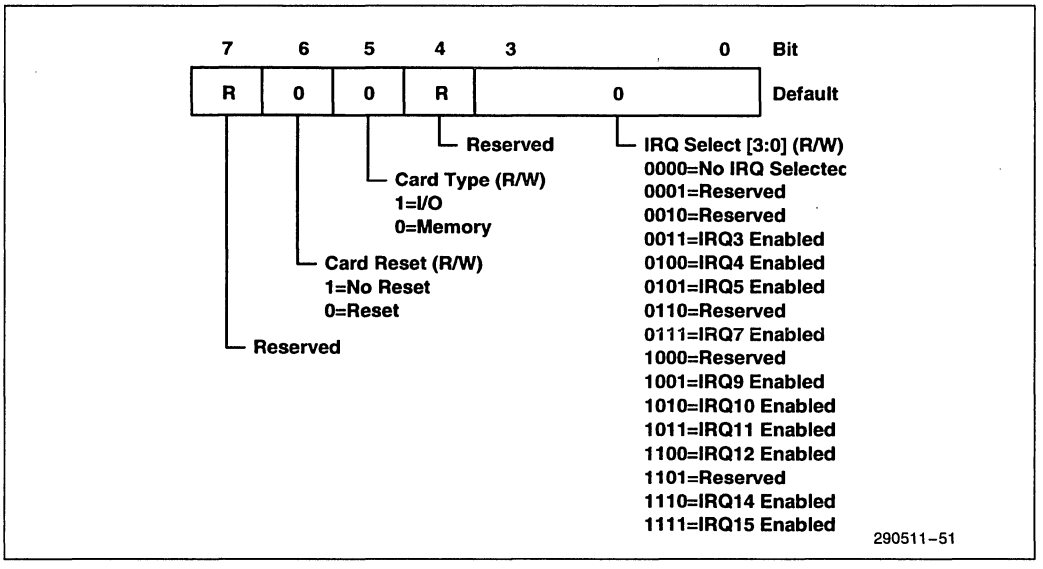


Figure 32. I/O Control Register

5

3.2.2.2 CSCICR—Card Status Change Interrupt Configuration Register

Register Offset: Socket A—05h
Socket B—45h
Socket C—85h
Socket D—C5h
Default value: 00h
Access: Read/Write
Size: 8 bits

This register controls Card Status Change Interrupt steering and the Card Status Change Interrupt enables.

Bits[7:4]: Card Status Change Interrupt Select
This field selects Card Status Change Interrupt routing.

Bit 3: Card Status Change Interrupt Enable
Setting this bit to 1 enables a Card Status Change Interrupt when a change in the CD1 or CD2 signal occurs, or when a Software Interrupt is generated by writing to bit 5 of the CDGEN register. Setting the bit to 0 disables the generation of a card status change interrupt when the CD1 or CD2 signals change state, or upon software command.

Bit 2: Ready Interrupt Enable
Setting this bit to 1 enables a Card Status Change Interrupt when a low to high transition occurs on the RDY-BSY# signal. Setting the bit to 0 disables the interrupt. The bit has no effect when the interface is configured for I/O PC Cards.

Bit 1: Battery Warning Interrupt Enable
Setting this bit to 1 enables a Card Status Change Interrupt when a battery warning condition is detected. Setting the bit to 0 disables the interrupt.

Bit 0: Battery Dead Enable
Setting this bit to 1 enables a Card Status Change Interrupt when a battery dead condition is detected in a memory PC Card, and when the STSCHG# signal is pulled low by an I/O PC Card. Setting the bit to 0 disables the interrupt.

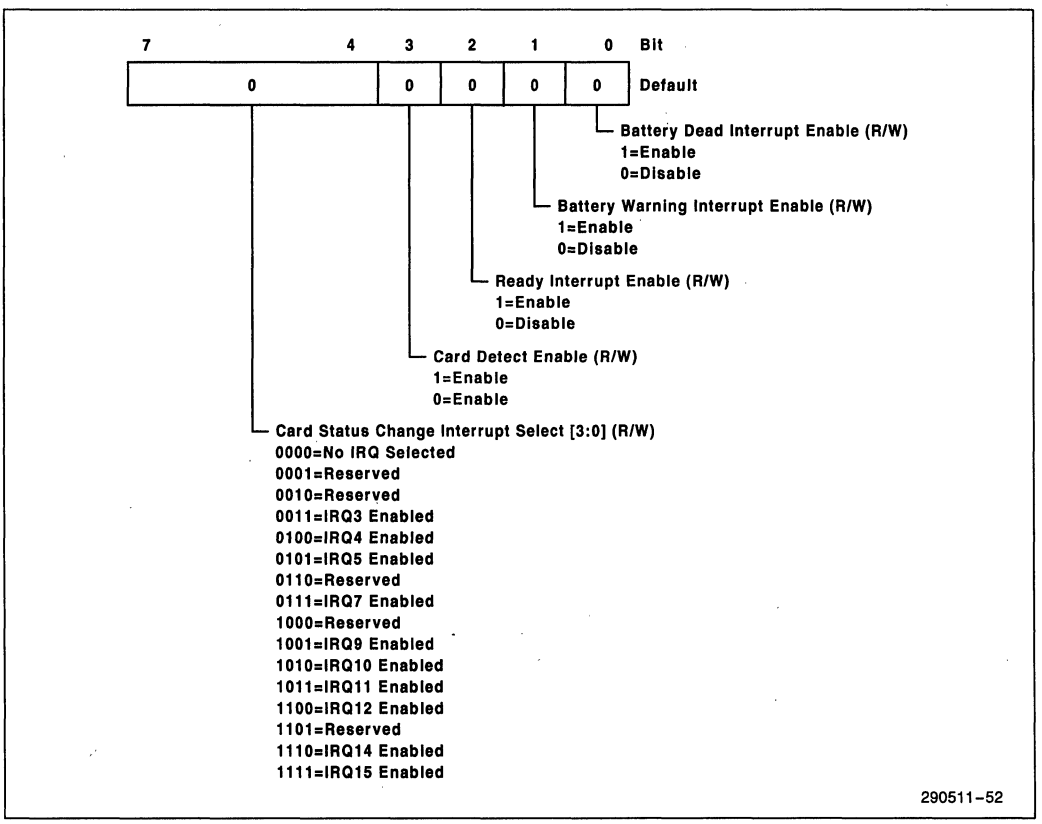


Figure 33. Card Status Change Interrupt Configuration Register

3.2.3 I/O MAPPING CONTROL REGISTERS

The I/O Mapping Control Registers are listed in Table 10. The registers are 8 bit read/write registers that specify data path sizes and start and stop addresses for the two I/O windows.

Table 10. I/O Mapping Control Registers

Register Offset Sockets A:D				Mnemonic	Register Name	Access
A	B	C	D			
07h	47h	87h	C7h	IOCREG	I/O Control	R/W
08h	48h	88h	C8h	IOSL0	I/O Address 0 Start Low Byte	R/W
09h	49h	89h	C9h	IOSH0	I/O Address 0 Start High Byte	R/W
0Ah	4Ah	8Ah	CAh	IOSTL0	I/O Address 0 Stop Low Byte	R/W
0Bh	4Bh	8Bh	CBh	IOSTH0	I/O Address 0 Stop High Byte	R/W
0Ch	4Ch	8Ch	CCh	IOSL1	I/O Address 1 Start Low Byte	R/W
0Dh	4Dh	8Dh	CDh	IOSH1	I/O Address 1 Start High Byte	R/W
0Eh	4Eh	8Eh	CEh	IOSTL1	I/O Address 1 Stop Low Byte	R/W
0Fh	4Fh	8Fh	CFh	IOSTH1	I/O Address 1 Stop High Byte	R/W

3.2.3.1 IOCREG—I/O Control Register

Register Offset: Socket A—07h
Socket B—47h
Socket C—87h
Socket D—C7h
Default value: 00h
Access: Read/Write
Size: 8 bits

This register controls the I/O data path size for I/O windows 0 and 1. In order to be compatible with some software and hardware implementations such as an IDE interface, it is necessary that the PC Card decode two consecutive I/O addresses to determine the cycle data width. To meet the system bus timings, this type of PC Card must decode address lines A[9:0] before the card enable signal becomes active at the interface. The card decodes the address and responds to a 16-bit cycle by asserting the IOIS16# signal. The PPEC qualifies IOIS16# with the card enable signals to control internal data assembly/disassembly logic.

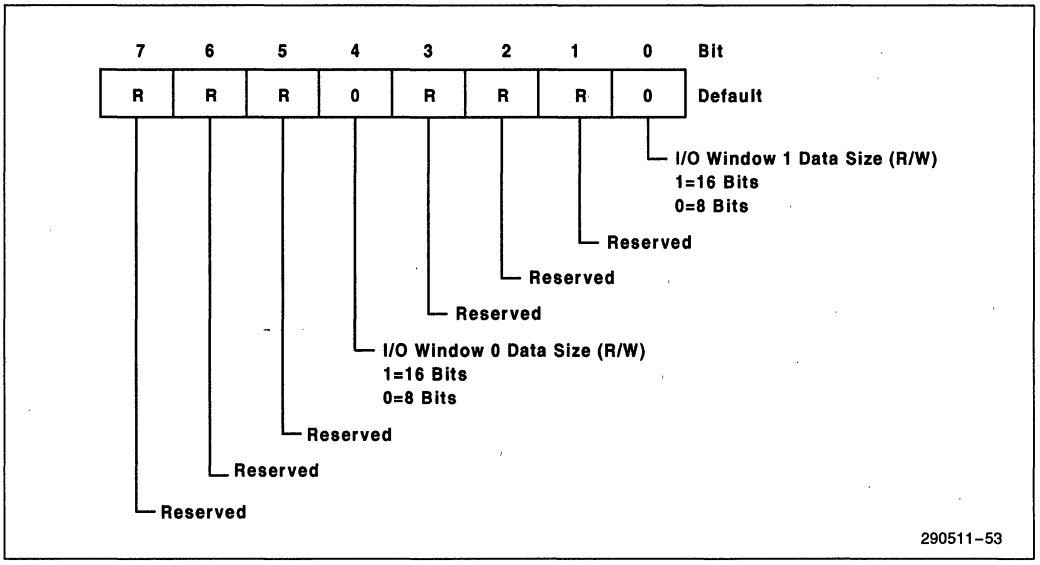


Figure 34. I/O Control Register

Bits[7:5]: Reserved

Bit 4: I/O Window 1 Data Size
This bit selects a 16-bit I/O data path to the PC Card when set to 1, and an 8-bit path when set to 0.

Bits[3:1]: Reserved

Bit 0: I/O Window 0 Data Size
This bit selects a 16-bit I/O data path to the PC Card when set to 1, and an 8-bit path when set to 0.

3.2.3.2 IOSL0—I/O Address 0 Start Low Byte Register

Register Offset:	Socket A—08h Socket B—48h Socket C—88h Socket D—C8h
Default value:	00h
Access:	Read/Write
Size:	8 bits

This register holds the low order address bits that are used to determine the start address of I/O address window 0. This provides a minimum 1 byte window for I/O address window 0 if the Start and Stop addresses are the same.

Bits[7:0]: I/O Window 0 Start Address Low Bytes
This field holds start address bits A[7:0] of I/O address window 0.

3.2.3.3 IOSH0—I/O Address 0 Start High Byte Register

Register Offset:	Socket A—09h Socket B—49h Socket C—89h Socket D—C9h
Default value:	00h
Access:	Read/Write
Size:	8 bits

This register holds the high order address bits that are used to determine the start address of I/O address window 0.

Bits[7:0]: I/O Window 0 Start Address High Bytes
This field holds start address bits A[15:8] of I/O address window 0.

3.2.3.4 IOSTL0—I/O Address 0 Stop Low Byte Register

Register Offset:	Socket A—0Ah Socket B—4Ah Socket C—8Ah Socket D—CAh
Default value:	00h
Access:	Read/Write
Size:	8 bits

This register holds the low order address bits that are used to determine the start address of I/O address window 0. This provides a minimum 1 byte window for I/O address window 0 if the Start and Stop addresses are the same.

Bits[7:0]: I/O Window 0 Stop Address Low Bytes
This field holds stop address bits A[7:0] of I/O address window 0.

3.2.3.5 IOSTH0—I/O Address 0 Stop High Byte Register

Register Offset:	Socket A—0Bh Socket B—4Bh Socket C—8Bh Socket D—CBh
Default value:	00h
Access:	Read/Write
Size:	8 bits

This register holds the high order address bits that are used to determine the stop address of I/O address window 0. This provides a minimum 1 byte window for I/O address window 0 if the Start and Stop addresses are the same.

Bits[7:0]: I/O Window 0 Stop Address High Bytes
This field holds stop address bits A[15:8] of I/O address window 0.

3.2.3.6 IOSL1—I/O Address 1 Start Low Byte Register

Register Offset:	Socket A—0Ch
	Socket B—4Ch
	Socket C—8Ch
	Socket D—CCh
Default value:	00h
Access:	Read/Write
Size:	8 bits

This register holds the low order address bits that are used to determine the start address of I/O address window 1. This provides a minimum 1 byte window for I/O address window 0 if the Start and Stop addresses are the same.

Bits[7:0]: I/O Window 1 Start Address Low Bytes
This field holds start address bits A[7:0] of I/O address window 1.

3.2.3.7 IOSH1—I/O Address 1 Start High Byte Register

Register Offset:	Socket A—0Dh
	Socket B—4Dh
	Socket C—8Dh
	Socket D—CDh
Default value:	00h
Access:	Read/Write
Size:	8 bits

This register holds the high order address bits that are used to determine the start address of I/O address window 1.

Bits[7:0]: I/O Window 1 Start Address High Bytes
This field holds start address bits A[15:8] of I/O address window 1.

3.2.3.8 IOSTL1—I/O Address 1 Stop Low Byte Register

Register Offset:	Socket A—0Eh
	Socket B—4Eh
	Socket C—8Eh
	Socket D—CEh
Default value:	00h
Access:	Read/Write
Size:	8 bits

This register holds the low order address bits that are used to determine the start address of I/O address window 1. This provides a minimum 1 byte window for I/O address window 1 if the Start and Stop addresses are the same.

Bits[7:0]: I/O Window 1 Stop Address Low Bytes
This field holds stop address bits A[7:0] of I/O address window 1.

3.2.3.9 IOSTH1—I/O Address 1 Stop High Byte Register

Register Offset:	Socket A—0Fh
	Socket B—4Fh
	Socket C—8Fh
	Socket D—CFh
Default value:	00h
Access:	Read/Write
Size:	8 bits

This register holds the high order address bits that are used to determine the stop address of I/O address window 1. This provides a minimum 1 byte window for I/O address window 1 if the Start and Stop addresses are the same.

Bits[7:0]: I/O Window 1 Stop Address High Bytes
This field holds stop address bits A[15:8] of I/O address window 1.

3.2.4 MEMORY MAPPING CONTROL REGISTERS

The Memory Mapping Control Registers are 8 bit, read/write registers that specify the starting and stopping addresses of the five memory windows. The registers are listed in Table 11.

The registers are identical for each window. Therefore, only one of each type of register is shown in the following descriptions. The addresses for all five windows for each socket are shown, however, in the *Register Offset* section of each register description.

Table 11. Memory Mapping Control Registers

Register Offset Sockets A:D				Mnemonic	Register Name	Access
A	B	C	D			
10h	50h	90h	D0h	SMSL0	System Memory Address Mapping Window 0 Start Low Byte	R/W
18h	58h	98h	D8h	SMSL1	System Memory Address Mapping Window 1 Start Low Byte	R/W
20h	60h	A0h	E0h	SMSL2	System Memory Address Mapping Window 2 Start Low Byte	R/W
28h	68h	A8h	E8h	SMSL3	System Memory Address Mapping Window 3 Start Low Byte	R/W
30h	70h	B0h	F0h	SMSL4	System Memory Address Mapping Window 4 Start Low Byte	R/W
11h	51h	91h	D1h	SMSH0	System Memory Address Mapping Window 0 Start High Byte	R/W
19h	59h	99h	D9h	SMSH1	System Memory Address Mapping Window 1 Start High Byte	R/W
21h	61h	A1h	E1h	SMSH2	System Memory Address Mapping Window 2 Start High Byte	R/W
29h	69h	A9h	E9h	SMSH3	System Memory Address Mapping Window 3 Start High Byte	R/W
31h	71h	B1h	F1h	SMSH4	System Memory Address Mapping Window 4 Start High Byte	R/W
12h	52h	92	D2h	SMSTL0	System Memory Address Mapping Window 0 Stop Low Byte	R/W
1Ah	5Ah	9Ah	DAh	SMSTL1	System Memory Address Mapping Window 1 Stop Low Byte	R/W
22h	62h	A2h	E2h	SMSTL2	System Memory Address Mapping Window 2 Stop Low Byte	R/W
2Ah	6Ah	AAh	EAh	SMSTL3	System Memory Address Mapping Window 3 Stop Low Byte	R/W
32	72h	B2h	F2h	SMSTL4	System Memory Address Mapping Window 4 Stop Low Byte	R/W
13h	53h	93h	D3h	SMSTH0	System Memory Address Mapping Window 0 Stop High Byte	R/W
1Bh	5Bh	9Bh	DBh	SMSTH1	System Memory Address Mapping Window 1 Stop High Byte	R/W
23h	63h	A3h	E3h	SMSTH2	System Memory Address Mapping Window 2 Stop High Byte	R/W

5

Table 11. Memory Mapping Control Registers (Continued)

Register Offset Sockets A:D				Mnemonic	Register Name	Access
A	B	C	D			
2Bh	6Bh	ABh	EBh	SMSTH3	System Memory Address Mapping Window 3 Stop High Byte	R/W
33h	73h	B3h	F3h	SMSTH4	System Memory Address Mapping Window 4 Stop High Byte	R/W
14h	54h	94h	D4h	OFFL0	Card Memory Offset Address 0 Low Byte	R/W
1Ch	5Ch	9Ch	DCh	OFFL1	Card Memory Offset Address 1 Low Byte	R/W
24h	64h	A4h	E4h	OFFL2	Card Memory Offset Address 2 Low Byte	R/W
2Ch	6Ch	ACh	ECh	OFFL3	Card Memory Offset Address 3 Low Byte	R/W
34h	74h	B4h	F4h	OFFL4	Card Memory Offset Address 4 Low Byte	R/W
15h	55h	95h	D5h	OFFH0	Card Memory Offset Address 0 High Byte	R/W
1Dh	5Dh	9Dh	DDh	OFFH1	Card Memory Offset Address 1 High Byte	R/W
25h	65h	A5h	E5h	OFFH2	Card Memory Offset Address 2 High Byte	R/W
2Dh	6D	ADh	EDh	OFFH3	Card Memory Offset Address 3 High Byte	R/W
35h	75h	B5h	F5h	OFFH4	Card Memory Offset Address 4 High Byte	R/W

3.2.4.1 SMSL[4:0]—System Memory Address Mapping Windows 0-4 Start Low Byte Registers

Register Offset:

	Window 0	Window 1	Window 2	Window 3	Window 4
Socket A	10h	18h	20h	28h	30h
Socket B	50h	58h	60h	68h	70h
Socket C	90h	98h	A0h	A8h	B0h
Socket D	D0h	D8h	E0h	E8h	F0h

Default value: 00h
Access: Read/Write
Size: 8 bits

These five registers hold the low order address bits that determine the start address of the corresponding system memory address mapping windows. The register contents correspond to PCI memory address bits A[19:12], and are used to determine whether memory accesses are valid.

Bits[7:0]: System Memory Window Start Address
This field holds the system memory window start address bits A[19:12].

3.2.4.2 SMSH[4:0]—System Memory Address Mapping Windows 0-4 Start High Byte Registers

Register Offset:

	Window 0	Window 1	Window 2	Window 3	Window 4
Socket A	11h	19h	21h	29h	31h
Socket B	51h	59h	61h	69h	71h
Socket C	91h	99h	A1h	A9h	B1h
Socket D	D1h	D9h	E1h	E9h	F1h

Default value: 00h
Access: Read/Write
Size: 8 bits

These five registers hold the high order address bits that determine the start address of the corresponding system memory address mapping windows. The address bits correspond to PCI memory address bits A[23:20], and are used to determine whether memory accesses are valid. The data path size of each window is controlled by a bit in its corresponding register.

Bit 7: Data Size
This bit selects an 8-bit memory data path to the PC Card when set to 0, and a 16-bit memory data path when set to 1.

Bit 6: Reserved

Bits[5:4]: Scratch Bits

These bits can be used for general-purpose register storage and retrieval.

Bits[3:0]: Memory Window Start Address
These are high order address bits that determine the start address of the system memory address mapping window.

3.2.4.3 SMSTL[4:0]—System Memory Address Mapping Windows 0–4 Stop Low Byte Register

Register Offset:

	Window 0	Window 1	Window 2	Window 3	Window 4
Socket A	12h	1Ah	22h	2Ah	32h
Socket B	52h	59h	62h	6Ah	72h
Socket C	91h	99h	A2h	AAh	B2h
Socket D	D2h	D9h	E2h	EAh	F2h

Default value: 00h
Access: Read/Write
Size: 8 bits

These five registers hold the low order address bits that determine the stop address of the corresponding system memory address mapping windows. The register contents correspond to PCI memory address bits A[19:12], and are used to determine whether memory accesses are valid.

Bits[7:0]: System Memory Window Stop Address
This field holds the system memory window stop address bits A[19:12].

3.2.4.4 SMSTH[4:0]—System Memory Address Mapping Windows 0–4 Stop High Byte Registers

Register Offset:

	Window 0	Window 1	Window 2	Window 3	Window 4
Socket A	13h	1Bh	23h	2Bh	33h
Socket B	53h	5Bh	63h	6Bh	73h
Socket C	93h	9Bh	A3h	ABh	B3h
Socket D	D3h	DBh	E3h	EBh	F3h

Default value: 10100000b
Access: Read/Write
Size: 8 bits

These five registers contain the high order address bits that determine the stop address of the corresponding system memory address mapping windows. The address bits correspond to PCI memory address bits A[23:20], and are used to determine whether memory accesses are valid. Two bits in each of the registers select delays for 16-bit accesses to the corresponding system memory window.

Bits[7:5]: Memory Window Timing Select
PCMCIA timing parameters are independently configured for each Common Memory Window by programming these timing bits. Timing Mode 101 is the default for this field, and cannot be changed (i.e. writes to the bits 5-7 are ignored) until PPEC-PCMCIA PCICON register bit 5 (Enhanced PCMCIA Timing Mode Enable) is set to 1.

Bit 4: Reserved

Bits[3:0]: Memory Window Stop Address
This field holds system memory window stop address bits A23:A20.

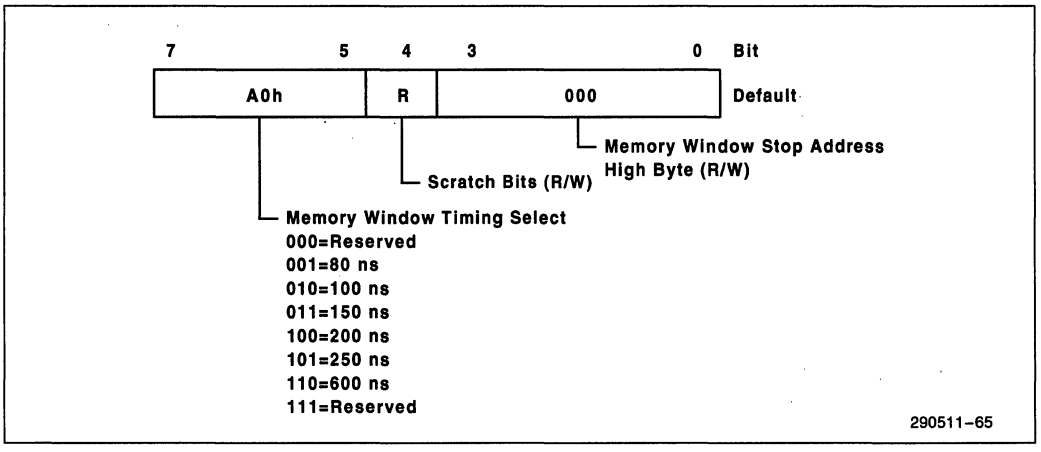


Figure 35. System Memory Address Mapping Windows 0–4 Stop High Byte Registers

3.2.4.5 OFFL[4:0]—Card Memory Offset Address 0-4 Low Byte Registers

Register Offset:

	Window 0	Window 1	Window 2	Window 3	Window 4
Socket A	14h	1Ch	24h	2Ch	34h
Socket B	54h	5Ch	64h	6Ch	74h
Socket C	94h	9Ch	A4h	ACh	B4h
Socket D	D4h	DCh	E4h	ECh	F4h

Default value: 00h
Access: Read/Write
Size: 8 bits

These five registers contain the low order address bits that are added to system address bits A[19:12] to generate the memory addresses for the PC Cards.

Bits[7:0]: Card Memory Offset Address
These bits are added to PCI memory address bits A[19:12] to generate the memory address for the PC Card.

3.2.4.6 OFFH[4:0]—Card Memory Offset Address 0 High Byte Registers

Register Offset:

	Window 0	Window 1	Window 2	Window 3	Window 4
Socket A	15h	1Dh	25h	2Dh	35h
Socket B	55h	5Dh	65h	6Dh	75h
Socket C	95h	9Dh	A5h	ADh	B5h
Socket D	D5h	DDh	E5h	EDh	F5h

Default value: 00h
Access: Read/Write
Size: 8 bits

Bits[5:0] of the registers are added to PCI memory address bits A[23:20] to generate the memory addresses for the PC Cards. The registers also control the PC Card memory software write protect for the corresponding system memory windows, and select whether the memory windows are mapped to attribute memory, or to common memory on the PC Cards.

Bit 7: Write Protect
When this bit is set to 1, write operations to the PC Card through the corresponding system memory window are inhibited. When set to 0, write operations are allowed. The WP Switch on the memory card sets the Memory Write Protect bit in the Interface Status Register, but does not alone block memory write cycles.

Bit 6: Register Active
When this bit is set to 1, accesses to the system memory window result in attribute memory on the PC Card being accessed by asserting REG low. When set to 0, accesses to the system memory result in common memory on the PC Card being accessed by driving REG high.

Bits[5:0]: Card Memory Offset Address
These bits are added to PCI memory address bits A[23:20] to generate the memory address for the PC Card.

4.0 ELECTRICAL CHARACTERISTICS

4.1 Maximum Ratings

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

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

Supply Voltages
with Respect to Ground . . . −0.5V to V_{CC} + 0.5V

Voltage On Any Pin . . . . . −0.5V to V_{CC} + 0.5V

Power Dissipation 1.0W

The junction temperature for the PPEC is 95°C with a case temperature of 85°C.

NOTICE: This data sheet contains information on products in the sampling and initial production phases of development. 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.*

4.2 Characteristics

4.2.1 PCI INTERFACE DC SPECIFICATIONS

Table 12. PCI Interface DC Characteristics (V_{CC} = 5V*5%, T_{case} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Unit	Test Condition	Note
V_{IL}	Input Low Voltage		0.8	V		
V_{IH}	Input High Voltage	2.0	V_{CC}+0.5	V		
V_{OL}	Output Low Voltage		0.55	V	I_{OL} = 6 mA	
V_{OH}	Output High Voltage	2.4		V	I_{OH}= −2.0 mA	
I_{IL}	Low-Level Input Current		−70	μA	V_{IN} = 0.5V	
I_{IH}	High-Level Input Current		70	A	V_{IN} = 2.7V	
C_{IN}	Input Capacitance		10	pF		
C_{OUT}	Output Capacitance		10	pF		
C_{CLK}	PCICLK Input Capacitance		12	pF		
I_{CC}	V_{CC} Supply Current		200	mA		

4.2.2 PCMCIA INTERFACE DC SPECIFICATIONS

Table 13. PCMCIA Interface DC Specifications (V_{CC} = 5V*5%, T_{case} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Unit	Test Condition	Note
V_{IL}	Input Low Voltage		0.8	V		
V_{IH}	Input High Voltage	2.0	V_{CC}+0.5	V		
V_{OL1}	Output Low Voltage		0.5	V	I_{OL} = −8 mA	1
V_{OH1}	Output High Voltage	2.4 2.8	V_{CC}3/5	V	I_{OH} = 4 mA I_{OH} = 2 mA	1,3,4
V_{OL2}	Output Low Voltage		0.5	V	I_{OL} = −4 mA	2
V_{OH2}	Output High Voltage	2.4 2.8	V_{CC}3/5	V	I_{OH} = 2 mA I_{OH} = 1 mA	2,3,4
I_{IL}	Low-level Input Current		−10	A		
I_{IH}	High-level Input Current		10	A		
C_{IN}	Capacitance Input		10	pF		
C_{OUT}	Capacitance Output		10	pF		
I_{CC}	V_{CC} Supply Current		200	mA		

NOTES:
1. V_{OL1} and V_{OH1} apply to PCMCIA signals that are shared in Mode 1: REG#, OE#, IOWR#, IORD#, CDATA[15:0], CADR[25:0].
2. V_{OL2} and V_{OH2} apply to all PCMCIA signals that are not listed in Note 1.
3. V_{OH} = 2.8V is the minimum high-state voltage specified by the PCMCIA specification.
4. V_{CC}3/5 is the voltage applied to the SOCKETPWR pins. This voltage may be set at 5V ±10% or 3.3V ±0.3V.

5

4.3 AC Characteristics

4.3.1 CLOCK SIGNAL AC SPECIFICATIONS

Table 14. Clock Signal AC Characteristics (V_{CC} = 5V ±5%, T_{case} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Unit	Note	Figure
t1a	Cycle Time	30		ns		36
t1b	High Time	12		ns	At 2.0V	36
t1c	Low Time	12		ns	At 0.8V	36
t1d	Rise Time		3	ns	0.8V to 2.0V	36
t1e	Fall Time		3	ns	2.0V to 0.8V	36

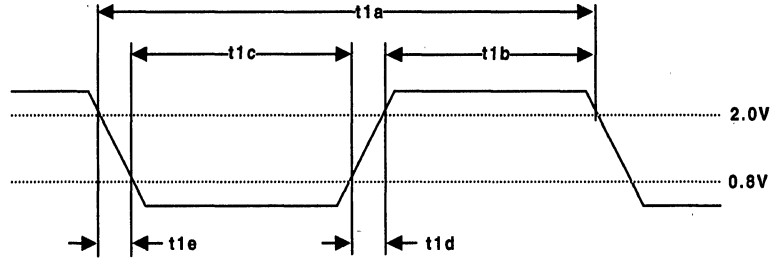


Figure 36. PCICLK Timing

4.3.2 PCI INTERFACE AC SPECIFICATIONS

Table 15. PCI Interface AC Characteristics (V_{CC} = 5V ±5%, T_{case} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Unit	Note	Figure
	PCIRST#					
t2a	Pulse Width	1		ms		37
t2b	PCICLK Active Setup to PCIRST# Negated	100		s		37
	AD[31:0], C/BE[3:0], FRAME#, IRDY#, PAR, PERR#, SERR#, TRDY#, DEVSEL#, STOP#, PCILOCK#, IDSEL					
t3a	Delay from PCICLK Rising	2	11	ns		38
t3b	Setup to PCICLK Rising	7		ns		38
t3c	Hold from PCICLK Rising	0		ns		38

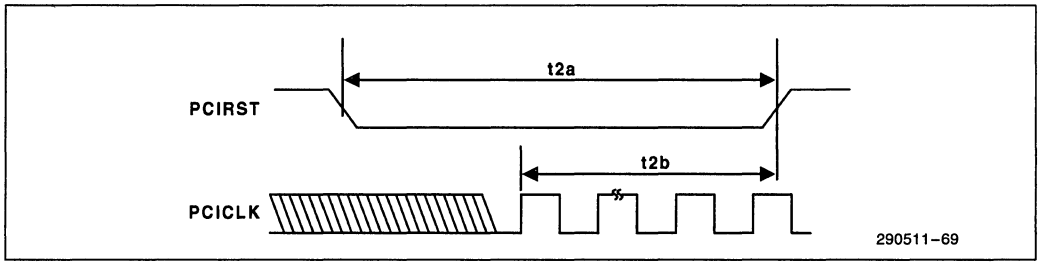


Figure 37. PCIRT# Timing

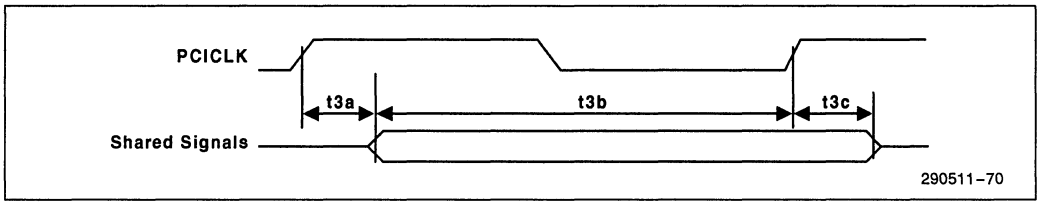


Figure 38. Shared Signal Timing

5

4.3.3 SYSTEM SIGNAL AC SPECIFICATIONS

Table 16. System Signal AC Characteristics (V_{CC} = 5V ±5%, T_{case} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Unit	Note	Figure
	IRQx					
t4a	Card Status change to IRQx/INTx# Valid		2 CLKs + 20	ns	Card Status change can be caused by any CSC event, such as the assertion of BVD[1,0], CD[1,0], STSCHG#, etc.	39
t4b	PC Card IREQ# to IRQx/INTx# Delay		35	ns		39
	SPKROUT#					
t4c	SPKR# to SPKROUT# Delay		35	ns		39

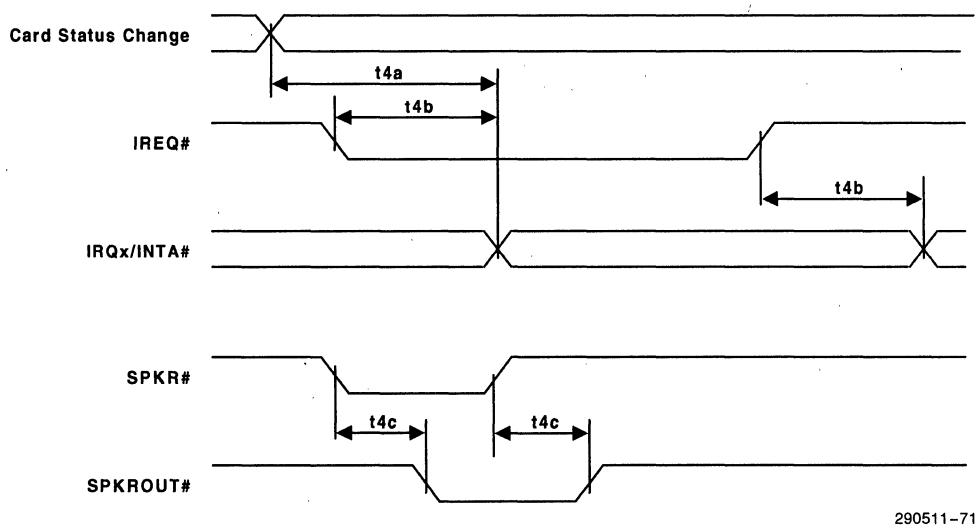


Figure 39. System Signal Timing

4.3.4 POWER WRITE SIGNAL AC CHARACTERISTICS

Table 17. Power Write Signal AC Characteristics (V_{CC} = 5V ±5%, T_{case} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Unit	Note	Figure
t5a	Data Valid Setup to PWRWR# Asserted	5		ns		40
t5b	Data Valid Hold from PWRWR# Negated	1		CLK		40
t5c	Pulse Width	5		CLK		40

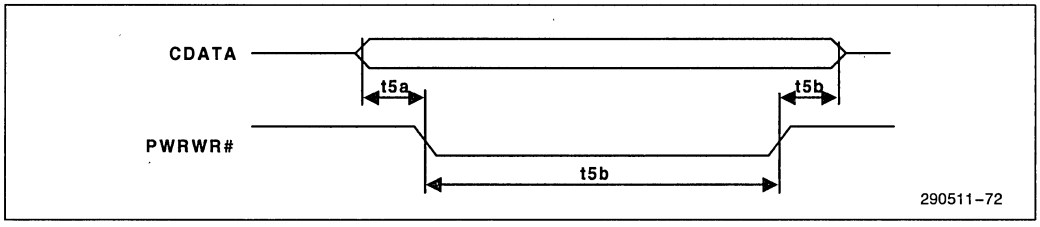


Figure 40. Power Write Signal Timing

4.3.5 PCMCIA MEMORY SIGNAL AC CHARACTERISTICS

Table 18 lists signal timing parameters for memory read and write operations. All of the parameters listed in the table apply to Common Memory operations.

Attribute Memory writes and 5.0V Attribute Memory reads use the timing for 250ns cards (see *Notes* column); 3.3V Attribute Memory reads use the timing for 600 ns cards.

Table 18. PCMCIA Memory Signal AC Characteristics (V_{CC} = 5V ±5%, T_{case} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Unit	Note	Figure
	CDATA[15:0]					
t7a	Valid Setup to WE# Asserted	2 CLKs − 20 2 CLKs − 20 2 CLKs − 20 2 CLKs − 20 3 CLKs − 20(1) 4 CLKs − 20(2)		ns	80 ns Card 100 ns Card 150 ns Card 200 ns Card 250 ns Card 600 ns Card	42
t7b	Hold from WE# Negated	1 CLK − 10 1 CLK − 10 2 CLKs − 10 2 CLKs − 10 2 CLKs − 10(1) 5 CLKs − 10(2)		ns	80 ns Card 100 ns Card 150 ns Card 200 ns Card 250 ns Card 600 ns Card	42

5

Table 18. PCMCIA Memory Signal AC Characteristics (V_{CC} = 5V ±5%, T_{case} = 0°C to +85°C)
(Continued)

Symbol	Parameter	Min	Max	Unit	Note	Figure
	CADR[25:0],REG#,CE[2:1]#,EXTDIR					
t6a	Valid Setup to OE# Asserted	1 CLK − 20 2 CLKs − 20 2 CLKs − 20 2 CLKs − 20 3 CLKs − 20(1) 4 CLKs − 20(2)		ns	80 ns Card 100 ns Card 150 ns Card 200 ns Card 250 ns Card 600 ns Card	41
t6b	Valid Setup to data Latched	4 CLKs − 20 5 CLKs − 20 6 CLKs − 20 8 CLKs − 20 10 CLKs − 20(1) 22 CLKs − 20(2)		ns	80 ns Card 100 ns Card 150 ns Card 200 ns Card 250 ns Card 600 ns Card	41
t7c	Valid Setup to WE# Asserted	2 CLKs − 20 2 CLKs − 20 2 CLKs − 20 2 CLKs − 20 3 CLKs − 20(1) 4 CLKs − 20(2)		ns	80 ns Card 100 ns Card 150 ns Card 200 ns Card 250 ns Card 600 ns Card	42
t7d	Hold from WE# Negated	1 CLK − 10 1 CLK − 10 2 CLKs − 10 2 CLKs − 10 2 CLKs − 10(1) 5 CLKs − 10(2)		ns	80 ns Card 100 ns Card 150 ns Card 200 ns Card 250 ns Card 600 ns Card	42
	OE#					
t6c	OE# Asserted to Data Latched	3 CLKs − 20 3 CLKs − 20 4 CLKs − 20 6 CLKs − 20 7 CLKs − 20(1) 18 CLKs − 20(2)		ns	80 ns Card 100 ns Card 150 ns Card 200 ns Card 250 ns Card 600 ns Card	41
t6d	Data Latched to OE# Negated	0		ns		41
	WE#					
t7e	Pulse Width	3 CLKs + 0 3 CLKs + 0 3 CLKs + 0 5 CLKs + 0 6 CLKs + 0(1) 11 CLKs + 0(2)		ns	80 ns Card 100 ns Card 150 ns Card 200 ns Card 250 ns Card 600 ns Card	42

Table 18. PCMCIA Memory Signal AC Characteristics
(V_{CC} = 5V ±5%, T_{case} = 0°C to +85°C) (Continued)

Symbol	Parameter	Min	Max	Unit	Note	Figure
	WAIT#					
t6e	WAIT# Negated to Data Latched	1 CLK + 0	2 CLKs + 10	ns	All Cards	41
t7f	WAIT# Negated to WE# Negated	1 CLK + 0	2 CLKs + 10	ns	All Cards	42
t6f	Valid Delay from CADR[25:0] Valid		50			41

NOTES:
1. Applies to Common Memory reads and writes, Attribute Memory writes, and 5.0V Attribute Memory reads.
2. Applies to Common Memory reads and writes, and 3.3V Attribute Memory reads.

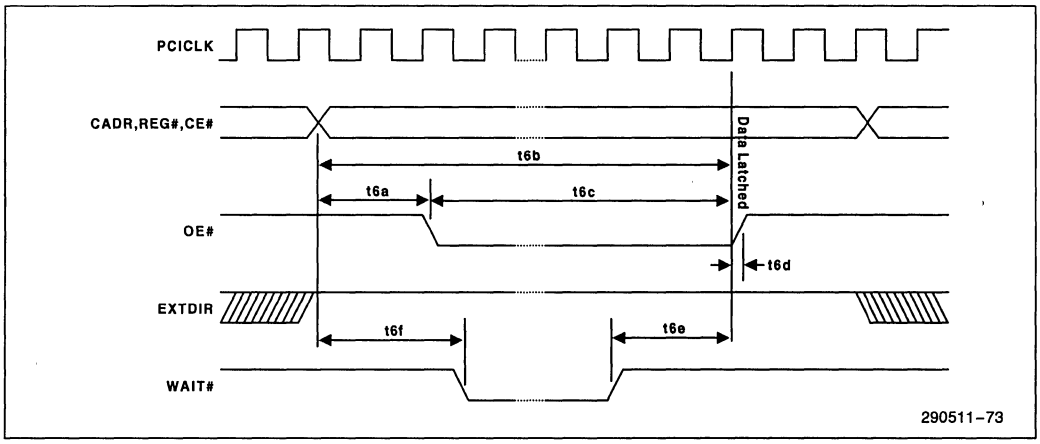


Figure 41. Memory Read Timing

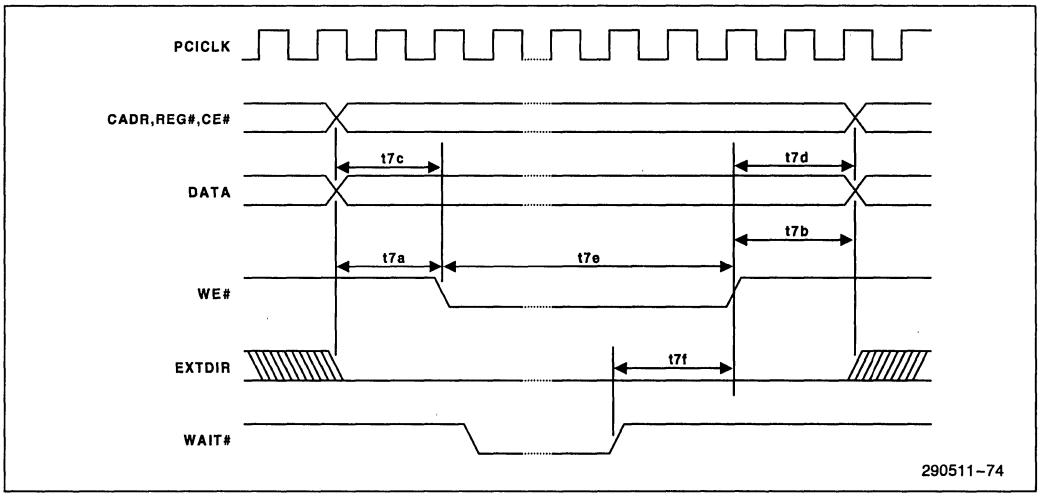


Figure 42. Memory Write Timing

5

4.3.6 PCMCIA I/O SIGNAL AC CHARACTERISTICS

Table 19 lists signal timing parameters for I/O read and write operations.

Table 19. PCMCIA I/O Signal AC Characteristics

Symbol	Parameter	Min	Max	Unit	Note	Figure
	CDATA[15:0]					
t9a	Valid Setup to IOWR# Asserted	4 CLKs − 20		ns		44
t9b	Hold from IOWR# Negated	2 CLKs − 10		ns		44
	CADR[25:0],REG#,CE[2:1]#,EXTDIR					
t8a	Valid Setup to IORD# Asserted	4 CLKs − 20				43
t8b	Valid Setup to Data Latched	10 CLKs − 20				43
t9c	Valid Setup to IOWR# Asserted	4 CLKs − 20				44
t9d	Hold from IOWR# Negated	2 CLKs − 10				44
	IORD#					
t8c	IORD# Asserted to Data Latched	6 CLKs − 20				43
t8d	Data Latched to OE# Negated	0				43
	IOWR#					
t9e	Pulse Width	6 CLKs + 0				44
	WAIT#					
t8e	WAIT# Negated to Data Latched	1 CLK + 0				43
t9f	WAIT# Negated to IOWR# Negated	1 CLK + 0				44
t8f,t9g	Valid Delay from IORD#/IOWR# Valid		50			43, 44
	IOIS16#					
t10g	Valid Delay from CADR[25:0] Valid		50			44

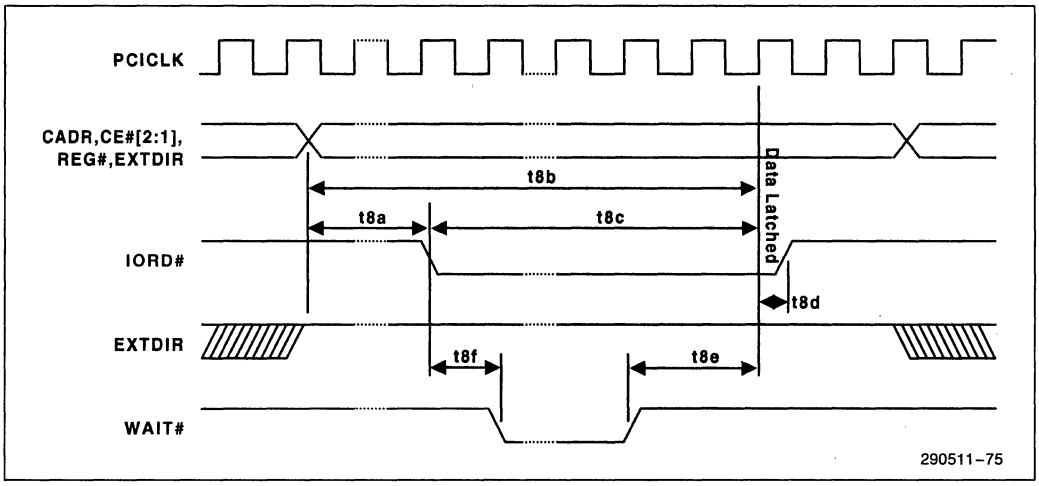


Figure 43. I/O Read Timing

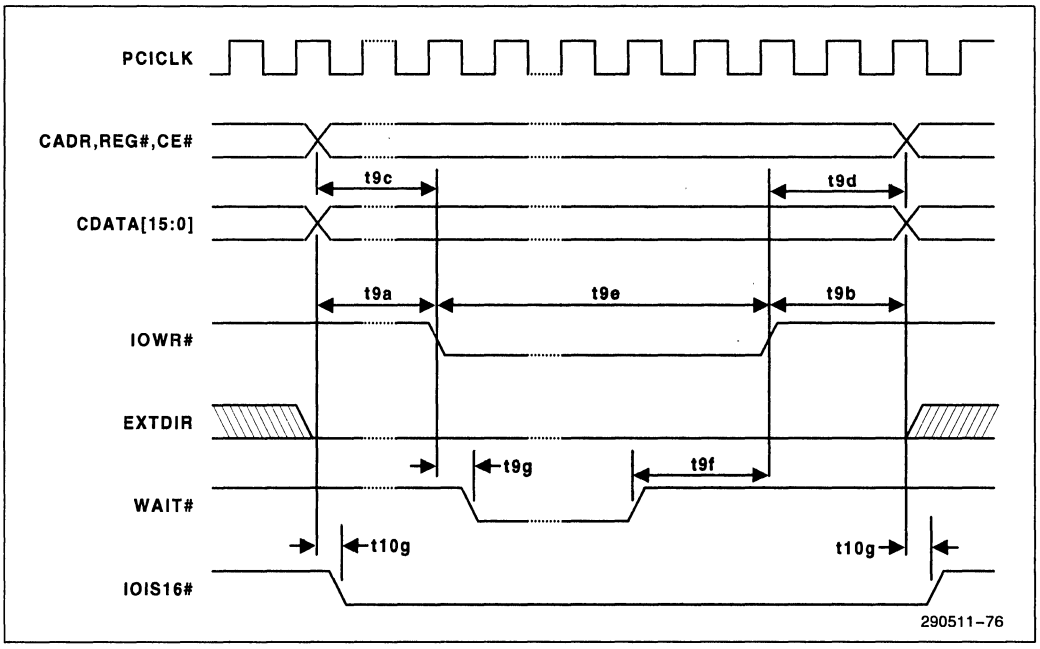


Figure 44. I/O Write Timing

5

4.3.7 IDE SIGNAL AC CHARACTERISTICS

Table 20 lists signal timing parameters with respect to PCI clock edges for IDE read and write operations. Note that integral number of clock delays are programmable for all IDE signals.

Table 20. IDE Signal AC Characteristics

Symbol	Parameter	Min	Max	Unit	Note	Figure
t10a	DIOR#/DIOW# Delay from PCI Clock	2	20	ns		45
t10b	Address/Data/CS Delay from PCI Clock	2	30	ns		45
T_{pw}	DIOR#/DIOW# Duration	1	8	CLK	Programmable	45
T_{su}	Address/Data/CS Setup to DIOR#/DIOW#	1	4	CLK	Programmable	45
T_{cyc}	Address/Data/CS Cycle Time	5	31	CLK	Programmable	45

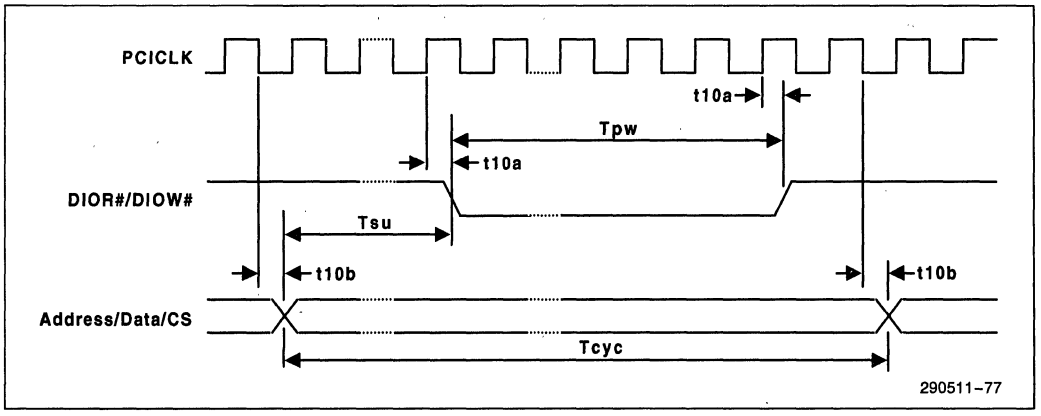


Figure 45. IDE Timing

8231A
ARITHMETIC PROCESSING UNIT

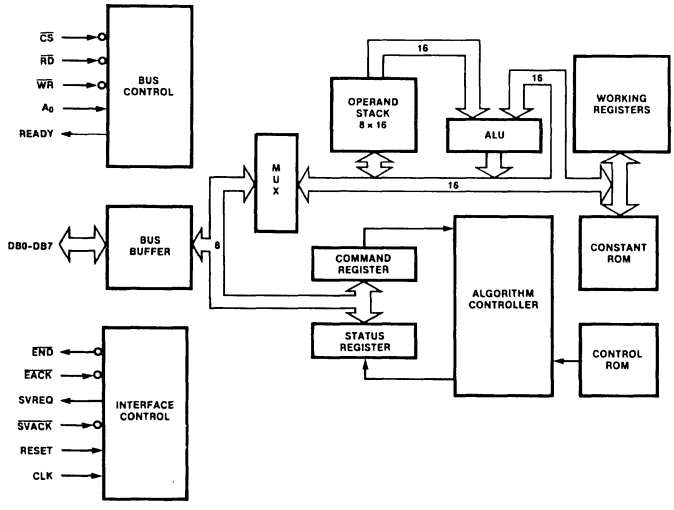
- **Fixed Point Single and Double Precision (16/32 Bit)**
- **Floating Point Single Precision (32 Bit)**
- **Binary Data Formats**
- **Add, Subtract, Multiply and Divide**
- **Trignometric and Inverse Trigonometric Functions**
- **Square Roots, Logarithms, Exponentiation**
- **Float to Fixed and Fixed to Float Conversions**
- **Stack Oriented Operand Storage**
- **Compatible with all Intel and most other Microprocessor Families**
- **Direct Memory Access or Programmed I/O Data Transfers**
- **End of Execution Signal**
- **General Purpose 8-Bit Data Bus Interface**
- **Standard 24 Pin Package**
- **+12V and +5V Power Supplies**
- **Advanced N-Channel Silicon Gate HMOS Technology**

The Intel 8231A Arithmetic Processing Unit (APU) is a monolithic HMOS LSI device that provides high performance fixed and floating point arithmetic and floating point trigonometric operations. It may be used to enhance the mathematical capability of a wide variety of processor-oriented systems. Chebyshev polynomials are used in the implementation of the APU algorithms.

All transfers, including operand, result, status and command information, take place over an 8-bit bidirectional data bus. Operands are pushed onto an internal stack and commands are issued to perform operations on the data and the stack. Results are then available to be retrieved from the stack.

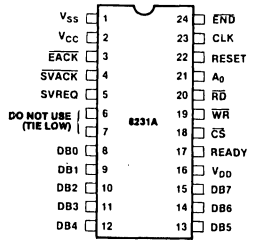
Transfers to and from the APU may be handled by the associated processor using conventional programmed I/O, or may be handled by a direct memory access controller for improved performance. Upon completion of each command, the APU issues an end of execution signal that may be used as an interrupt by the CPU to help coordinate program execution.

5



231305-1

Figure 1. Block Diagram



231305-2

Figure 2. Pin Configuration

The complete document for this product is available from Intel's Literature Center at 1-800-548-4725.

September 1993
Order Number: 231305-003

8237A
HIGH PERFORMANCE PROGRAMMABLE DMA CONTROLLER (8237A-5)

- **Enable/Disable Control of Individual DMA Requests**
- **Four Independent DMA Channels**
- **Independent Autoinitialization of All Channels**
- **Memory-to-Memory Transfers**
- **Memory Block Initialization**
- **Address Increment or Decrement**
- **High Performance: Transfers up to 1.6M Bytes/Second with 5 MHz 8237A-5**
- **Directly Expandable to Any Number of Channels**
- **End of Process Input for Terminating Transfers**
- **Software DMA Requests**
- **Independent Polarity Control for DREQ and DACK Signals**
- **Available in EXPRESS — Standard Temperature Range**
- **Available in 40-Lead Cerdip and Plastic Packages**

(See Packaging Spec, Order #231369)

The 8237A Multimode Direct Memory Access (DMA) Controller is a peripheral interface circuit for microprocessor systems. It is designed to improve system performance by allowing external devices to directly transfer information from the system memory. Memory-to-memory transfer capability is also provided. The 8237A offers a wide variety of programmable control features to enhance data throughput and system optimization and to allow dynamic reconfiguration under program control.

The 8237A is designed to be used in conjunction with an external 8-bit address latch. It contains four independent channels and may be expanded to any number of channels by cascading additional controller chips. The three basic transfer modes allow programmability of the types of DMA service by the user. Each channel can be individually programmed to Autoinitialize to its original condition following an End of Process ($\overline{EOP}$). Each channel has a full 64K address and word count capability.

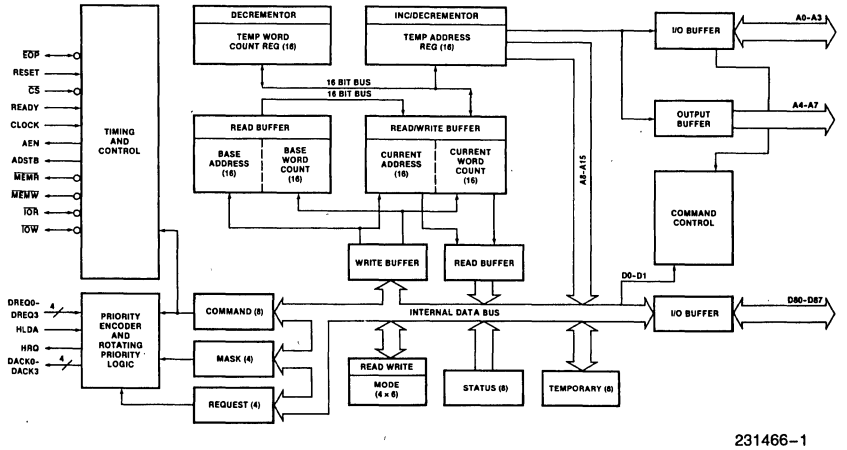


Figure 1. Block Diagram

Figure 2. Pin Configuration

The complete document for this product is available from Intel's Literature Center at 1-800-548-4725.

September 1993
Order Number: 231466-005

82C37A-5
CHMOS HIGH PERFORMANCE PROGRAMMABLE DMA CONTROLLER

- **Pin Compatible with NMOS 8237A-5**
- **Enable/Disable Control of Individual DMA Requests**
- **Fully Static Design with Frequency Range from DC to 5 MHz**
- **Low Power Operation**
- **Four Independent DMA Channels**
- **Independent Autoinitialization of all Channels**
- **Memory-to-Memory Transfers**
- **Memory Block Initialization**
- **Address Increment or Decrement**
- **High performance: 5 MHz Speed Transfers up to 1.6 MBytes/Second**
- **Directly Expandable to any Number of Channels**
- **End of Process Input for Terminating Transfers**
- **Software DMA Requests**
- **Independent Polarity Control for DREQ and DACK Signals**
- **Available in 40-Lead Plastic DIP**

The Intel 82C37A-5 Multimode Direct Memory Access (DMA) Controller is a CHMOS peripheral interface circuit for microprocessor systems. It is designed to improve system performance by allowing external devices to directly transfer information from the system memory. Memory-to-memory transfer capability is also provided. The 82C37A-5 offers a wide variety of programmable control features to enhance data throughput and system optimization and to allow dynamic reconfiguration under program control.

The 82C37A-5 is designed to be used in conjunction with an external 8-bit address register. It contains four independent channels and may be expanded to any number of channels by cascading additional controller chips.

The three basic transfer modes allow programmability of the types of DMA service by the user. Each channel can be individually programmed to Autoinitialize to its original condition following an End of Process ($\overline{EOP}$).

Each channel has a full 64K address and word count capability.

5

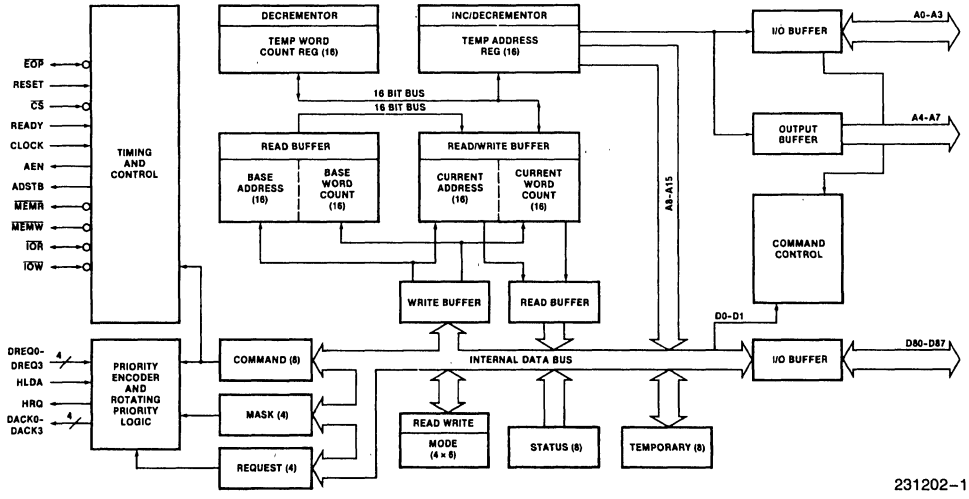


Figure 1. Block Diagram

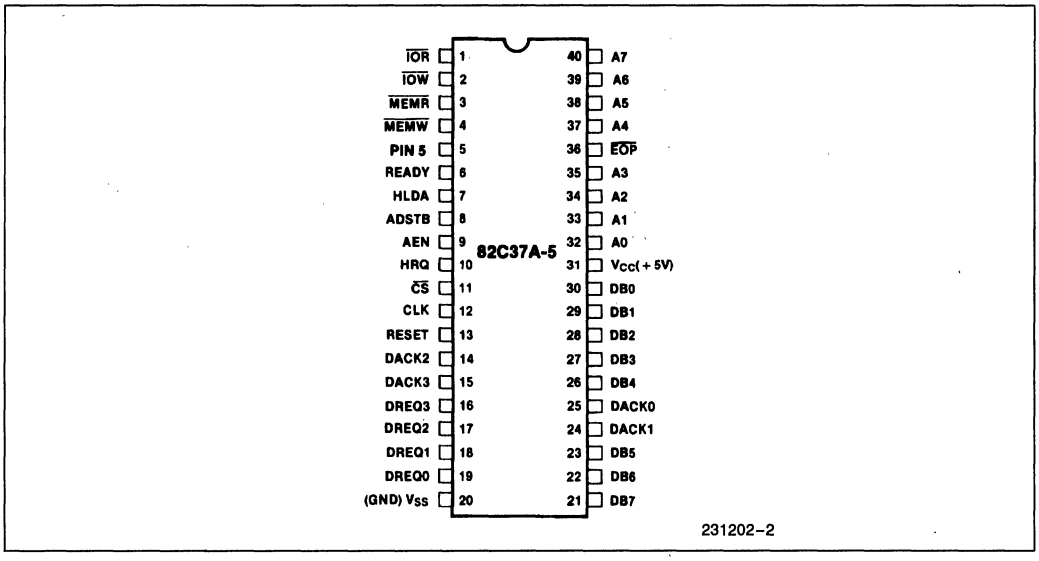


Figure 2. 82C37A-5
40-Lead DIP Configuration

Table 1. Pin Description

Symbol	Type	Name and Function
V_{CC}		**POWER:** +5 volt supply.
V_{SS}		**GROUND:** Ground.
CLK	I	**CLOCK INPUT:** Clock Input controls the internal operations of the 82C37A-5 and its rate of data transfers. The input may be driven at up to 5 MHz for the 82C37A-5.
$\overline{CS}$	I	**CHIP SELECT:** Chip Select is an active low input used to select the 82C37A-5 as an I/O device during the Idle cycle. This allows CPU communication on the data bus.
RESET	I	**RESET:** Reset is an active high input which clears the Command, Status, Request and Temporary registers. It also clears the first/last flip-flop and sets the Mask register. Following a Reset the device is in the Idle cycle.
READY	I	**READY:** Ready is an input used to extend the memory read and write pulses from the 82C37A-5 to accommodate slow memories or I/O peripheral devices. Ready must not make transitions during its specified setup/hold time.
HLDA	I	**HOLD ACKNOWLEDGE:** The active high Hold Acknowledge from the CPU indicates that it has relinquished control of the system busses.
DREQ0–DREQ3	I	**DMA REQUEST:** The DMA Request lines are individual asynchronous channel request inputs used by peripheral circuits to obtain DMA service. In fixed Priority, DREQ0 has the highest priority and DREQ3 has the lowest priority. A request is generated by activating the DREQ line of a channel. DACK will acknowledge the recognition of DREQ signal. Polarity of DREQ is programmable. Reset initializes these lines to active high. DREQ must be maintained until the corresponding DACK goes active.
DB0–DB7	I/O	**DATA BUS:** The Data Bus lines are bidirectional three-state signals connected to the system data bus. The outputs are enabled in the Program condition during the I/O Read to output the contents of an Address register, a Status register, the Temporary register or a Word Count register to the CPU. The outputs are disabled and the inputs are read during an I/O Write cycle when the CPU is programming the 82C37A-5 control registers. During DMA cycles the most significant 8 bits of the address are output onto the data bus to be strobed into an external latch by ADSTB. In memory-to-memory operations, data from the memory comes into the 82C37A-5 on the data bus during the read-from-memory transfer. In the write-to-memory transfer, the data bus outputs place the data into the new memory location.
$\overline{IOR}$	I/O	**I/O READ:** I/O Read is a bidirectional active low three-state line. In the Idle cycle, it is an input control signal used by the CPU to read the control registers. In the Active cycle, it is an output control signal used by the 82C37A-5 to access data from a peripheral during a DMA Write transfer.
$\overline{IOW}$	I/O	**I/O WRITE:** I/O Write is a bidirectional active low three-state line. In the Idle cycle, it is an input control signal used by the CPU to load information into the 82C37A-5. In the Active cycle, it is an output control signal used by the 82C37A-5 to load data to the peripheral during a DMA Read transfer.

5

Table 1. Pin Description (Continued)

Symbol	Type	Name and Function
$\overline{EOP}$	I/O	**END OF PROCESS:** End of Process is an active low bidirectional signal. Information concerning the completion of DMA services is available at the bidirectional $\overline{EOP}$ pin. The 82C37A-5 allows an external signal to terminate an active DMA service. This is accomplished by pulling the $\overline{EOP}$ input low with an external $\overline{EOP}$ signal. The 82C37A-5 also generates a pulse when the terminal count (TC) for any channel is reached. This generates an $\overline{EOP}$ signal which is output through the $\overline{EOP}$ Line. The reception of $\overline{EOP}$, either internal or external, will cause the 82C37A-5 to terminate the service, reset the request, and, if Autoinitialize is enabled, to write the base registers to the current registers of that channel. The mask bit and TC bit in the status word will be set for the currently active channel by $\overline{EOP}$ unless the channel is programmed for Autoinitialize. In that case, the mask bit remains unchanged. During memory-to-memory transfers, $\overline{EOP}$ will be output when the TC for channel 1 occurs. $\overline{EOP}$ should be tied high with a pull-up resistor if it is not used to prevent erroneous end of process inputs.
A0–A3	I/O	**ADDRESS:** The four least significant address lines are bidirectional three-state signals. In the Idle cycle they are inputs and are used by the CPU to address the register to be loaded or read. In the Active cycle they are outputs and provide the lower 4 bits of the output address.
A4–A7	O	**ADDRESS:** The four most significant address lines are three-state outputs and provide 4 bits of address. These lines are enabled only during the DMA service.
HRQ	O	**HOLD REQUEST:** This is the Hold Request to the CPU and is used to request control of the system bus. If the corresponding mask bit is clear, the presence of any valid DREQ causes 82C37A-5 to issue the HRQ. After HRQ goes active at least one clock cycle (TCY) must occur before HLDA goes active.
DACK0–DACK3	O	**DMA ACKNOWLEDGE:** DMA Acknowledge is used to notify the individual peripherals when one has been granted a DMA cycle. The sense of these lines is programmable. Reset initializes them to active low.
AEN	O	**ADDRESS ENABLE:** Address Enable enables the 8-bit latch containing the upper 8 address bits onto the system address bus. AEN can also be used to disable other system bus drivers during DMA transfers. AEN is active HIGH.
ADSTB	O	**ADDRESS STROBE:** The active high, Address Strobe is used to strobe the upper address byte into an external latch.
$\overline{MEMR}$	O	**MEMORY READ:** The Memory Read signal is an active low three-state output used to access data from the selected memory location during a DMA Read or a memory-to-memory transfer.
$\overline{MEMW}$	O	**MEMORY WRITE:** The Memory Write is an active low three-state output used to write data to the selected memory location during a DMA Write or a memory-to-memory transfer.
PIN5	I	**PIN5:** This pin should always be at a logic HIGH level. An internal pull-up resistor will establish a logic HIGH when the pin is left floating. It is recommended, however, that PIN5 be connected to V_{CC}.

FUNCTIONAL DESCRIPTION

The 82C37A-5 block diagram includes the major logic blocks and all of the internal registers. The data interconnection paths are also shown. Not shown are the various control signals between the blocks. The 82C37A-5 contains 344 bits of internal memory in the form of registers. Figure 3 lists these registers by name and shows the size of each. A detailed description of the registers and their functions can be found under Register Description.

Name	Size	Number
Base Address Registers	16 bits	4
Base Word Count Registers	16 bits	4
Current Address Registers	16 bits	4
Current Word Count Registers	16 bits	4
Temporary Address Register	16 bits	1
Temporary Word Count Register	16 bits	1
Status Register	8 bits	1
Command Register	8 bits	1
Temporary Register	8 bits	1
Mode Registers	6 bits	4
Mask Register	4 bits	1
Request Register	4 bits	1

Figure 3. 82C37A-5 Internal Registers

The 82C37A-5 contains three basic blocks of control logic. The Timing Control block generates internal timing and external control signals for the 82C37A-5. The Program Command Control block decodes the various commands given to the 82C37A-5 by the microprocessor prior to servicing a DMA Request. It also decodes the Mode Control word used to select the type of DMA during the servicing. The Priority Encoder block resolves priority contention between DMA channels requesting service simultaneously.

DMA Operation

The 82C37A-5 is designed to operate in two major cycles. These are called Idle and Active cycles. Each device cycle is made up of a number of states. The 82C37A-5 can assume seven separate states, each composed of one full clock period. State 1 (S1) is the inactive state. It is entered when the 82C37A-5 has no valid DMA requests pending. While in S1, the DMA controller is inactive but may be in the Program Condition, being programmed by the processor. State 0 (S0) is the first state of a DMA service. The 82C37A-5 has requested a hold but the processor has not yet returned an acknowledge. The 82C37A-5 may still be programmed until it receives HLDA from the CPU. An acknowledge from the CPU will signal that DMA transfers may begin. S1, S2, S3 and S4 are the working states of the DMA service. If more time is needed to complete a transfer than is available with normal timing, wait states (SW) can be inserted between S2 or S3 and S4 by the use of the Ready line on the 82C37A-5. Note that the data is transferred directly from the I/O device to memory (or vice versa) with $\overline{IOR}$ and $\overline{MEMW}$ (or $\overline{MEMR}$ and $\overline{IOW}$) being active at the same time. The data is not read into or driven out of the 82C37A-5 in I/O-to-memory or memory-to-I/O DMA transfers.

Memory-to-memory transfers require a read-from and a write-to-memory to complete each transfer. The states, which resemble the normal working states, use two digit numbers for identification. Eight states are required for a single transfer. The first four states (S11, S12, S13, S14) are used for the read-from-memory half and the last four states (S21, S22, S23, S24) for the write-to-memory half of the transfer.

IDLE CYCLE

When no channel is requesting service, the 82C37A-5 will enter the Idle cycle and perform "S1" states. In this cycle the 82C37A-5 will sample the DREQ lines every clock cycle to determine if any channel is requesting a DMA service. The device will also sample $\overline{CS}$, looking for an attempt by the microprocessor to write or read the internal registers of the 82C37A-5. When $\overline{CS}$ is low and HLDA is low, the 82C37A-5 enters the Program Condition. The CPU can now establish, change or inspect the internal definition of the part by reading from or writing to the internal registers. Address lines A0–A3 are inputs to the device and select which registers will be read or written. The $\overline{IOR}$ and $\overline{IOW}$ lines are used to select and time reads or writes. Due to the number and size of the internal registers, an internal flip-flop is used to generate an additional bit of address. This bit is used to determine the upper or lower byte of the 16-bit Address and Word Count registers. The flip-flop is reset by Master Clear or Reset. A separate software command can also reset this flip-flop.

Special software commands can be executed by the 82C37A-5 in the Program Condition. These commands are decoded as sets of addresses with the $\overline{CS}$ and $\overline{IOW}$. The commands do not make use of the data bus. Instructions include Clear First/Last Flip-Flop and Master Clear.

ACTIVE CYCLE

When the 82C37A-5 is in the Idle cycle and a non-masked channel requests a DMA service, the device

will output an HRQ to the microprocessor and enter the Active cycle. It is in this cycle that the DMA service will take place, in one of four modes:

Single Transfer Mode — In Single Transfer mode the device is programmed to make one transfer only. The word count will be decremented and the address decremented or incremented following each transfer. When the word count "rolls over" from zero to FFFFH, a Terminal Count (TC) will cause an Autoinitialize if the channel has been programmed to do so.

DREQ must be held active until DACK becomes active in order to be recognized. If DREQ is held active throughout the single transfer, HRQ will go inactive and release the bus to the system. It will again go active and, upon receipt of a new HLDA, another single transfer will be performed, in 8080A, 8085AH, 80C88, or 80C86 system this will ensure one full machine cycle execution between DMA transfers. Details of timing between the 82C37A-5 and other bus control protocols will depend upon the characteristics of the microprocessor involved.

Block Transfer Mode — In Block Transfer mode the device is activated by DREQ to continue making transfers during the service until a TC, caused by word count going to FFFFH, or an external End of Process (EOP) is encountered. DREQ need only be held active until DACK becomes active. Again, an Autoinitialization will occur at the end of the service if the channel has been programmed for it.

Demand Transfer Mode — In Demand Transfer mode the device is programmed to continue making transfers until a TC or external $\overline{EOP}$ is encountered or until DREQ goes inactive. Thus transfers may continue until the I/O device has exhausted its data capacity. After the I/O device has had a chance to catch up, the DMA service is re-established by means of a DREQ. During the time between services when the microprocessor is allowed to operate, the intermediate values of address and word count are stored in the 82C37A-5 Current Address and Current Word Count registers. Only an $\overline{EOP}$ can cause an Autoinitialize at the end of the service. $\overline{EOP}$ is generated either by TC or by an external signal.

Cascade Mode — This mode is used to cascade more than one 82C37A-5 together for simple system expansion. The HRQ and HLDA signals from the additional 82C37A-5 are connected to the DREQ and DACK signals of a channel of the initial 82C37A-5. This allows the DMA requests of the additional device to propagate through the priority network circuitry of the preceding device. The priority chain is preserved and the new device must wait for its turn to acknowledge requests. Since the cascade channel of the initial 82C37A-5 is used only for prioritizing the additional device, it does not output any address or control signals of its own. These could conflict with the outputs of the active channel in the added device. The 82C37A-5 will respond to DREQ and DACK but all other outputs except HRQ will be disabled. The ready input is ignored.

Figure 4 shows two additional devices cascaded into an initial device using two of the previous channels. This forms a two level DMA system. More 82C37A-5s could be added at the second level by using the remaining channels of the first level. Additional devices can also be added by cascading into the channels of the second level devices, forming a third level.

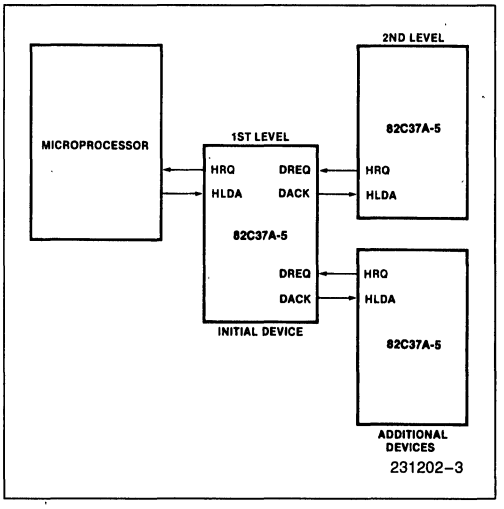


Figure 4. Cascaded 82C37A-5s

TRANSFER TYPES

Each of the three active transfer modes can perform three different types of transfers. These are Read, Write and Verify. Write transfers move data from and I/O device to the memory by activating $\overline{MEMW}$ and $\overline{IOR}$. Read transfers move data from memory to an I/O device by activating $\overline{MEMR}$ and $\overline{IOW}$. Verify transfers are pseudo transfers. The 82C37A-5 operates as in Read or Write transfers generating addresses, and responding to EOP, etc. However, the memory and I/O control lines all remain inactive. The ready input is ignored in verify mode.

Memory-to-Memory — To perform block moves of data from one memory address space to another with a minimum of program effort and time, the 82C37A-5 includes a memory-to-memory transfer feature. Programming a bit in the Command register selects channels 0 to 1 to operate as memory-to-memory transfer channels. The transfer is initiated by setting the software DREQ for channel 0. The

82C37A-5 requests a DMA service in the normal manner. After HLDA is true, the device, using four state transfers in Block Transfer mode, reads data from the memory. The channel 0 Current Address register is the source for the address used and is decremented or incremented in the normal manner. The data byte read from the memory is stored in the 82C37A-5 internal Temporary register. Channel 1 then performs a four-state transfer of the data from the Temporary register to memory using the address in its Current Address register and incrementing or decrementing it in the normal manner. The channel 1 current Word Count is decremented. When the word count of channel 1 goes to FFFFH, a TC is generated causing an $\overline{EOP}$ output terminating the service.

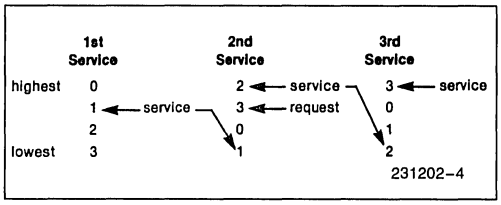
Channel 0 may be programmed to retain the same address for all transfers. This allows a single word to be written to a block of memory.

The 82C37A-5 will respond to external $\overline{EOP}$ signals during memory-to-memory transfers. Data comparators in block search schemes may use this input to terminate the service when a match is found. The timing of memory-to-memory transfers is found in Figure 12. Memory-to-memory operations can be detected as an active AEN with no DACK outputs.

Autoinitialize — By programming a bit in the Mode register, a channel may be set up as an Autoinitialize channel. During Autoinitialize initialization, the original values of the Current Address and Current Word Count registers are automatically restored from the Base Address and Base Word count registers of that channel following $\overline{EOP}$. The base registers are loaded simultaneously with the current registers by the microprocessor and remain unchanged throughout the DMA service. The mask bit is not altered when the channel is in Autoinitialize. Following Autoinitialize the channel is ready to perform another DMA service, without CPU intervention, as soon as a valid DREQ is detected. In order to Autoinitialize both channels in a memory-to-memory transfer, both word counts should be programmed identically. If interrupted externally, $\overline{EOP}$ pulses should be applied in both bus cycles.

Priority — The 82C37A-5 has two types of priority encoding available as software selectable options. The first is Fixed Priority which fixes the channels in priority order based upon the descending value of their number. The channel with the lowest priority is 3 followed by 2, 1 and the highest priority channel, 0. After the recognition of any one channel for service, the other channels are prevented from interfering with that service until it is completed.

The second scheme is Rotating Priority. The last channel to get service becomes the lowest priority channel with the others rotating accordingly.



With Rotating Priority in a single chip DMA system, any device requesting service is guaranteed to be recognized after no more than three higher priority services have occurred. This prevents any one channel from monopolizing the system.

Compressed Timing — In order to achieve even greater throughput where system characteristics permit, the 82C37A-5 can compress the transfer time to two clock cycles. From Figure 11 it can be seen that state S3 is used to extend the access time of the read pulse. By removing state S3, the read pulse width is made equal to the write pulse width and a transfer consists only of state S2 to change the address and state S4 to perform the read/write. S1 states will still occur when A8–A15 need updating (see Address Generation). Timing for compressed transfers is found in Figure 14.

Address Generation — In order to reduce pin count, the 82C37A-5 multiplexes the eight higher order address bits on the data lines. State S1 is used to output the higher order address bits to an external latch from which they may be placed on the address bus. The falling edge of Address Strobe (ADSTB) is used to load these bits from the data lines to the latch. Address Enable (AEN) is used to enable the bits onto the address bus through a three-state enable. The lower order address bits are output by the 82C37A-5 directly. Lines A0–A7 should be connected to the address bus. Figure 11 shows the time relationships between CLK, AEN, ADSTB, DB0–DB7 and A0–A7.

During Block and Demand Transfer mode services, which include multiple transfers, the addresses generated will be sequential. For many transfers the data held in the external address latch will remain the same. This data need only change when a carry or borrow from A7 to A8 takes place in the normal sequence of addresses. To save time and speed transfers, the 82C37A-5 executes S1 states only when updating of A8–A15 in the latch is necessary. This means for long services, S1 states and Address Strobes may occur only once every 256 transfers, a savings of 255 clock cycles for each 256 transfers.

REGISTER DESCRIPTION

Current Address Register — Each channel has a 16-bit Current Address register. This register holds

the value of the address used during DMA transfers. The address is automatically incremented or decremented after each transfer and the intermediate values of the address are stored in the Current Address register during the transfer. This register is written or read by the microprocessor in successive 8-bit bytes. It may also be reinitialized by an Autoinitialize back to its original value. Autoinitialize takes place only after an $\overline{EOP}$.

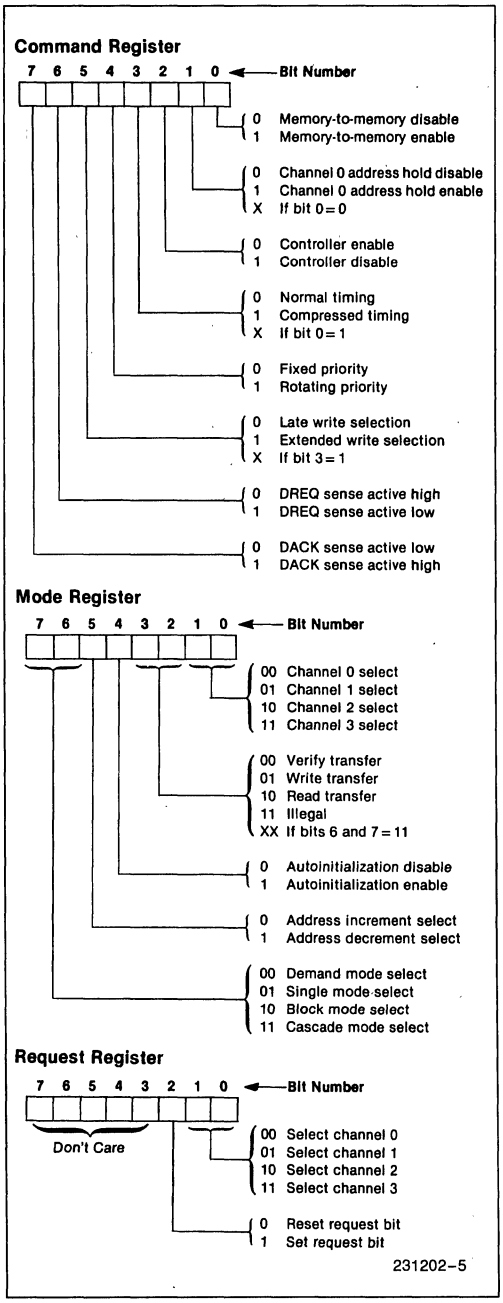
Current Word Register — Each channel has a 16-bit Current Word Count register. This register determines the number of transfers to be performed. The actual number of transfers will be one more than the number programmed in the Current Word Count register (i.e., programming a count of 100 will result in 101 transfers). The word count is decremented after each transfer. The intermediate value of the word count is stored in the register during the transfer. When the value in the register goes from zero to FFFFH, a TC will be generated. This register is loaded or read in successive 8-bit bytes by the microprocessor in the Program Condition. Following the end of a DMA service it may also be reinitialized by an Autoinitialization back to its original value. Autoinitialize can occur only when an $\overline{EOP}$ occurs. If it is not Autoinitialized, this register will have a count of FFFFH after TC.

Base Address and Base Word Count Registers — Each channel has a pair of Base Address and Base Word Count registers. These 16-bit registers store the original value of their associated current registers. During Autoinitialize these values are used to restore the current registers to their original values. The base registers are written simultaneously with their corresponding current register in 8-bit bytes in the Program Condition by the microprocessor. These registers cannot be read by the microprocessor.

Command Register — This 8-bit register controls the operation of the 82C37A-5. It is programmed by the microprocessor in the Program Condition and is cleared by Reset or a Master Clear instruction. The following table lists the function of the command bits. See Figure 6 for address coding.

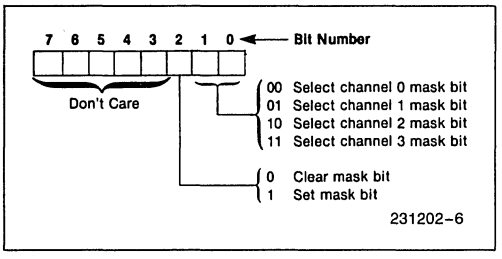
Mode Register — Each channel has a 6-bit Mode register associated with it. When the register is being written to by the microprocessor in the Program Condition, bits 0 and 1 determine which channel Mode register is to be written.

Request Register — The 82C37A-5 can respond to requests for DMA service which are initiated by software as well as by a DREQ. Each channel has a request bit associated with it in the 4-bit Request register. These are non-maskable and subject to prioritization by the Priority Encoder network. Each register bit is set or reset separately under software control or is cleared upon generation of a TC or external $\overline{EOP}$. The entire register is cleared by a Reset. To set or reset a bit, the software loads the proper form of the data word. See Figure 5 for register ad-

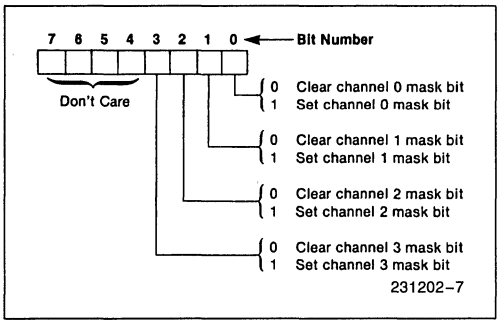


dress coding. In order to make a software request, the channel must be in Block Mode.

Mask Register — Each channel has associated with it a mask bit which can be set to disable the incoming DREQ. Each mask bit is set when its associated channel produces an $\overline{EOP}$ if the channel is not programmed for Autoinitialize. Each bit of the 4-bit Mask register may also be set or cleared separately under software control. The entire register is also set by a Reset. This disables all DMA requests until a clear Mask register instruction allows them to occur. The instruction to separately set or clear the mask bits is similar in form to that used with the Request register. See Figure 5 for instruction addressing.



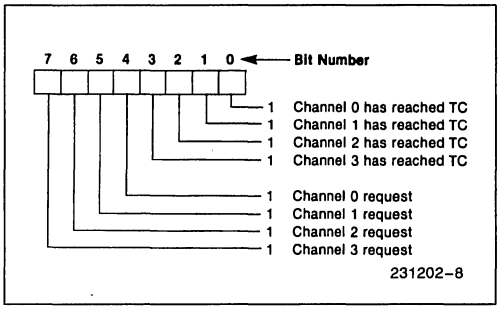
All four bits of the Mask register may also be written with a single command.



Register	Operation	Signals						
		$\overline{CS}$	$\overline{IOR}$	$\overline{IOW}$	A3	A2	A1	A0
Command	Write	0	1	0	1	0	0	0
Mode	Write	0	1	0	1	0	1	1
Request	Write	0	1	0	1	0	0	1
Mask	Set/Reset	0	1	0	1	0	1	0
Mask	Write	0	1	0	1	1	1	1
Temporary	Read	0	0	1	1	1	0	1
Status	Read	0	0	1	1	0	0	0

Figure 5. Definition of Register Codes

Status Register — The Status register is available to be read out of the 82C37A-5 by the microprocessor. It contains information about the status of the devices at this point. This information includes which channels have reached a terminal count and which channels have pending DMA requests. Bits 0–3 are set every time a TC is reached by that channel or an external $\overline{EOP}$ is applied. These bits are cleared upon Reset and on each Status Read. Bits 4–7 are set whenever their corresponding channel is requesting service.



Temporary Register — The Temporary register is used to hold data during memory-to-memory transfers. Following the completion of the transfers, the last word moved can be read by the microprocessor in the Program Condition. The Temporary register always contains the last byte transferred in the previous memory-to-memory operation, unless cleared by a Reset.

Software Commands — These are additional special software commands which can be executed in the Program Condition. They do not depend on any specific bit pattern on the data bus. The three software commands are:

Clear First/Last Flip-Flop: This command is executed prior to writing or reading new address or word count information to the 82C37A-5. This initializes the flip-flop to a known state so that subsequent accesses to register contents by the microprocessor will address upper and lower bytes in the correct sequence.

Master Clear: This software instruction has the same effect as the hardware Reset. The Command, Status, Request, Temporary, and Internal First/Last Flip-Flop registers are cleared and the Mask register is set. The 82C37A-5 will enter the Idle cycle.

Clear Mask Register: This command clears the mask bits of all four channels, enabling them to accept DMA requests.

5

Figure 6 lists the address codes for the software commands:

Signals						Operation
A3	A2	A1	A0	$\overline{IOR}$	$\overline{IOW}$	
1	0	0	0	0	1	Read Status Register
1	0	0	0	1	0	Write Command Register
1	0	0	1	0	1	Illegal
1	0	0	1	1	0	Write Request Register
1	0	1	0	0	1	Illegal
1	0	1	0	1	0	Write Single Mask Register Bit
1	0	1	1	0	1	Illegal
1	0	1	1	1	0	Write Mode Register
1	1	0	0	0	1	Illegal
1	1	0	0	1	0	Clear Byte Pointer Flip-Flop
1	1	0	1	0	1	Read Temporary Register
1	1	0	1	1	0	Master Clear
1	1	1	0	0	1	Illegal
1	1	1	0	1	0	Clear Mask Register
1	1	1	1	0	1	Illegal
1	1	1	1	1	0	Write All Mask Register Bits

Figure 6. Software Command Codes

PROGRAMMING

The 82C37A-5 will accept programming from the host processor any time that HLDA is inactive; this is true even if HRQ is active. The responsibility of the host is to assure that programming and HLDA are mutually exclusive. Note that a problem can occur if a DMA request occurs, on an unmasked channel while the 82C37A-5 is being programmed. For instance, the CPU may be starting to reprogram the two byte Address register of channel 1 when channel 1 receives a DMA request. If the 82C37A-5 is enabled (bit 2 in the command register is 0) and channel 1 is unmasked, a DMA service will occur after only one byte of the Address register has been reprogrammed. This can be avoided by disabling the controller (setting bit 2 in the command register) or masking the channel before programming any other registers. Once the programming is complete, the controller can be enabled/unmasked.

Channel	Register	Operation	Signals							Internal Flip-Flop	Data Bus DB0–DB7
			$\overline{CS}$	$\overline{IOR}$	$\overline{IOW}$	A3	A2	A1	A0		
0	Base and Current Address	Write	0	1	0	0	0	0	0	0	A0–A7
			0	1	0	0	0	0	0	1	A8–A15
	Current Address	Read	0	0	1	0	0	0	0	0	A0–A7
			0	0	1	0	0	0	0	1	A8–A15
	Base and Current Word Count	Write	0	1	0	0	0	0	1	0	W0–W7
			0	1	0	0	0	0	1	1	W8–W15
	Current Word Count	Read	0	0	1	0	0	0	1	0	W0–W7
			0	0	1	0	0	0	1	1	W8–W15
1	Base and Current Address	Write	0	1	0	0	0	1	0	0	A0–A7
			0	1	0	0	0	1	0	1	A8–A15
	Current Address	Read	0	0	1	0	0	1	0	0	A0–A7
			0	0	1	0	0	1	0	1	A8–A15
	Base and Current Word Count	Write	0	1	0	0	0	1	1	0	W0–W7
			0	1	0	0	0	1	1	1	W8–W15
	Current Word Count	Read	0	0	1	0	0	1	1	0	W0–W7
			0	0	1	0	0	1	1	1	W8–W15
2	Base and Current Address	Write	0	1	0	0	1	0	0	0	A0–A7
			0	1	0	0	1	0	0	1	A8–A15
	Current Address	Read	0	0	1	0	1	0	0	0	A0–A7
			0	0	1	0	1	0	0	1	A8–A15
	Base and Current Word Count	Write	0	1	0	0	1	0	1	0	W0–W7
			0	1	0	0	1	0	1	1	W8–W15
	Current Word Count	Read	0	0	1	0	1	0	1	0	W0–W7
			0	0	1	0	1	0	1	1	W8–W15
3	Base and Current Address	Write	0	1	0	0	1	1	0	0	A0–A7
			0	1	0	0	1	1	0	1	A8–A15
	Current Address	Read	0	0	1	0	1	1	0	0	A0–A7
			0	0	1	0	1	1	0	1	A8–A15
	Base and Current Word Count	Write	0	1	0	0	1	1	1	0	W0–W7
			0	1	0	0	1	1	1	1	W8–W15
	Current Word Count	Read	0	0	1	0	1	1	1	0	W0–W7
			0	0	1	0	1	1	1	1	W8–W15

Figure 7. Word Count and Address Register Command Codes

After power-up it is suggested that all internal locations, especially the Mode registers, be loaded with some valid value. This should be done even if some channels are unused.

APPLICATION INFORMATION

Figure 8 shows a convenient method for configuring a DMA system with the 82C37A-5 controller and an 8080A/8085AH microprocessor system. The multimode DMA controller issues a HRQ to the processor whenever there is at least one valid DMA request from a peripheral device. When the processor replies with a HLDA signal, the 82C37A-5 takes control of the address bus, the data bus and the control bus. The address for the first transfer operation comes out in two bytes — the least significant 8 bits on the eight address outputs and the most significant 8 bits on the data bus. The contents of the data bus are then latched into the 8-bit latch to complete the full 16 bits of the address bus. After the initial transfer takes place, the latch is updated only after a carry or borrow is generated in the least significant address byte. Four DMA channels are provided when one 82C37A-5 is used.

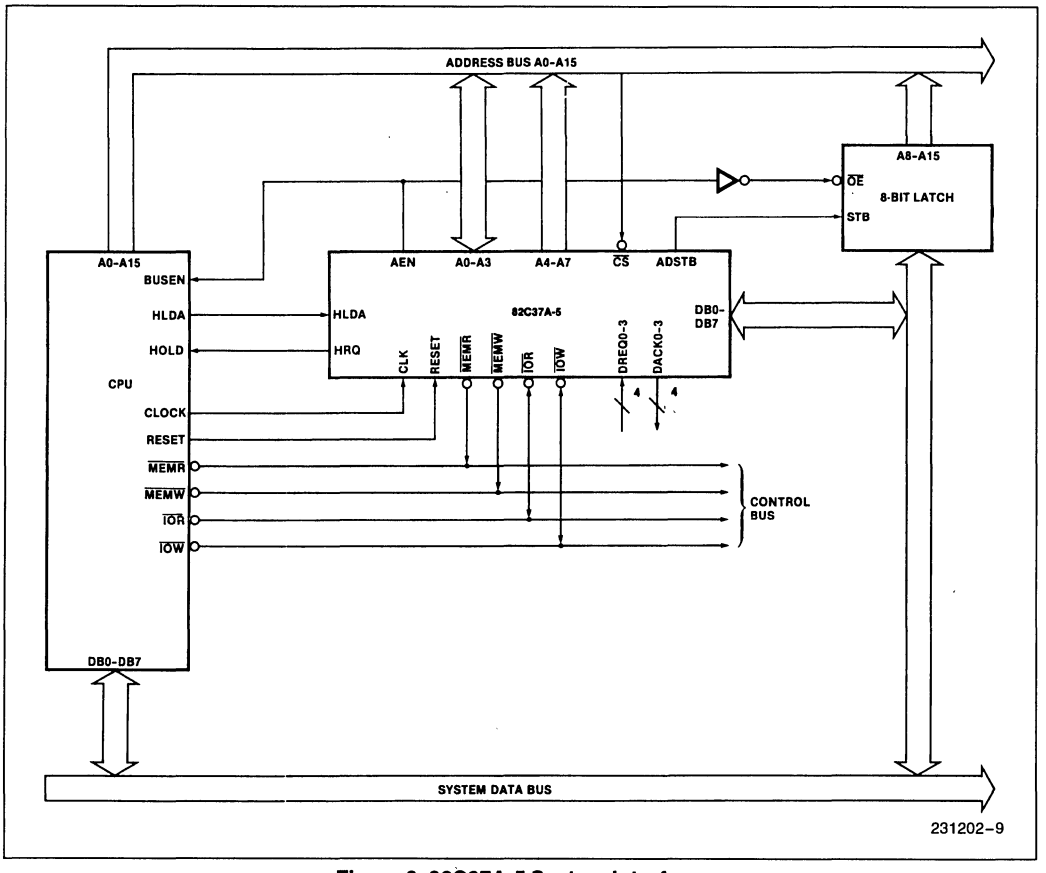


Figure 8. 82C37A-5 System Interface

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature under Bias .....0°C to 70°C
Case Temperature .....0°C to +75°C
Storage Temperature .....−55°C to +150°C
Voltage on Any Pin with Respect to Ground.....−0.5V to +7V
Power Dissipation.....1.0 Watt

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**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.*

D.C. CHARACTERISTICS

T_A = 0°C to 70°C, T_{CASE} = 0°C to 75°C, V_{CC} = +5.0V ±5%, GND = 0V

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
V_{OH}	Output High Voltage	3.7			V	I_{OH} = −1.0 mA
V_{OL}	Output LOW Voltage			0.40	V	I_{OL} = 3.2 mA
V_{IH}	Input HIGH Voltage	2.2		V_{CC} + 0.5	V	
V_{IL}	Input LOW Voltage	−0.5		0.8	V	
I_{LI}	Input Load Current			±10	μA	0V ≤ V_{IN} ≤ V_{CC}
I_{LO}	Output Leakage Current			±10	μA	0V ≤ V_{OUT} ≤ V_{CC}
I_{CC}	V_{CC} Supply Current			10	mA	(Note 1)
I_{CCS}	Standby Supply Current			10	μA	HLDA = 0V, V_{IL} = 0V, V_{IH} = V_{CC}
C_O	Output Capacitance		4	8	pF	fc = 1.0 MHz, Inputs = 0V
C_I	Input Capacitance		8	15	pF	
C_{IO}	I/O Capacitance		10	18	pF	

A.C. CHARACTERISTICS—DMA (MASTER) MODE

T_A = 0°C to 70°C, T_{CASE} = 0°C to 75°C, V_{CC} = +5V ±5%, GND = 0V

Symbol	Parameter	Min	Max	Unit
TAEL	AEN HIGH from CLK LOW (S1) Delay Time		200	ns
TAET	AEN LOW from CLK HIGH (SI) Delay Time		130	ns
TAFAB	ADR Active to Float Delay from CLK HIGH		90	ns
TAFC	$\overline{READ}$ or $\overline{WRITE}$ Float from CLK HIGH		120	ns
TAFDB	DB Active to Float Delay from CLK HIGH		170	ns
TAHR	ADR from $\overline{READ}$ HIGH Hold Time	TCY-100		ns
TAHS	DB from ADSTB LOW Hold Time	30		ns
TAHW	ADR from $\overline{WRITE}$ HIGH Hold Time	TCY-50		ns
TAK	DACK Valid from CLK LOW Delay Time (Note 3)		170	ns
	$\overline{EOP}$ HIGH from CLK HIGH Delay Time (Note 4)		170	ns
	$\overline{EOP}$ LOW from CLK HIGH Delay Time		170	ns
TASM	ADR Stable from CLK HIGH		170	ns
TASS	DB to ADSTB LOW Setup Time	100		ns
TCH	Clock High Time (Transitions ≤ 10 ns)	68		ns
TCL	Clock LOW Time (Transitions ≤ 10 ns) Auto Initialize Enabled	115		ns
TCY	CLK Cycle Time	200		ns
TDCL	CLK HIGH to $\overline{READ}$ or $\overline{WRITE}$ LOW Delay (Note 2)		190	ns
TDCTR	$\overline{READ}$ HIGH from CLK HIGH (S4) Delay Time (Note 2)		190	ns
TDCTW	$\overline{WRITE}$ HIGH from CLK HIGH (S4) Delay Time (Note 2)		130	ns
TDQ1	HRQ Valid from CLK HIGH Delay Time		120	ns
TEPS	$\overline{EOP}$ LOW from CLK LOW Setup Time	40		ns
TEPW	$\overline{EOP}$ Pulse Width	220		ns
TFAAB	ADR Float to Active Delay from CLK HIGH		170	ns
TFAC	$\overline{READ}$ or $\overline{WRITE}$ Active from CLK HIGH		150	ns
TFADB	DB Float to Active Delay from CLK HIGH		200	ns
THS	HLDA Valid to CLK HIGH Setup Time	75		ns
TIDH	Input Data from $\overline{MEMR}$ HIGH Hold Time	0		ns
TIDS	Input Data to $\overline{MEMR}$ HIGH Setup Time	170		ns
TODH	Output Data from $\overline{MEMW}$ HIGH Hold Time	10		ns
TODV	Output Data Valid to $\overline{MEMW}$ HIGH	125		ns
TQS	DREQ to CLK LOW (SI, S4) Setup Time (Note 3)	0		ns
TRH	CLK to READY LOW Hold Time	20		ns
TRS	READY to CLK LOW Setup Time	60		ns
TSTL	ADSTB HIGH from CLK HIGH Delay Time		130	ns
TSTT	ADSTB LOW from CLK HIGH Delay Time		90	ns

5

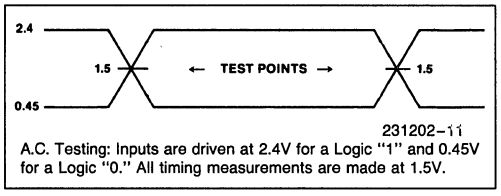
A.C. CHARACTERISTICS—PERIPHERAL (SLAVE) MODE

T_A = 0°C to 70°C, T_{CASE} = 0°C to 75°C, V_{CC} = +5V ±5%, GND = 0V

Symbol	Parameter	Min	Max	Unit
TAR	ADR Valid or $\overline{CS}$ LOW to $\overline{READ}$ LOW	50		ns
TAW	ADR Valid to $\overline{WRITE}$ HIGH Setup Time	130		ns
TCW	CS LOW to $\overline{WRITE}$ HIGH Setup Time	130		ns
TDW	Data Valid to $\overline{WRITE}$ HIGH Setup Time	130		ns
TRA	ADR or CS Hold from $\overline{READ}$ HIGH	0		ns
TRDE	Data Access from $\overline{READ}$ LOW		140	ns
TRDF	DB Float Delay from $\overline{READ}$ HIGH	0	70	ns
TRSTD	Power Supply HIGH to RESET LOW Setup Time	500		ns
TRSTS	RESET to First $\overline{IOWR}$	2TCY		ns
TRSTW	RESET Pulse Width	300		ns
TRW	$\overline{READ}$ Width	200		ns
TWA	ADR from $\overline{WRITE}$ HIGH Hold Time	20		ns
TWC	CS HIGH from $\overline{WRITE}$ HIGH Hold Time	20		ns
TWD	Data from $\overline{WRITE}$ HIGH Hold Time	30		ns
TWWS	Write Width	160		ns

NOTES:
1. Input frequency 5 MHz, when RESET, V_{IN} = 0V/V_{CC}, C_L = 0 pF.
2. The net $\overline{IOW}$ or $\overline{MEMW}$ Pulse width for normal write will be TCY-100 ns and for extended write will be 2TCY-100 ns. The net $\overline{IOR}$ or $\overline{MEMR}$ pulse width for normal read will be 2TCY-50 ns and for compressed read will be TCY-50 ns.
3. DREQ and DACK signals may be active high or active low. Timing diagrams assume the active high mode for DREQ and active low for DACK.
4. $\overline{EOP}$ is an open collector output. This parameter assumes the presence of a 2.2K pullup to V_{CC}.

A.C. TESTING INPUT/OUTPUT WAVEFORM



A.C. Testing: Inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0." All timing measurements are made at 1.5V.

WAVEFORMS

SLAVE MODE WRITE TIMING

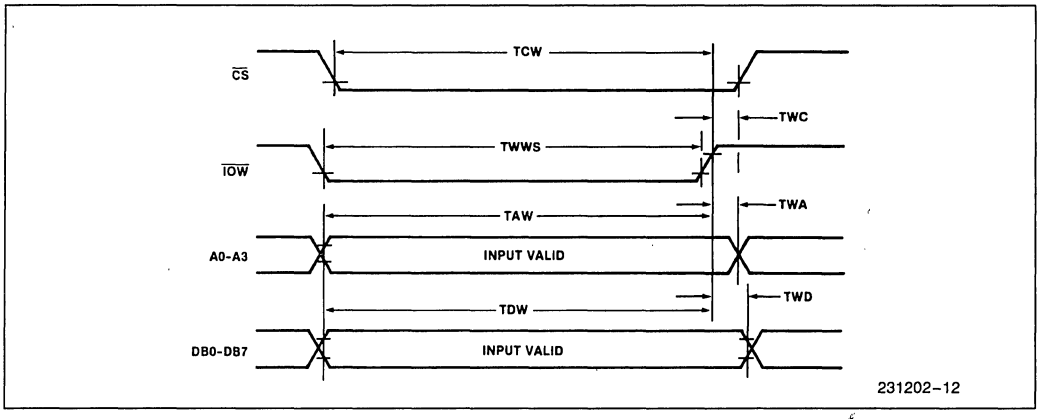


Figure 9. Slave Mode Write

SLAVE MODE READ TIMING

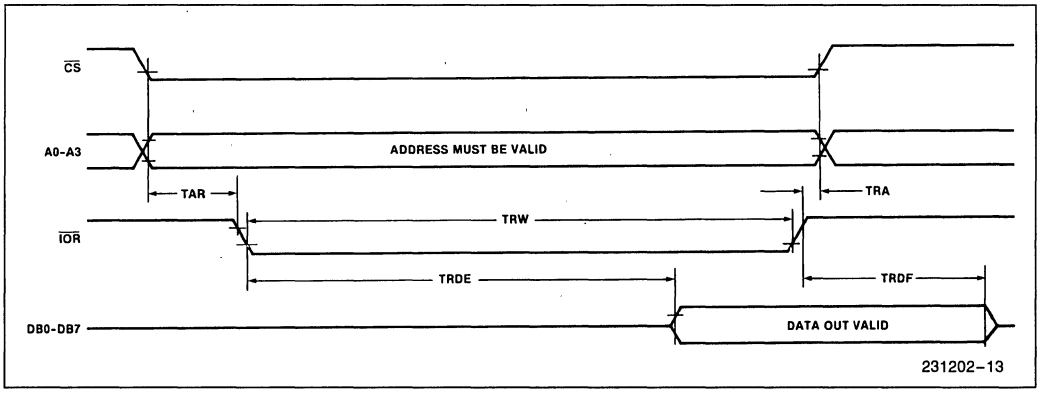


Figure 10. Slave Mode Read

5

WAVEFORMS (Continued)

DMA TRANSFER TIMING

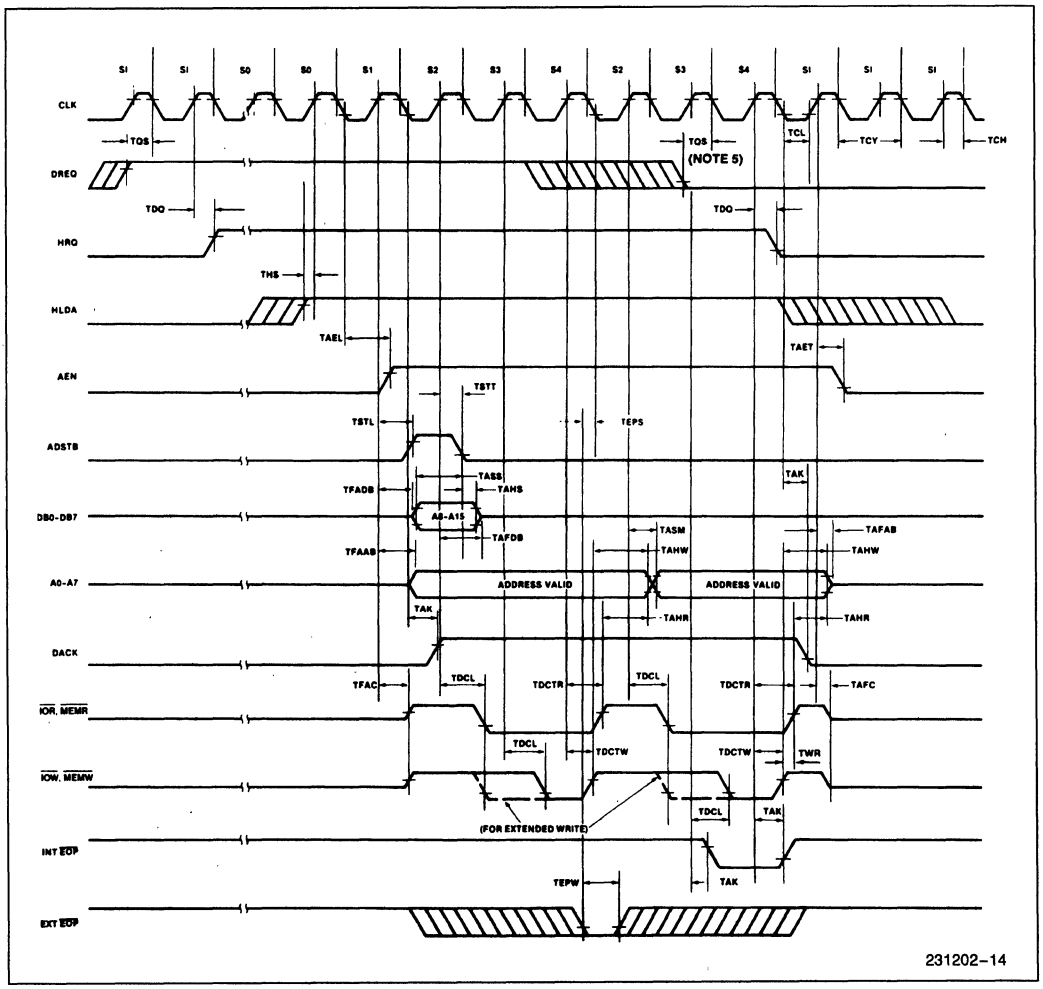


Figure 11. DMA Transfer

WAVEFORMS (Continued)

MEMORY-TO-MEMORY TRANSFER TIMING

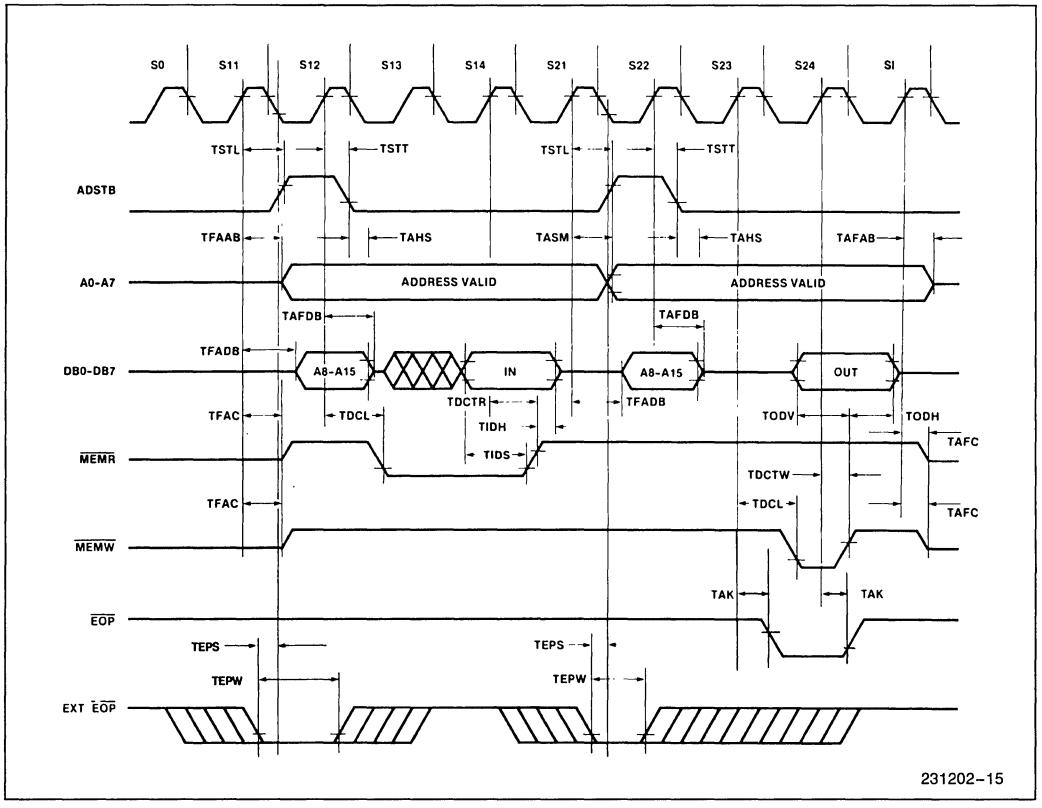


Figure 12. Memory-to-Memory Transfer

READY TIMING

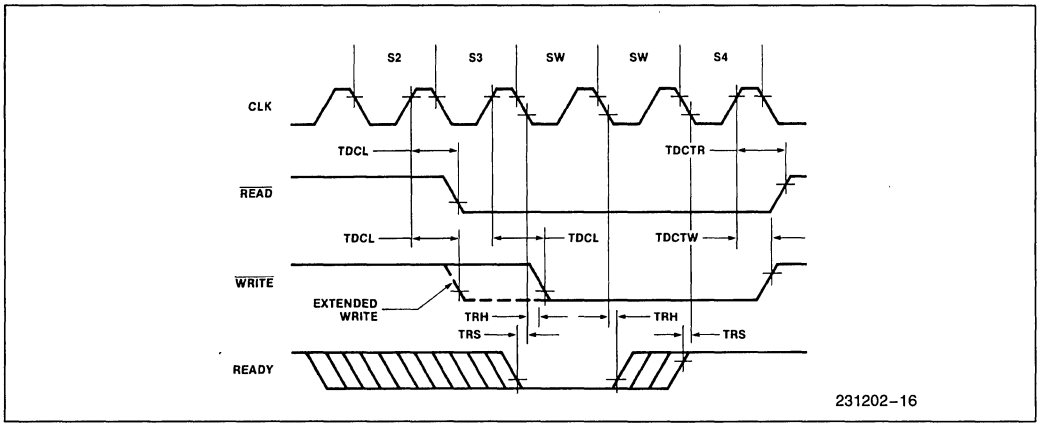


Figure 13. Ready

WAVEFORMS (Continued)

COMPRESSED TRANSFER TIMING

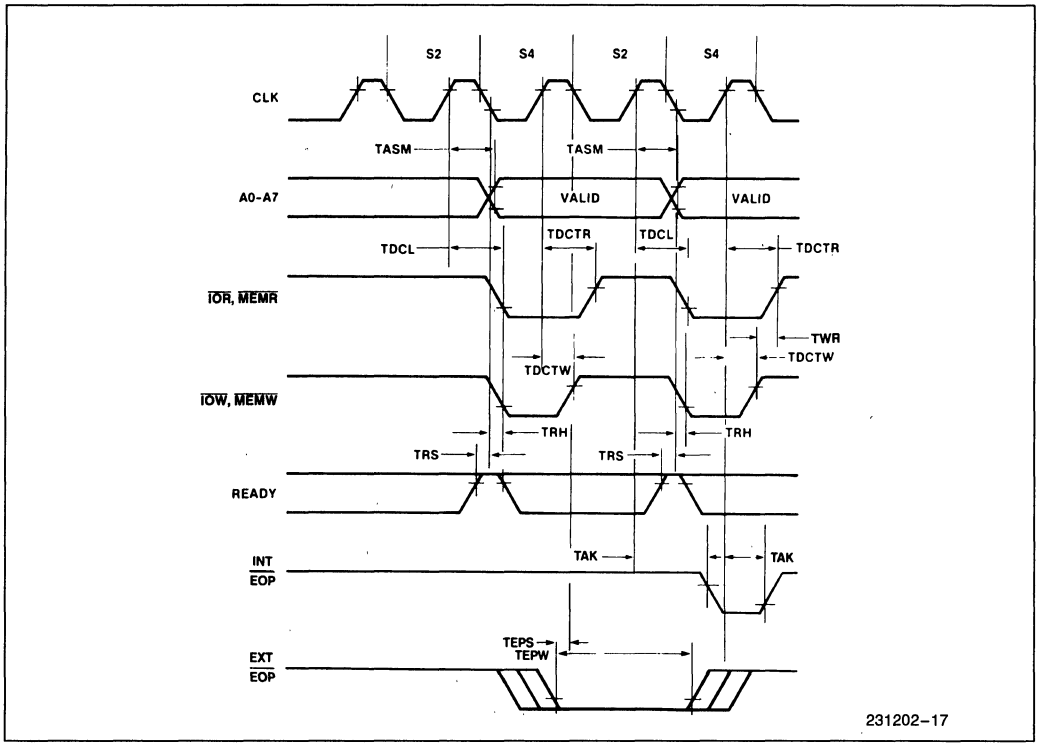


Figure 14. Compressed Transfer

RESET TIMING

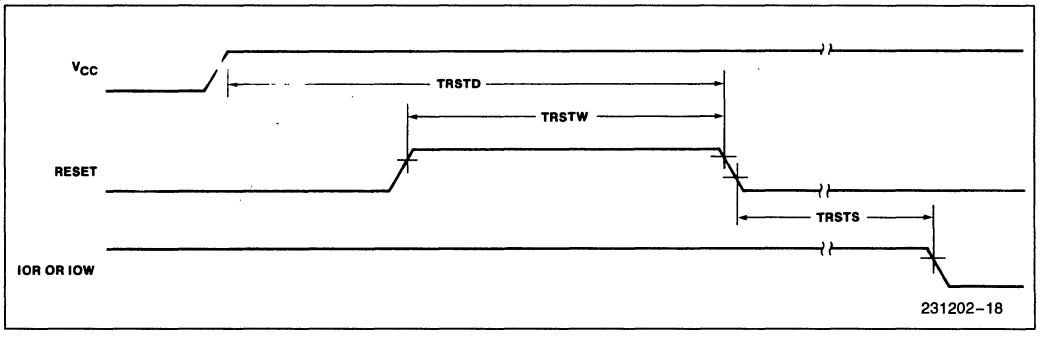


Figure 15. Reset

DATA SHEET REVISION REVIEW

The following list represents key differences between this and the -004 data sheet. Please review this summary carefully.

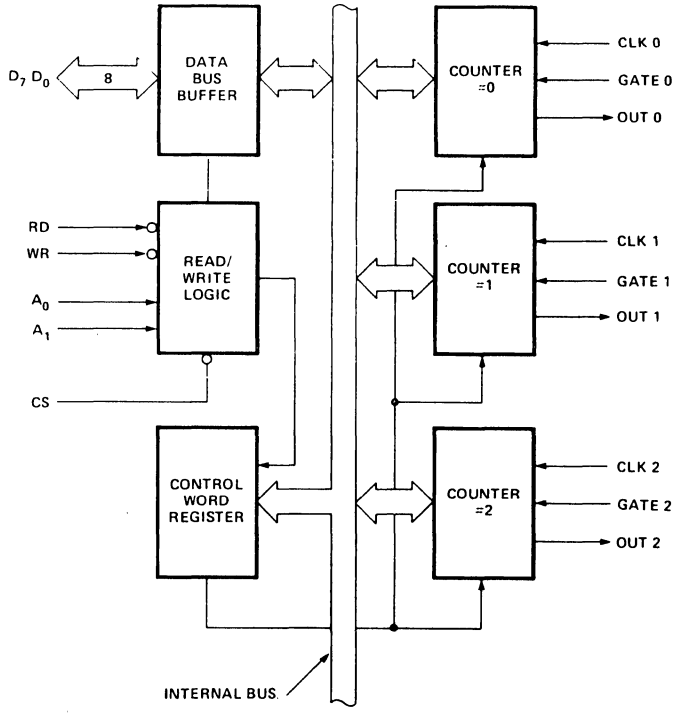
1. The "PRELIMINARY" markings have been removed from the data sheet. The 82C37A-5 is no longer a preliminary part.
2. A section of the Functional Description describing 82C37A-5 operation with the 8085 CPU has been deleted.

8253/8253-5
PROGRAMMABLE INTERVAL TIMER

- **MCS-85 Compatible 8253-5**
- **3 Independent 16-Bit Counters**
- **DC to 2.6 MHz**
- **Programmable Counter Modes**
- **Count Binary or BCD**
- **Single +5V Supply**
- **Available in EXPRESS**
  - **Standard Temperature Range**
  - **Extended Temperature Range**

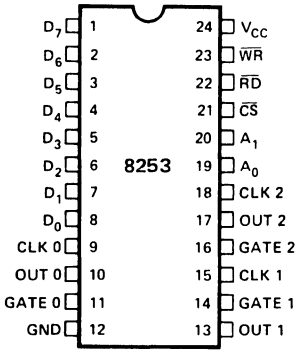
The Intel 8253 is a programmable counter/timer device designed for use as an Intel microcomputer peripheral. It uses NMOS technology with a single +5V supply and is packaged in a 24-pin plastic DIP.

It is organized as 3 independent 16-bit counters, each with a count rate of up to 2.6 MHz. All modes of operation are software programmable.



231306-1

Figure 1. Block Diagram



231306-2

Figure 2. Pin Configuration

5

The complete document for this product is available from Intel's Literature Center at 1-800-548-4725.

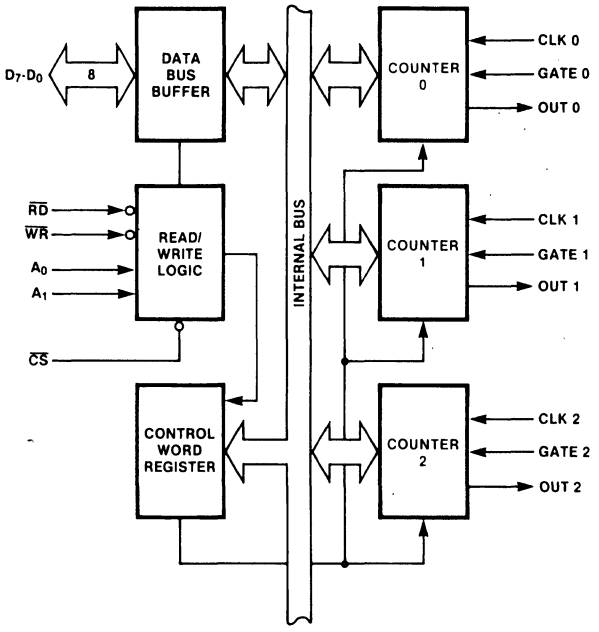
November 1986
Order Number: 231306-001

8254
PROGRAMMABLE INTERVAL TIMER

- **Compatible with All Intel and Most Other Microprocessors**
- **Handles Inputs from DC to 10 MHz**
  - **— 8 MHz 8254**
  - **— 10 MHz 8254-2**
- **Status Read-Back Command**
- **Six Programmable Counter Modes**
- **Three Independent 16-Bit Counters**
- **Binary or BCD Counting**
- **Single +5V Supply**
- **Available in EXPRESS**
  - **— Standard Temperature Range**

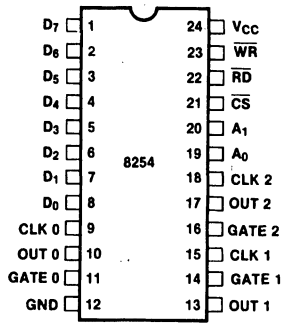
The Intel 8254 is a counter/timer device designed to solve the common timing control problems in microcomputer system design. It provides three independent 16-bit counters, each capable of handling clock inputs up to 10 MHz. All modes are software programmable. The 8254 is a superset of the 8253.

The 8254 uses HMOS technology and comes in a 24-pin plastic or CERDIP package.



231164-1

Figure 1. 8254 Block Diagram



231164-2

Figure 2. Pin Configuration

The complete document for this product is available from Intel's Literature Center at 1-800-548-4725.

September 1993
Order Number: 231164-005

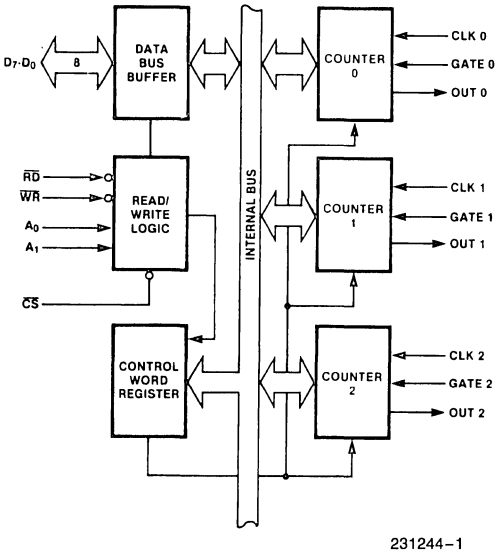
82C54
CHMOS PROGRAMMABLE INTERVAL TIMER

- **Compatible with all Intel and most other microprocessors**
- **High Speed, "Zero Wait State" Operation with 8 MHz 8086/88 and 80186/188**
- **Handles Inputs from DC — 10 MHz for 82C54-2**
- **Available in EXPRESS — Standard Temperature Range — Extended Temperature Range**
- **Three independent 16-bit counters**
- **Low Power CHMOS — I_{CC} = 10 mA @ 8 MHz Count frequency**
- **Completely TTL Compatible**
- **Six Programmable Counter Modes**
- **Binary or BCD counting**
- **Status Read Back Command**
- **Available in 24-Pin DIP and 28-Pin PLCC**

The Intel 82C54 is a high-performance, CHMOS version of the industry standard 8254 counter/timer which is designed to solve the timing control problems common in microcomputer system design. It provides three independent 16-bit counters, each capable of handling clock inputs up to 10 MHz. All modes are software programmable. The 82C54 is pin compatible with the HMOS 8254, and is a superset of the 8253.

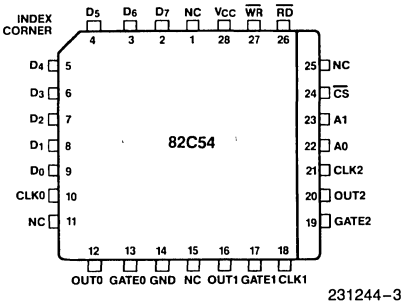
Six programmable timer modes allow the 82C54 to be used as an event counter, elapsed time indicator, programmable one-shot, and in many other applications.

The 82C54 is fabricated on Intel's advanced CHMOS III technology which provides low power consumption with performance equal to or greater than the equivalent HMOS product. The 82C54 is available in 24-pin DIP and 28-pin plastic leaded chip carrier (PLCC) packages.



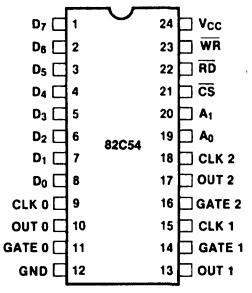
231244-1

Figure 1. 82C54 Block Diagram



231244-3

PLASTIC LEADED CHIP CARRIER



231244-2

Diagrams are for pin reference only.
Package sizes are not to scale.

Figure 2. 82C54 Pinout

Table 1. Pin Description

Symbol	Pin Number		Type	Function
	DIP	PLCC		
D_7-D_0	1-8	2-9	I/O	Data: Bidirectional tri-state data bus lines, connected to system data bus.
CLK 0	9	10	I	Clock 0: Clock input of Counter 0.
OUT 0	10	12	O	Output 0: Output of Counter 0.
GATE 0	11	13	I	Gate 0: Gate input of Counter 0.
GND	12	14		Ground: Power supply connection.
OUT 1	13	16	O	Out 1: Output of Counter 1.
GATE 1	14	17	I	Gate 1: Gate input of Counter 1.
CLK 1	15	18	I	Clock 1: Clock input of Counter 1.
GATE 2	16	19	I	Gate 2: Gate input of Counter 2.
OUT 2	17	20	O	Out 2: Output of Counter 2.
CLK 2	18	21	I	Clock 2: Clock input of Counter 2.
A_1, A_0	20-19	23-22	I	Address: Used to select one of the three Counters or the Control Word Register for read or write operations. Normally connected to the system address bus.
				A_1 / A_0 / Selects 0 / 0 / Counter 0 0 / 1 / Counter 1 1 / 0 / Counter 2 1 / 1 / Control Word Register
$\overline{CS}$	21	24	I	Chip Select: A low on this input enables the 82C54 to respond to $\overline{RD}$ and $\overline{WR}$ signals. $\overline{RD}$ and $\overline{WR}$ are ignored otherwise.
$\overline{RD}$	22	26	I	Read Control: This input is low during CPU read operations.
$\overline{WR}$	23	27	I	Write Control: This input is low during CPU write operations.
V_{CC}	24	28		Power: +5V power supply connection.
NC		1, 11, 15, 25		No Connect

FUNCTIONAL DESCRIPTION

General

The 82C54 is a programmable interval timer/counter designed for use with Intel microcomputer systems. It is a general purpose, multi-timing element that can be treated as an array of I/O ports in the system software.

The 82C54 solves one of the most common problems in any microcomputer system, the generation of accurate time delays under software control. Instead of setting up timing loops in software, the programmer configures the 82C54 to match his requirements and programs one of the counters for the desired delay. After the desired delay, the 82C54 will interrupt the CPU. Software overhead is minimal and variable length delays can easily be accommodated.

Some of the other counter/timer functions common to microcomputers which can be implemented with the 82C54 are:

- Real time clock
- Even counter
- Digital one-shot
- Programmable rate generator
- Square wave generator
- Binary rate multiplier
- Complex waveform generator
- Complex motor controller

Block Diagram

DATA BUS BUFFER

This 3-state, bi-directional, 8-bit buffer is used to interface the 82C54 to the system bus (see Figure 3).

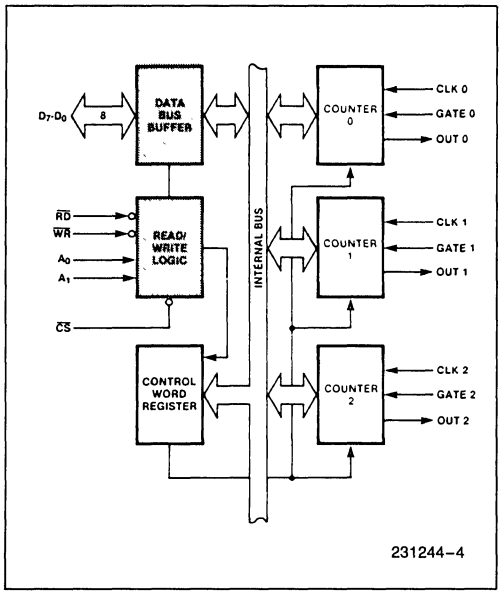


Figure 3. Block Diagram Showing Data Bus Buffer and Read/Write Logic Functions

READ/WRITE LOGIC

The Read/Write Logic accepts inputs from the system bus and generates control signals for the other functional blocks of the 82C54. A_1 and A_0 select one of the three counters or the Control Word Register to be read from/written into. A "low" on the $\overline{RD}$ input tells the 82C54 that the CPU is reading one of the counters. A "low" on the $\overline{WR}$ input tells the 82C54 that the CPU is writing either a Control Word or an initial count. Both $\overline{RD}$ and $\overline{WR}$ are qualified by $\overline{CS}$; $\overline{RD}$ and $\overline{WR}$ are ignored unless the 82C54 has been selected by holding $\overline{CS}$ low.

The WR# and CLK signals should be synchronous. This is accomplished by using a CLK input signal to the 82C54 counters which is a derivative of the system clock source. Another technique is to externally synchronize the WR# and CLK input signals. This is done by gating WR# with CLK.

CONTROL WORD REGISTER

The Control Word Register (see Figure 4) is selected by the Read/Write Logic when A_1, A_0 = 11. If the CPU then does a write operation to the 82C54, the data is stored in the Control Word Register and is interpreted as a Control Word used to define the operation of the Counters.

The Control Word Register can only be written to; status information is available with the Read-Back Command.

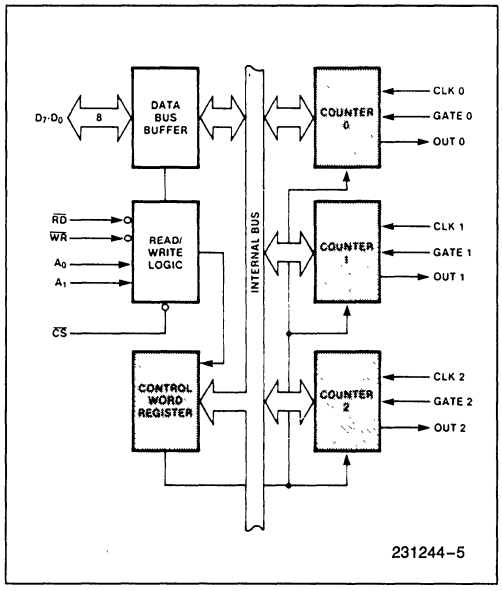


Figure 4. Block Diagram Showing Control Word Register and Counter Functions

COUNTER 0, COUNTER 1, COUNTER 2

These three functional blocks are identical in operation, so only a single Counter will be described. The internal block diagram of a single counter is shown in Figure 5.

The Counters are fully independent. Each Counter may operate in a different Mode.

The Control Word Register is shown in the figure; it is not part of the Counter itself, but its contents determine how the Counter operates.

5

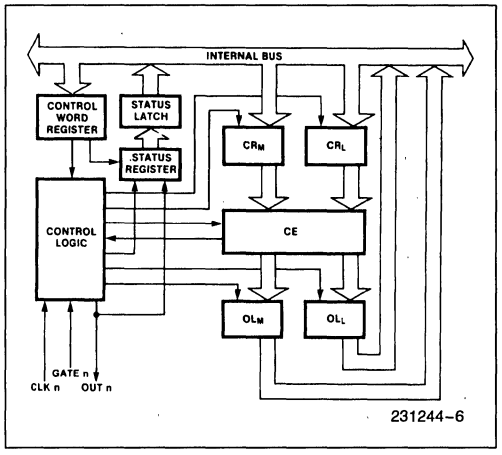


Figure 5. Internal Block Diagram of a Counter

The status register, shown in the Figure, when latched, contains the current contents of the Control Word Register and status of the output and null count flag. (See detailed explanation of the Read-Back command.)

The actual counter is labelled CE (for "Counting Element"). It is a 16-bit presettable synchronous down counter.

OL_M and OL_L are two 8-bit latches. OL stands for "Output Latch"; the subscripts M and L stand for "Most significant byte" and "Least significant byte" respectively. Both are normally referred to as one unit and called just OL. These latches normally "follow" the CE, but if a suitable Counter Latch Command is sent to the 82C54, the latches "latch" the present count until read by the CPU and then return to "following" the CE. One latch at a time is enabled by the counter's Control Logic to drive the internal bus. This is how the 16-bit Counter communicates over the 8-bit internal bus. Note that the CE itself cannot be read; whenever you read the count, it is the OL that is being read.

Similarly, there are two 8-bit registers called CR_M and CR_L (for "Count Register"). Both are normally referred to as one unit and called just CR. When a new count is written to the Counter, the count is stored in the CR and later transferred to the CE. The Control Logic allows one register at a time to be loaded from the internal bus. Both bytes are transferred to the CE simultaneously. CR_M and CR_L are cleared when the Counter is programmed. In this way, if the Counter has been programmed for one byte counts (either most significant byte only or least significant byte only) the other byte will be zero. Note that the CE cannot be written into; whenever a count is written, it is written into the CR.

The Control Logic is also shown in the diagram. CLK n, GATE n, and OUT n are all connected to the outside world through the Control Logic.

82C54 SYSTEM INTERFACE

The 82C54 is treated by the systems software as an array of peripheral I/O ports; three are counters and the fourth is a control register for MODE programming.

Basically, the select inputs A_0, A_1 connect to the A_0, A_1 address bus signals of the CPU. The $\overline{CS}$ can be derived directly from the address bus using a linear select method. Or it can be connected to the output of a decoder, such as an Intel 8205 for larger systems.

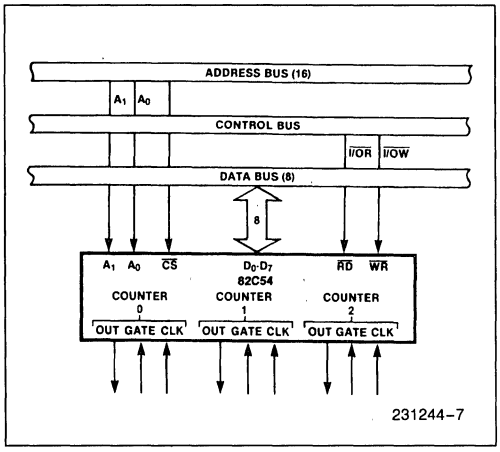


Figure 6. 82C54 System Interface

OPERATIONAL DESCRIPTION

General

After power-up, the state of the 82C54 is undefined. The Mode, count value, and output of all Counters are undefined.

How each Counter operates is determined when it is programmed. Each Counter must be programmed before it can be used. Unused counters need not be programmed.

Programming the 82C54

Counters are programmed by writing a Control Word and then an initial count. The control word format is shown in Figure 7.

All Control Words are written into the Control Word Register, which is selected when A_1, A_0 = 11. The Control Word itself specifies which Counter is being programmed.

By contrast, initial counts are written into the Counters, not the Control Word Register. The A_1, A_0 inputs are used to select the Counter to be written into. The format of the initial count is determined by the Control Word used.

Control Word Format

A_1, A_0 = 11 $\overline{CS}$ = 0 $\overline{RD}$ = 1 $\overline{WR}$ = 0

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

SC — Select Counter:

SC1	SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Read-Back Command (See Read Operations)

RW — Read/Write:

RW1	RW0	
0	0	Counter Latch Command (see Read Operations)
0	1	Read/Write least significant byte only.
1	0	Read/Write most significant byte only.
1	1	Read/Write least significant byte first, then most significant byte.

NOTE: Don't care bits (X) should be 0 to insure compatibility with future Intel products.

M — MODE:

M2	M1	M0	
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

BCD:

0	Binary Counter 16-bits
1	Binary Coded Decimal (BCD) Counter (4 Decades)

Figure 7. Control Word Format

Write Operations

The programming procedure for the 82C54 is very flexible. Only two conventions need to be remembered:

1) For each Counter, the Control Word must be written before the initial count is written.
2) The initial count must follow the count format specified in the Control Word (least significant byte only, most significant byte only, or least significant byte and then most significant byte).

Since the Control Word Register and the three Counters have separate addresses (selected by the A_1, A_0 inputs), and each Control Word specifies the Counter it applies to (SC0, SC1 bits), no special instruction sequence is required. Any programming sequence that follows the conventions above is acceptable.

A new initial count may be written to a Counter at any time without affecting the Counter's programmed Mode in any way. Counting will be affected as described in the Mode definitions. The new count must follow the programmed count format.

If a Counter is programmed to read/write two-byte counts, the following precaution applies: A program must not transfer control between writing the first and second byte to another routine which also writes into that same Counter. Otherwise, the Counter will be loaded with an incorrect count.

		A_1	A_0
Control Word —	Counter 0	1	1
LSB of count —	Counter 0	0	0
MSB of count —	Counter 0	0	0
Control Word —	Counter 1	1	1
LSB of count —	Counter 1	0	1
MSB of count —	Counter 1	0	1
Control Word —	Counter 2	1	1
LSB of count —	Counter 2	1	0
MSB of count —	Counter 2	1	0

		A_1	A_0
Control Word —	Counter 2	1	1
Control Word —	Counter 1	1	1
Control Word —	Counter 0	1	1
LSB of count —	Counter 2	1	0
MSB of count —	Counter 2	1	0
LSB of count —	Counter 1	0	1
MSB of count —	Counter 1	0	1
LSB of count —	Counter 0	0	0
MSB of count —	Counter 0	0	0

		A_1	A_0
Control Word —	Counter 0	1	1
Counter Word —	Counter 1	1	1
Control Word —	Counter 2	1	1
LSB of count —	Counter 2	1	0
LSB of count —	Counter 1	0	1
LSB of count —	Counter 0	0	0
MSB of count —	Counter 0	0	0
MSB of count —	Counter 1	0	1
MSB of count —	Counter 2	1	0

		A_1	A_0
Control Word —	Counter 1	1	1
Control Word —	Counter 0	1	1
LSB of count —	Counter 1	0	1
Control Word —	Counter 2	1	1
LSB of count —	Counter 0	0	0
MSB of count —	Counter 1	0	1
LSB of count —	Counter 2	1	0
MSB of count —	Counter 0	0	0
MSB of count —	Counter 2	1	0

NOTE:
In all four examples, all counters are programmed to read/write two-byte counts.
These are only four of many possible programming sequences.

Figure 8. A Few Possible Programming Sequences

Read Operations

It is often desirable to read the value of a Counter without disturbing the count in progress. This is easily done in the 82C54.

There are three possible methods for reading the counters: a simple read operation, the Counter Latch Command, and the Read-Back Command. Each is explained below. The first method is to perform a simple read operation. To read the Counter, which is selected with the A1, A0 inputs, the CLK input of the selected Counter must be inhibited by using either the GATE input or external logic. Otherwise, the count may be in the process of changing when it is read, giving an undefined result.

COUNTER LATCH COMMAND

The second method uses the "Counter Latch Command". Like a Control Word, this command is written to the Control Word Register, which is selected when A_1, A_0 = 11. Also like a Control Word, the SC0, SC1 bits select one of the three Counters, but two other bits, D5 and D4, distinguish this command from a Control Word.

A_1, A_0 = 11; $\overline{CS}$ = 0; $\overline{RD}$ = 1; $\overline{WR}$ = 0

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	0	0	X	X	X	X

SC1, SC0 - specify counter to be latched

SC1	SC0	Counter
0	0	0
0	1	1
1	0	2
1	1	Read-Back Command

D5,D4 - 00 designates Counter Latch Command

X - don't care

NOTE:
Don't care bits (X) should be 0 to insure compatibility with future Intel products.

Figure 9. Counter Latching Command Format

The selected Counter's output latch (OL) latches the count at the time the Counter Latch Command is received. This count is held in the latch until it is read by the CPU (or until the Counter is reprogrammed). The count is then unlatched automatically and the OL returns to "following" the counting element (CE). This allows reading the contents of the Counters "on the fly" without affecting counting in progress. Multiple Counter Latch Commands may be used to latch more than one Counter. Each latched Counter's OL holds its count until it is read. Counter Latch Commands do not affect the programmed Mode of the Counter in any way.

If a Counter is latched and then, some time later, latched again before the count is read, the second Counter Latch Command is ignored. The count read will be the count at the time the first Counter Latch Command was issued.

With either method, the count must be read according to the programmed format; specifically, if the Counter is programmed for two byte counts, two bytes must be read. The two bytes do not have to be read one right after the other; read or write or programming operations of other Counters may be inserted between them.

Another feature of the 82C54 is that reads and writes of the same Counter may be interleaved; for example, if the Counter is programmed for two byte counts, the following sequence is valid.

1. Read least significant byte.
2. Write new least significant byte.
3. Read most significant byte.
4. Write new most significant byte.

If a Counter is programmed to read/write two-byte counts, the following precaution applies; A program must not transfer control between reading the first and second byte to another routine which also reads from that same Counter. Otherwise, an incorrect count will be read.

READ-BACK COMMAND

The third method uses the Read-Back command. This command allows the user to check the count value, programmed Mode, and current state of the OUT pin and Null Count flag of the selected counter(s).

The command is written into the Control Word Register and has the format shown in Figure 10. The command applies to the counters selected by setting their corresponding bits D3,D2,D1 = 1.

A0, A1 = 11 $\overline{CS}$ = 0 $\overline{RD}$ = 1 $\overline{WR}$ = 0

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	1	$\overline{COUNT}$	$\overline{STATUS}$	CNT 2	CNT 1	CNT 0	0

D_5: 0 = Latch count of selected counter(s)
D_4: 0 = Latch status of selected counter(s)
D_3: 1 = Select counter 2
D_2: 1 = Select counter 1
D_1: 1 = Select counter 0
D_0: Reserved for future expansion; must be 0

Figure 10. Read-Back Command Format

The read-back command may be used to latch multiple counter output latches (OL) by setting the $\overline{COUNT}$ bit D5 = 0 and selecting the desired counter(s). This single command is functionally equivalent to several counter latch commands, one for each counter latched. Each counter's latched count is held until it is read (or the counter is reprogrammed). That counter is automatically unlatched when read, but other counters remain latched until they are read. If multiple count read-back commands are issued to the same counter without reading the

5

count, all but the first are ignored; i.e., the count which will be read is the count at the time the first read-back command was issued.

The read-back command may also be used to latch status information of selected counter(s) by setting $\overline{\text{STATUS}}$ bit D4 = 0. Status must be latched to be read; status of a counter is accessed by a read from that counter.

The counter status format is shown in Figure 11. Bits D5 through D0 contain the counter's programmed Mode exactly as written in the last Mode Control Word. OUTPUT bit D7 contains the current state of the OUT pin. This allows the user to monitor the counter's output via software, possibly eliminating some hardware from a system.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
OUTPUT	NULL COUNT	RW1	RW0	M2	M1	M0	BCD

D_7 1 = Out Pin is 1
0 = Out Pin is 0
D_6 1 = Null count
0 = Count available for reading
D_5-D_0 Counter Programmed Mode (See Figure 7)

Figure 11. Status Byte

NULL COUNT bit D6 indicates when the last count written to the counter register (CR) has been loaded into the counting element (CE). The exact time this happens depends on the Mode of the counter and is described in the Mode Definitions, but until the count is loaded into the counting element (CE), it can't be read from the counter. If the count is latched or read before this time, the count value will not reflect the new count just written. The operation of Null Count is shown in Figure 12.

THIS ACTION:	CAUSES:
A. Write to the control word register:[1]	Null count = 1
B. Write to the count register (CR);[2]	Null count = 1
C. New count is loaded into CE (CR → CE);	Null count = 0

[1] Only the counter specified by the control word will have its null count set to 1. Null count bits of other counters are unaffected.

[2] If the counter is programmed for two-byte counts (least significant byte then most significant byte) null count goes to 1 when the second byte is written.

Figure 12. Null Count Operation

If multiple status latch operations of the counter(s) are performed without reading the status, all but the first are ignored; i.e., the status that will be read is the status of the counter at the time the first status read-back command was issued.

Both count and status of the selected counter(s) may be latched simultaneously by setting both $\overline{\text{COUNT}}$ and $\overline{\text{STATUS}}$ bits D5,D4 = 0. This is functionally the same as issuing two separate read-back commands at once, and the above discussions apply here also. Specifically, if multiple count and/or status read-back commands are issued to the same counter(s) without any intervening reads, all but the first are ignored. This is illustrated in Figure 13.

If both count and status of a counter are latched, the first read operation of that counter will return latched status, regardless of which was latched first. The next one or two reads (depending on whether the counter is programmed for one or two type counts) return latched count. Subsequent reads return unlatched count.

Command								Description	Results
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0		
1	1	0	0	0	0	1	0	Read back count and status of Counter 0	Count and status latched for Counter 0
1	1	1	0	0	1	0	0	Read back status of Counter 1	Status latched for Counter 1
1	1	1	0	1	1	0	0	Read back status of Counters 2, 1	Status latched for Counter 2, but not Counter 1
1	1	0	1	1	0	0	0	Read back count of Counter 2	Count latched for Counter 2
1	1	0	0	0	1	0	0	Read back count and status of Counter 1	Count latched for Counter 1, but not status
1	1	1	0	0	0	1	0	Read back status of Counter 1	Command ignored, status already latched for Counter 1

Figure 13. Read-Back Command Example

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A_1	A_0	
0	1	0	0	0	Write into Counter 0
0	1	0	0	1	Write into Counter 1
0	1	0	1	0	Write into Counter 2
0	1	0	1	1	Write Control Word
0	0	1	0	0	Read from Counter 0
0	0	1	0	1	Read from Counter 1
0	0	1	1	0	Read from Counter 2
0	0	1	1	1	No-Operation (3-State)
1	X	X	X	X	No-Operation (3-State)
0	1	1	X	X	No-Operation (3-State)

Figure 14. Read/Write Operations Summary

Mode Definitions

The following are defined for use in describing the operation of the 82C54.

CLK PULSE: a rising edge, then a falling edge, in that order, of a Counter's CLK input.

TRIGGER: a rising edge of a Counter's GATE input.

COUNTER LOADING: the transfer of a count from the CR to the CE (refer to the "Functional Description")

MODE 0: INTERRUPT ON TERMINAL COUNT

Mode 0 is typically used for event counting. After the Control Word is written, OUT is initially low, and will remain low until the Counter reaches zero. OUT then goes high and remains high until a new count or a new Mode 0 Control Word is written into the Counter.

GATE = 1 enables counting; GATE = 0 disables counting. GATE has no effect on OUT.

After the Control Word and initial count are written to a Counter, the initial count will be loaded on the next CLK pulse. This CLK pulse does not decrement the count, so for an initial count of N, OUT does not go high until N + 1 CLK pulses after the initial count is written.

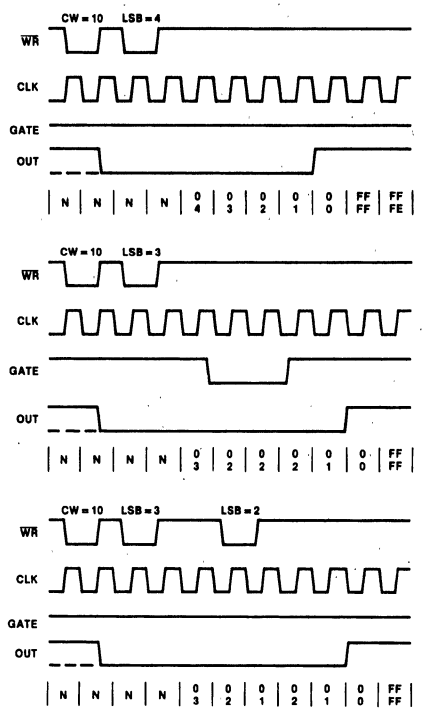
If a new count is written to the Counter, it will be loaded on the next CLK pulse and counting will continue from the new count. If a two-byte count is written, the following happens:

1) Writing the first byte does not disable counting. OUT is set low immediately (no clock pulse required).
2) Writing the second byte allows the new count to be loaded on the next CLK pulse.
3) When there is a count in progress, writing a new LSB before the counter has counted down to 0 and rolled over to FFFFh, WILL stop the counter. However, if the LSB is loaded AFTER the counter has rolled over to FFFFh, so that an MSB now exists in the counter, then the counter WILL NOT stop.

This allows the counting sequence to be synchronized by software. Again, OUT does not go high until N + 1 CLK pulses after the new count of N is written.

5

If an initial count is written while GATE = 0, it will still be loaded on the next CLK pulse. When GATE goes high, OUT will go high N CLK pulses later; no CLK pulse is needed to load the Counter as this has already been done.



NOTE:
The Following Conventions Apply To All Mode Timing Diagrams:
1. Counters are programmed for binary (not BCD) counting and for Reading/Writing least significant byte (LSB) only.
2. The counter is always selected ($\overline{CS}$ always low).
3. CW stands for "Control Word"; CW = 10 means a control word of 10, hex is written to the counter.
4. LSB stands for "Least Significant Byte" of count.
5. Numbers below diagrams are count values.
The lower number is the least significant byte.
The upper number is the most significant byte. Since the counter is programmed to Read/Write LSB only, the most significant byte cannot be read.
N stands for an undefined count.
Vertical lines show transitions between count values.

Figure 15. Mode 0

MODE 1: HARDWARE RETRIGGERABLE ONE-SHOT

OUT will be initially high. OUT will go low on the CLK pulse following a trigger to begin the one-shot pulse, and will remain low until the Counter reaches zero. OUT will then go high and remain high until the CLK pulse after the next trigger.

After writing the Control Word and initial count, the Counter is armed. A trigger results in loading the Counter and setting OUT low on the next CLK pulse, thus starting the one-shot pulse. An initial count of N will result in a one-shot pulse N CLK cycles in duration. The one-shot is retriggerable, hence OUT will remain low for N CLK pulses after any trigger. The one-shot pulse can be repeated without rewriting the same count into the counter. GATE has no effect on OUT.

If a new count is written to the Counter during a one-shot pulse, the current one-shot is not affected unless the Counter is retriggered. In that case, the Counter is loaded with the new count and the one-shot pulse continues until the new count expires.

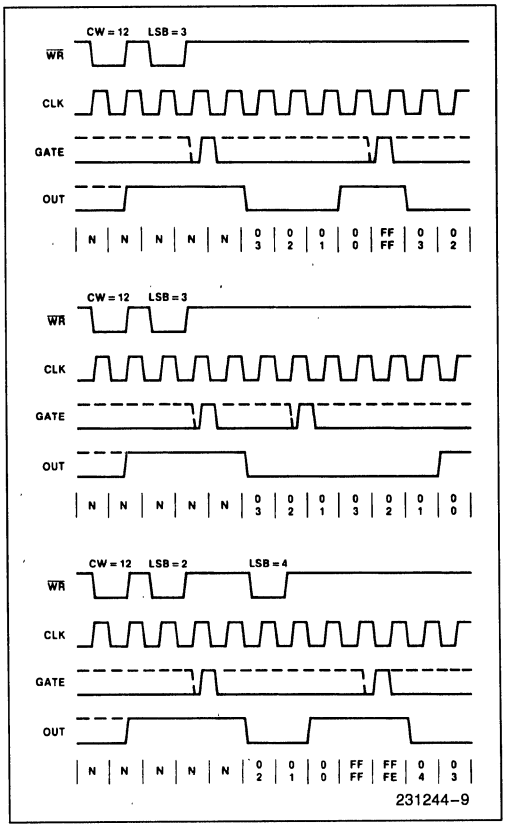


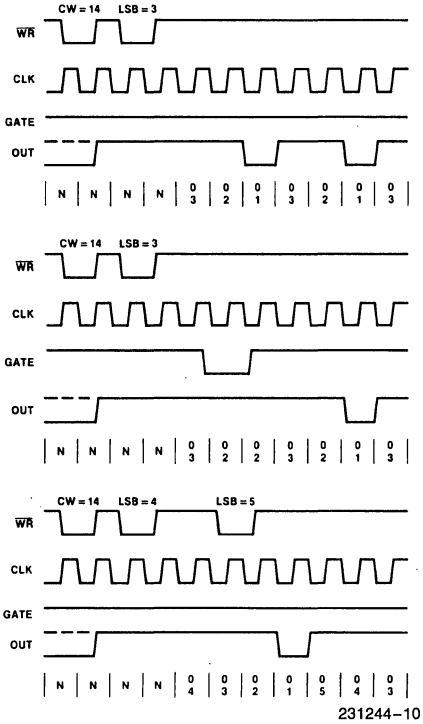
Figure 16. Mode 1

MODE 2: RATE GENERATOR

This Mode functions like a divide-by-N counter. It is typically used to generate a Real Time Clock interrupt. OUT will initially be high. When the initial count has decremented to 1, OUT goes low for one CLK pulse. OUT then goes high again, the Counter reloads the initial count and the process is repeated. Mode 2 is periodic; the same sequence is repeated indefinitely. For an initial count of N, the sequence repeats every N CLK cycles.

GATE = 1 enables counting; GATE = 0 disables counting. If GATE goes low during an output pulse, OUT is set high immediately. A trigger reloads the Counter with the initial count on the next CLK pulse; OUT goes low N CLK pulses after the trigger. Thus the GATE input can be used to synchronize the Counter.

After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. OUT goes low N CLK Pulses after the initial count is written. This allows the Counter to be synchronized by software also.



NOTE:
A GATE transition should not occur one clock prior to terminal count.

Figure 17. Mode 2

Writing a new count while counting does not affect the current counting sequence. If a trigger is received after writing a new count but before the end of the current period, the Counter will be loaded with the new count on the next CLK pulse and counting will continue from the new count. Otherwise, the new count will be loaded at the end of the current counting cycle. In mode 2, a COUNT of 1 is illegal.

MODE 3: SQUARE WAVE MODE

Mode 3 is typically used for Baud rate generation. Mode 3 is similar to Mode 2 except for the duty cycle of OUT. OUT will initially be high. When half the initial count has expired, OUT goes low for the remainder of the count. Mode 3 is periodic; the sequence above is repeated indefinitely. An initial count of N results in a square wave with a period of N CLK cycles.

GATE = 1 enables counting; GATE = 0 disables counting. If GATE goes low while OUT is low, OUT is set high immediately; no CLK pulse is required. A trigger reloads the Counter with the initial count on the next CLK pulse. Thus the GATE input can be used to synchronize the Counter.

After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. This allows the Counter to be synchronized by software also.

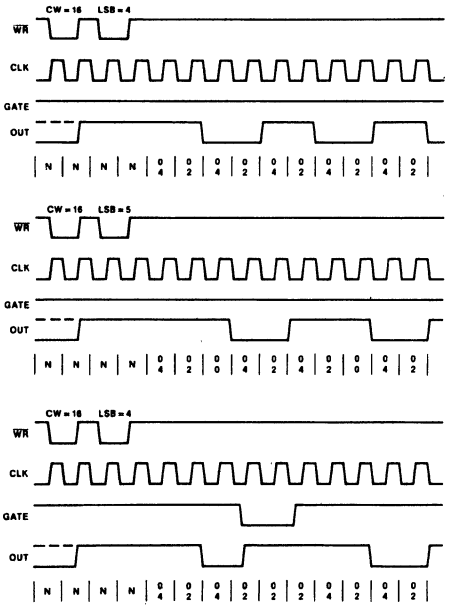
Writing a new count while counting does not affect the current counting sequence. If a trigger is received after writing a new count but before the end of the current half-cycle of the square wave, the Counter will be loaded with the new count on the next CLK pulse and counting will continue from the new count. Otherwise, the new count will be loaded at the end of the current half-cycle.

Mode 3 is implemented as follows:

Even counts: OUT is initially high. The initial count is loaded on one CLK pulse and then is decremented by two on succeeding CLK pulses. When the count expires OUT changes value and the Counter is reloaded with the initial count. The above process is repeated indefinitely.

Odd counts: OUT is initially high. The initial count minus one (an even number) is loaded on one CLK pulse and then is decremented by two on succeeding CLK pulses. One CLK pulse *after* the count expires, OUT goes low and the Counter is reloaded with the initial count minus one. Succeeding CLK pulses decrement the count by two. When the count expires, OUT goes high again and the Counter is reloaded with the initial count minus one. The above process is repeated indefinitely. So for odd counts,

OUT will be high for (N + 1)/2 counts and low for (N −1)/2 counts.



NOTE:
A GATE transition should not occur one clock prior to terminal count.

Figure 18. Mode 3

MODE 4: SOFTWARE TRIGGERED STROBE

OUT will be initially high. When the initial count expires, OUT will go low for one CLK pulse and then go high again. The counting sequence is "triggered" by writing the initial count.

GATE = 1 enables counting; GATE = 0 disables counting. GATE has no effect on OUT.

After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. This CLK pulse does not decrement the count, so for an initial count of N, OUT does not strobe low until N + 1 CLK pulses after the initial count is written.

If a new count is written during counting, it will be loaded on the next CLK pulse and counting will continue from the new count. If a two-byte count is written, the following happens:

1) Writing the first byte has no effect on counting.
2) Writing the second byte allows the new count to be loaded on the next CLK pulse.

This allows the sequence to be "retriggered" by software. OUT strobes low N+1 CLK pulses after the new count of N is written.

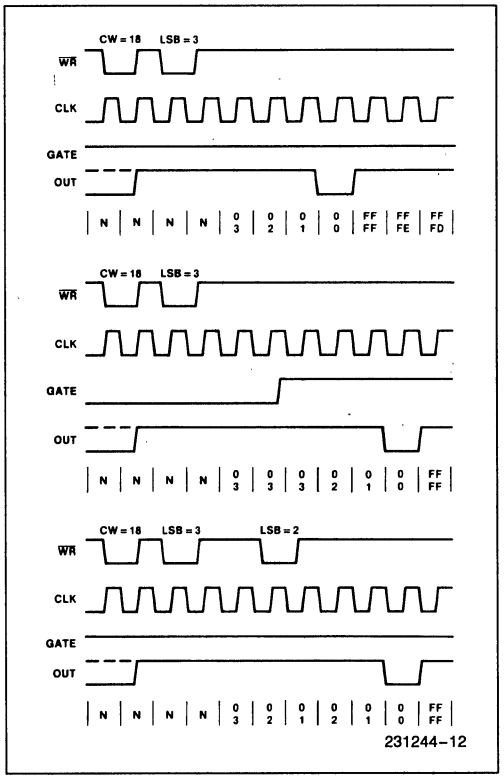


Figure 19. Mode 4

MODE 5: HARDWARE TRIGGERED STROBE (RETRIGGERABLE)

OUT will initially be high. Counting is triggered by a rising edge of GATE. When the initial count has expired, OUT will go low for one CLK pulse and then go high again.

After writing the Control Word and initial count, the counter will not be loaded until the CLK pulse after a trigger. This CLK pulse does not decrement the count, so for an initial count of N, OUT does not strobe low until N+1 CLK pulses after a trigger.

A trigger results in the Counter being loaded with the initial count on the next CLK pulse. The counting sequence is retriggerable. OUT will not strobe low for N + 1 CLK pulses after any trigger. GATE has no effect on OUT.

If a new count is written during counting, the current counting sequence will not be affected. If a trigger occurs after the new count is written but before the current count expires, the Counter will be loaded with the new count on the next CLK pulse and counting will continue from there.

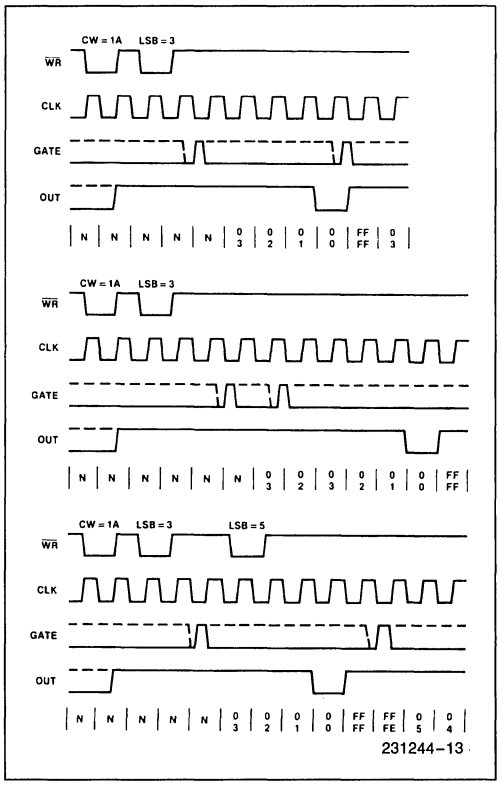


Figure 20. Mode 5

Signal Status Modes	Low Or Going Low	Rising	High
0	Disables counting	—	Enables counting
1	—	1) Initiates counting 2) Resets output after next clock	—
2	1) Disables counting 2) Sets output immediately high	Initiates counting	Enables counting
3	1) Disables counting 2) Sets output immediately high	Initiates counting	Enables counting
4	Disables counting	—	Enables counting
5	—	Initiates counting	—

Figure 21. Gate Pin Operations Summary

MODE	MIN COUNT	MAX COUNT
0	1	0
1	1	0
2	2	0
3	2	0
4	1	0

NOTE:
0 is equivalent to 2^{16} for binary counting and 10^4 for BCD counting

Figure 22. Minimum and Maximum initial Counts

5

Operation Common to All Modes

Programming

When a Control Word is written to a Counter, all Control Logic is immediately reset and OUT goes to a known initial state; no CLK pulses are required for this.

GATE

The GATE input is always sampled on the rising edge of CLK. In Modes 0, 2, 3, and 4 the GATE input is level sensitive, and the logic level is sampled on the rising edge of CLK. In Modes 1, 2, 3, and 5 the GATE input is rising-edge sensitive. In these Modes, a rising edge of GATE (trigger) sets an edge-sensitive flip-flop in the Counter. This flip-flop is then sampled on the next rising edge of CLK; the flip-flop is reset immediately after it is sampled. In this way, a trigger will be detected no matter when it occurs—a high logic level does not have to be maintained until the next rising edge of CLK. Note that in Modes 2 and 3, the GATE input is both edge- and level-sensitive. In Modes 2 and 3, if a CLK source other than the system clock is used, GATE should be pulsed immediately following $\overline{WR}$ of a new count value.

COUNTER

New counts are loaded and Counters are decremented on the falling edge of CLK.

The largest possible initial count is 0; this is equivalent to 2^{16} for binary counting and 10^4 for BCD counting.

The Counter does not stop when it reaches zero. In Modes 0, 1, 4, and 5 the Counter "wraps around" to the highest count, either FFFF hex for binary counting or 9999 for BCD counting, and continues counting. Modes 2 and 3 are periodic; the Counter reloads itself with the initial count and continues counting from there.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias.....0°C to 70°C
Storage Temperature .....−65° to +150°C
Supply Voltage .....−0.5 to +8.0V
Operating Voltage .....+4V to +7V
Voltage on any Input.....GND −2V to +6.5V
Voltage on any Output ..GND−0.5V to V_{CC} + 0.5V
Power Dissipation .....1 Watt

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**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.*

D.C. CHARACTERISTICS

(T_A=0°C to 70°C, V_{CC}=5V± 10%, GND=0V) (T_A = −40°C to +85°C for Extended Temperature)

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 2.5 mA
V_{OH}	Output High Voltage	3.0 V_{CC} − 0.4		V V	I_{OH} = −2.5 mA I_{OH} = −100 µA
I_{IL}	Input Load Current		±2.0	µA	V_{IN}=V_{CC} to 0V
I_{OFL}	Output Float Leakage Current		±10	µA	V_{OUT}=V_{CC} to 0.0V
I_{CC}	V_{CC} Supply Current		20	mA	Clk Freq= 8MHz 82C54 10MHz 82C54-2
I_{CCSB}	V_{CC} Supply Current-Standby		10	µA	CLK Freq = DC $\overline{CS}$ = V_{CC}. All Inputs/Data Bus V_{CC} All Outputs Floating
I_{CCSB1}	V_{CC} Supply Current-Standby		150	µA	CLK Freq = DC $\overline{CS}$ = V_{CC}. All Other Inputs, I/O Pins = V_{GND}, Outputs Open
C_{IN}	Input Capacitance		10	pF	f_c = 1 MHz
$C_{I/O}$	I/O Capacitance		20	pF	Unmeasured pins returned to GND(5)
C_{OUT}	Output Capacitance		20	pF	

A.C. CHARACTERISTICS

(T_A = 0°C to 70°C, V_{CC} = 5V ±10%, GND =0V) (T_A = −40°C to +85°C for Extended Temperature)

BUS PARAMETERS (Note 1)

READ CYCLE

Symbol	Parameter	82C54-2		Units
		Min	Max	
t_{AR}	Address Stable Before $\overline{RD}$ ↓	30		ns
t_{SR}	$\overline{CS}$ Stable Before $\overline{RD}$ ↓	0		ns
t_{RA}	Address Hold Time After $\overline{RD}$ ↑	0		ns
t_{RR}	$\overline{RD}$ Pulse Width	95		ns
t_{RD}	Data Delay from $\overline{RD}$ ↓		85	ns
t_{AD}	Data Delay from Address		185	ns
t_{DF}	$\overline{RD}$ ↑ to Data Floating	5	65	ns
t_{RV}	Command Recovery Time	165		ns

NOTE:
1. AC timings measured at V_{OH} = 2.0V, V_{OL} = 0.8V.

5

A.C. CHARACTERISTICS (Continued)

WRITE CYCLE

Symbol	Parameter	82C54-2		Units
		Min	Max	
t_{AW}	Address Stable Before $\overline{WR}\downarrow$	0		ns
t_{SW}	$\overline{CS}$ Stable Before $\overline{WR}\downarrow$	0		ns
t_{WA}	Address Hold Time After $\overline{WR}\uparrow$	0		ns
t_{WW}	$\overline{WR}$ Pulse Width	95		ns
t_{DW}	Data Setup Time Before $\overline{WR}\uparrow$	95		ns
t_{WD}	Data Hold Time After $\overline{WR}\uparrow$	0		ns
t_{RV}	Command Recovery Time	165		ns

CLOCK AND GATE

Symbol	Parameter	82C54-2		Units
		Min	Max	
t_{CLK}	Clock Period	100	DC	ns
t_{PWH}	High Pulse Width	30(3)		ns
t_{PWL}	Low Pulse Width	50(3)		ns
T_R	Clock Rise Time		25	ns
t_F	Clock Fall Time		25	ns
t_{GW}	Gate Width High	50		ns
t_{GL}	Gate Width Low	50		ns
t_{GS}	Gate Setup Time to CLK $\uparrow$	40		ns
t_{GH}	Gate Hold Time After CLK $\uparrow$	50(2)		ns
T_{OD}	Output Delay from CLK $\downarrow$		100	ns
t_{ODG}	Output Delay from Gate $\downarrow$		100	ns
t_{WC}	CLK Delay for Loading(4)	0	55	ns
t_{WG}	Gate Delay for Sampling(4)	−5	40	ns
t_{WO}	OUT Delay from Mode Write		240	ns
t_{CL}	CLK Set Up for Count Latch	−40	40	ns

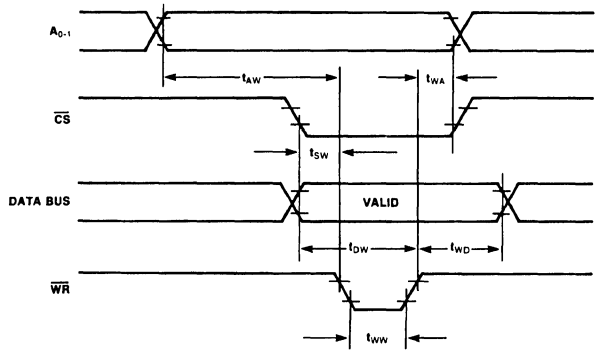
NOTES:
2. In Modes 1 and 5 triggers are sampled on each rising clock edge. A second trigger within 70 ns for the 82C54-2 of the rising clock edge may not be detected.
3. Low-going glitches that violate t_{PWH}, t_{PWL} may cause errors requiring counter reprogramming.
4. Except for Extended Temp., See Extended Temp. A.C. Characteristics below.
5. Sampled not 100% tested. T_A = 25°C.
6. If CLK present at T_{WC} min then Count equals N+2 CLK pulses, T_{WC} max equals Count N+1 CLK pulse. T_{WC} min to T_{WC} max, count will be either N+1 or N+2 CLK pulses.
7. In Modes 1 and 5, if GATE is present when writing a new Count value, at T_{WG} min Counter will not be triggered, at T_{WG} max Counter will be triggered.
8. If CLK present when writing a Counter Latch or ReadBack Command, at T_{CL} min CLK will be reflected in count value latched, at T_{CL} max CLK will not be reflected in the count value latched. Writing a Counter Latch or ReadBack Command between T_{CL} min and T_{WL} max will result in a latched count vallue which is ± one least significant bit.

EXTENDED TEMPERATURE (T_A = −40°C to +85°C for Extended Temperature)

Symbol	Parameter	82C54-2		Units
		Min	Max	
t_{WC}	CLK Delay for Loading	−25	25	ns
t_{WG}	Gate Delay for Sampling	−25	25	ns

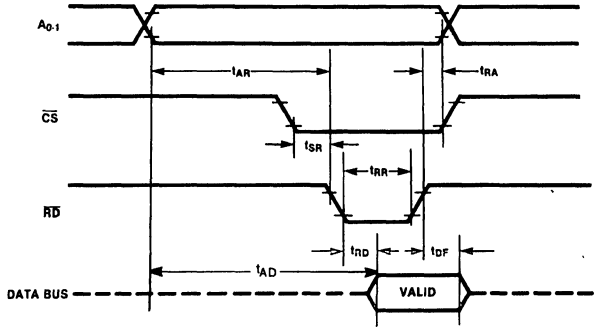
WAVEFORMS

WRITE



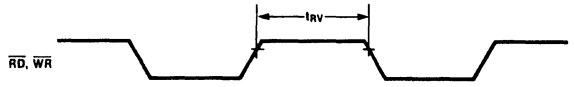
231244-14

READ



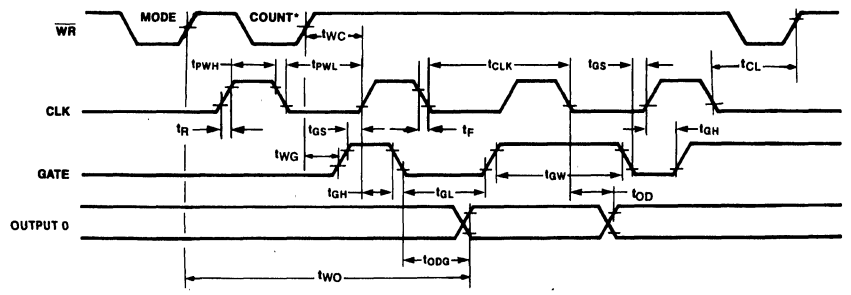
231244-15

RECOVERY



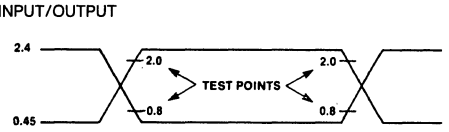
231244-16

CLOCK AND GATE



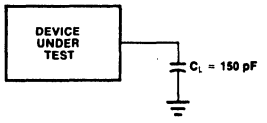
231244-17
* Last byte of count being written

A.C. TESTING INPUT, OUTPUT WAVEFORM



231244-18
A.C. Testing: Inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0." Timing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0."

A.C. TESTING LOAD CIRCUIT



231244-19
C_L = 150 pF
C_L includes jig capacitance

REVISION SUMMARY

The following list represents the key differences between Rev. 005 and 006 of the 82C54 Data Sheet.

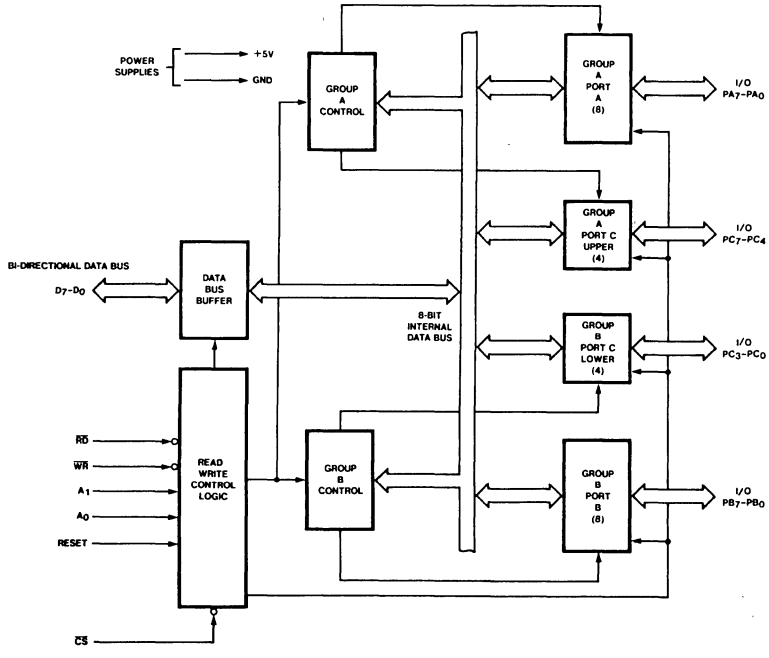
1. References to and specifications for the 8 MHz 82C54 are removed. Only the 10 MHz 82C52-2 remains in production.

8255A-5
PROGRAMMABLE PERIPHERAL INTERFACE

- **MCS-85 Compatible 8255A-5**
- **24 Programmable I/O Pins**
- **Completely TTL Compatible**
- **Fully Compatible with Intel Microprocessor Families**
- **Improved Timing Characteristics**
- **Direct Bit Set/Reset Capability Easing Control Application Interface**
- **Reduces System Package Count**
- **Improved DC Driving Capability**
- **Available in EXPRESS**
  - **— Standard Temperature Range**
  - **— Extended Temperature Range**
- **40 Pin DIP Package**

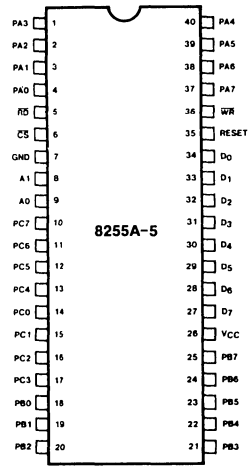
  (See Intel Packaging: Order Number: 240800-001, Package Type P)

The Intel 8255A is a general purpose programmable I/O device designed for use with Intel microprocessors. It has 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. In the first mode (MODE 0), each group of 12 I/O pins may be programmed in sets of 4 to be input or output. In MODE 1, the second mode, each group may be programmed to have 8 lines of input or output. Of the remaining 4 pins, 3 are used for handshaking and interrupt control signals. The third mode of operation (MODE 2) is a bidirectional bus mode which uses 8 lines for a bidirectional bus, and 5 lines, borrowing one from the other group, for handshaking.



231308-1

Figure 1. 8255A Block Diagram



231308-2

Figure 2. Pin Configuration

5

The complete document for this product is available from Intel's Literature Center at 1-800-548-4725.

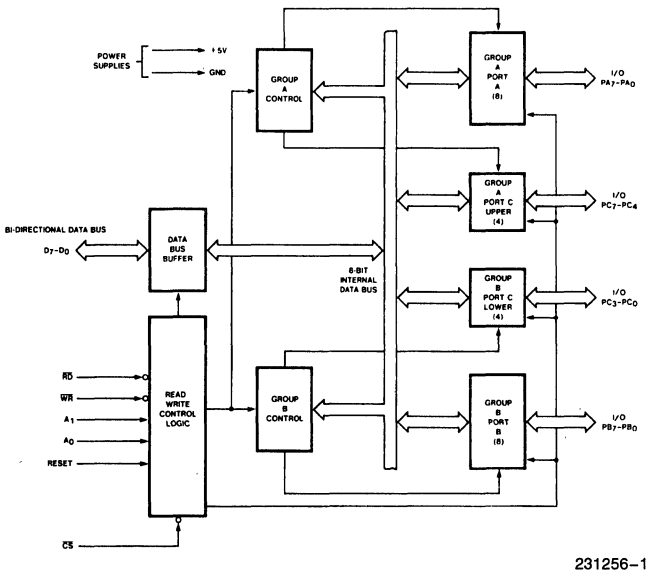
82C55A
CHMOS PROGRAMMABLE PERIPHERAL INTERFACE

- **Compatible with all Intel and Most Other Microprocessors**
- **High Speed, "Zero Wait State" Operation with 8 MHz 8086/88 and 80186/188**
- **24 Programmable I/O Pins**
- **Low Power CHMOS**
- **Completely TTL Compatible**
- **Control Word Read-Back Capability**
- **Direct Bit Set/Reset Capability**
- **2.5 mA DC Drive Capability on all I/O Port Outputs**
- **Available in 40-Pin DIP and 44-Pin PLCC**
- **Available in EXPRESS**
  - **— Standard Temperature Range**
  - **— Extended Temperature Range**

The Intel 82C55A is a high-performance, CHMOS version of the industry standard 8255A general purpose programmable I/O device which is designed for use with all Intel and most other microprocessors. It provides 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. The 82C55A is pin compatible with the NMOS 8255A and 8255A-5.

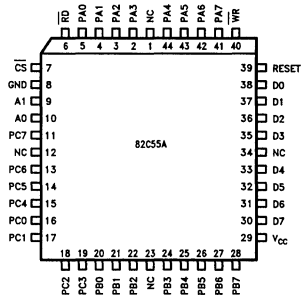
In MODE 0, each group of 12 I/O pins may be programmed in sets of 4 and 8 to be inputs or outputs. In MODE 1, each group may be programmed to have 8 lines of input or output. 3 of the remaining 4 pins are used for handshaking and interrupt control signals. MODE 2 is a strobed bi-directional bus configuration.

The 82C55A is fabricated on Intel's advanced CHMOS III technology which provides low power consumption with performance equal to or greater than the equivalent NMOS product. The 82C55A is available in 40-pin DIP and 44-pin plastic leaded chip carrier (PLCC) packages.

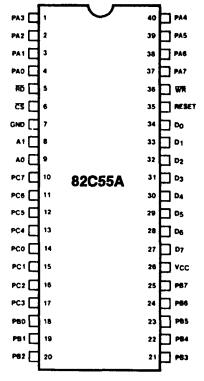


231256-1

Figure 1. 82C55A Block Diagram



231256-31



231256-2

Figure 2. 82C55A Pinout

Diagrams are for pin reference only. Package sizes are not to scale.

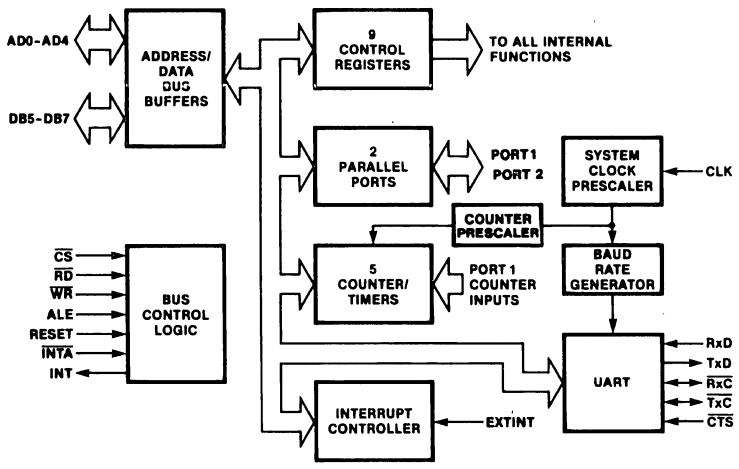
The complete document for this product is available from Intel's Literature Center at 1-800-548-4725.

October 1995
Order Number: 231256-004

8256AH
MULTIFUNCTION MICROPROCESSOR SUPPORT CONTROLLER

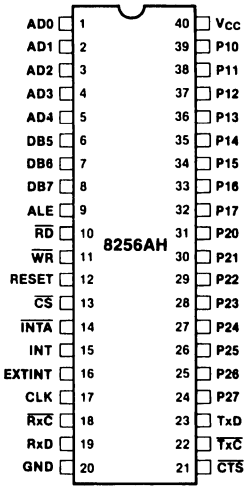
- **Programmable Serial Asynchronous Communications Interface for 5-, 6-, 7-, or 8-Bit Characters, 1, 1½, or 2 Stop Bits, and Parity Generation**
- **On-Board Baud Rate Generator Programmable for 13 Common Baud Rates up to 19.2 KBits/Second, or an External Baud Clock Maximum of 1M Bit/Second**
- **Five 8-Bit Programmable Timer/ Counters; Four Can Be Cascaded to Two 16-Bit Timer/Counters**
- **Two 8-Bit Programmable Parallel I/O Ports; Port 1 Can Be Programmed for Port 2 Handshake Controls and Event Counter Inputs**
- **Eight-Level Priority Interrupt Controller Programmable for 8085 or iAPX 86, iAPX 88 Systems and for Fully Nested Interrupt Capability**
- **Programmable System Clock to 1 ×, 2 ×, 3 ×, or 5 × 1.024 MHz**

The Intel 8256AH Multifunction Universal Asynchronous Receiver-Transmitter (MUART) combines five commonly used functions into a single 40-pin device. It is designed to interface to the 8086/88, iAPX 186/188, and 8051 to perform serial communications, parallel I/O, timing, event counting, and priority interrupt functions. All of these functions are fully programmable through nine internal registers. In addition, the five timer/counters and two parallel I/O ports can be accessed directly by the microprocessor.



230759-1

Figure 1. MUART Block Diagram



230759-2

Figure 2. MUART Pin Configuration

5

The complete document for this product is available from Intel's Literature Center at 1-800-548-4725.

October 1994
Order Number: 230759-002

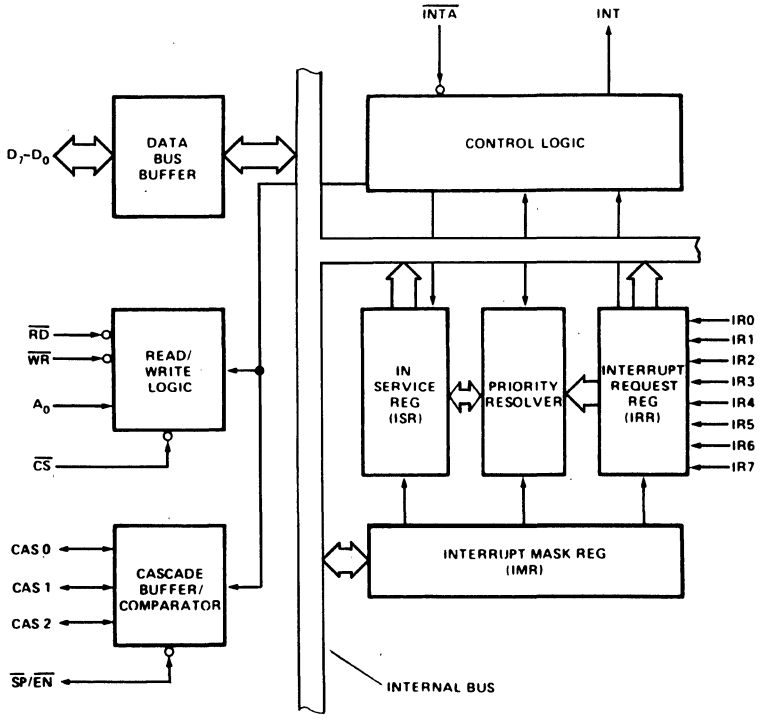
8259A
PROGRAMMABLE INTERRUPT CONTROLLER
(8259A/8259A-2)

- **8086, 8088 Compatible**
- **MCS-80, MCS-85 Compatible**
- **Eight-Level Priority Controller**
- **Expandable to 64 Levels**
- **Programmable Interrupt Modes**
- **Individual Request Mask Capability**
- **Single +5V Supply (No Clocks)**
- **Available in 28-Pin DIP and 28-Lead PLCC Package**
  (See Packaging Spec., Order #231369)
- **Available in EXPRESS**
  — Standard Temperature Range
  — Extended Temperature Range

The Intel 8259A Programmable Interrupt Controller handles up to eight vectored priority interrupts for the CPU. It is cascadable for up to 64 vectored priority interrupts without additional circuitry. It is packaged in a 28-pin DIP, uses NMOS technology and requires a single +5V supply. Circuitry is static, requiring no clock input.

The 8259A is designed to minimize the software and real time overhead in handling multi-level priority interrupts. It has several modes, permitting optimization for a variety of system requirements.

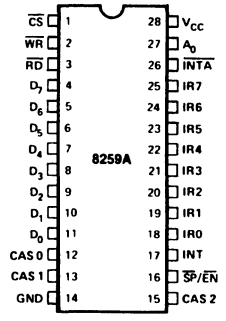
The 8259A is fully upward compatible with the Intel 8259. Software originally written for the 8259 will operate the 8259A in all 8259 equivalent modes (MCS-80/85, Non-Buffered, Edge Triggered).



231468-1

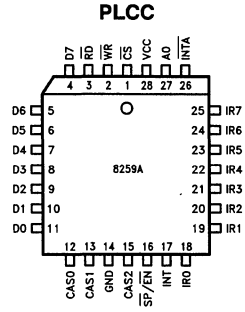
Figure 1. Block Diagram

DIP



231468-2

PLCC



231468-31

Figure 2. Pin Configurations

The complete document for this product is available from Intel's Literature Center at 1-800-548-4725.

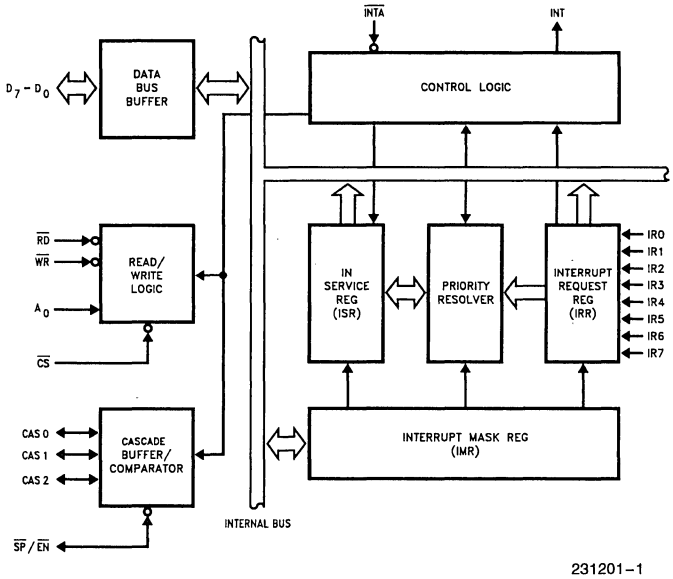
December 1988
Order Number: 231468-003

82C59A-2
CHMOS Programmable Interrupt Controller

- **Pin Compatible with NMOS 8259A-2**
- **Eight-Level Priority Controller**
- **Expandable to 64 levels**
- **Programmable Interrupt Modes**
- **Low Standby Power—10 μA**
- **Individual Request Mask Capability**
- **80C86/88 and 8080/85/86/88 Compatible**
- **Fully Static Design**
- **Single 5V Power Supply**
- **Available in 28-Pin Plastic DIP**
  (See Packaging Spec., Order #231369)

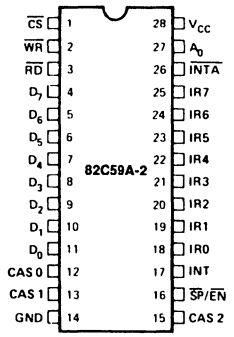
The Intel 82C59A-2 is a high performance CHMOS version of the NMOS 8259A-2 Priority Interrupt Controller. The 82C59A-2 is designed to relieve the system CPU from the task of polling in a multi-level priority interrupt system. The high speed and industry standard configuration of the 82C59A-2, make it compatible with microprocessors such as the 80C86/88, 8086/88 and 8080/85.

The 82C59A-2 can handle up to 8 vectored priority interrupts for the CPU and is cascadable to 64 without additional circuitry. It is designed to minimize the software and real time overhead in handling multi-level priority interrupts. Two modes of operation make the 82C59A-2 optimal for a variety of system requirements. Static CHMOS circuit design, requiring no clock input, insures low operating power. It is packaged in a 28-pin plastic DIP.



231201-1

Figure 1. Block Diagram



231201-2

Figure 2. Pin Configuration

5

Table 1. Pin Description

Symbol	Pin No.	Type	Name and Function
V_{CC}	28	I	**SUPPLY:** +5V Supply.
GND	14	I	**GROUND.**
$\overline{CS}$	1	I	**CHIP SELECT:** A low on this pin enables $\overline{RD}$ and $\overline{WR}$ communication between the CPU and the 82C59A-2. INTA functions are independent of CS.
$\overline{WR}$	2	I	**WRITE:** A low on this pin when $\overline{CS}$ is low enables the 82C59A-2 to accept command words from the CPU.
$\overline{RD}$	3	I	**READ:** A low on this pin when $\overline{CS}$ is low enables the 82C59A-2 to release status onto the data bus for the CPU.
D_7–D_0	4–11	I/O	**BIDIRECTIONAL DATA BUS:** Control, status and interrupt-vector information is transferred via this bus.
CAS_0–CAS_2	12, 13, 15	I/O	**CASCADE LINES:** The CAS lines form a private 82C59A-2 bus to control a multiple 82C59A-2 structure. These pins are outputs for a master 82C59A-2 and inputs for a slave 82C59A-2.
$\overline{SP}/\overline{EN}$	16	I/O	**SLAVE PROGRAM/ENABLE BUFFER:** This is a dual function pin. When in the Buffered Mode it can be used as an output to control buffer transceivers (EN). When not in the buffered mode it is used as an input to designate a master (SP = 1) or slave (SP = 0).
INT	17	O	**INTERRUPT:** This pin goes high whenever a valid interrupt request is asserted. It is used to interrupt the CPU, thus it is connected to the CPU's interrupt pin.
IR_0–IR_7	18–25	I	**INTERRUPT REQUESTS:** Asynchronous inputs. An interrupt request is executed by raising an IR input (low to high), and holding it high until it is acknowledged (Edge Triggered Mode), or just by a high level on an IR input (Level Triggered Mode). Internal pull-up resistors are implemented on IR0–7.
$\overline{INTA}$	26	I	**INTERRUPT ACKNOWLEDGE:** This pin is used to enable 82C59A-2 interrupt-vector data onto the data bus by a sequence of interrupt acknowledge pulses issued by the CPU.
A_0	27	I	**AO ADDRESS LINE:** This pin acts in conjunction with the $\overline{CS}$, $\overline{WR}$, and $\overline{RD}$ pins. It is used by the 82C59A-2 to decipher various Command Words the CPU writes and status the CPU wishes to read. It is typically connected to the CPU A0 address line (A1 for 80C86, 80C88).

FUNCTIONAL DESCRIPTION

Interrupts in Microcomputer Systems

Microcomputer system design requires that I/O devices such as keyboards, displays, sensors and other components receive servicing in an efficient manner so that large amounts of the total system tasks can be assumed by the microcomputer with little or no effect on throughput.

The most common method of servicing such devices is the *Polled* approach. This is where the processor must test each device in sequence and in effect "ask" each one if it needs servicing. It is easy to see that a large portion of the main program is looping through this continuous polling cycle and that such a method would have a serious, detrimental effect on system throughput, thus limiting the tasks that could be assumed by the microcomputer and reducing the cost effectiveness of using such devices.

A more desirable method would be one that would allow the microprocessor to be executing its main program and only stop to service peripheral devices when it is told to do so by the device itself. In effect, the method would provide an external asynchronous input that would inform the processor that it should complete whatever instruction that is currently being executed and fetch a new routine that will service the requesting device. Once this servicing is complete, however, the processor would resume exactly where it left off.

This method is called *Interrupt*. It is easy to see that system throughput would drastically increase, and thus more tasks could be assumed by the microcomputer to further enhance its cost effectiveness.

The Programmable Interrupt Controller (PIC) functions as an overall manager in an Interrupt-Driven system environment. It accepts requests from the peripheral equipment, determines which of the incoming requests is of the highest importance (priority), ascertains whether the incoming request has a higher priority value than the level currently being serviced, and issues an interrupt to the CPU based on this determination.

Each peripheral device or structure usually has a special program or "routine" that is associated with its specific functional or operational requirements; this is referred to as a "service routine". The PIC, after issuing an Interrupt to the CPU, must somehow input information into the CPU that can "point" the Program Counter to the service routine associated with the requesting device. This "pointer" is an address in a vectoring table and will often be referred to, in this document, as vectoring data.

The 82C59A-2

The 82C59A-2 is a device specifically designed for use in real time, interrupt driven microcomputer sys-

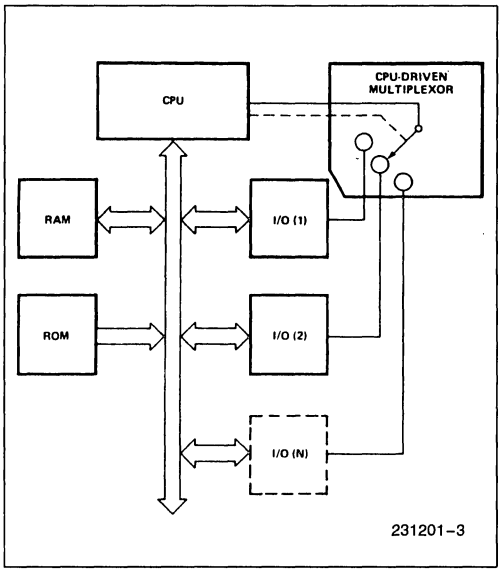


Figure 3a. Polled Method

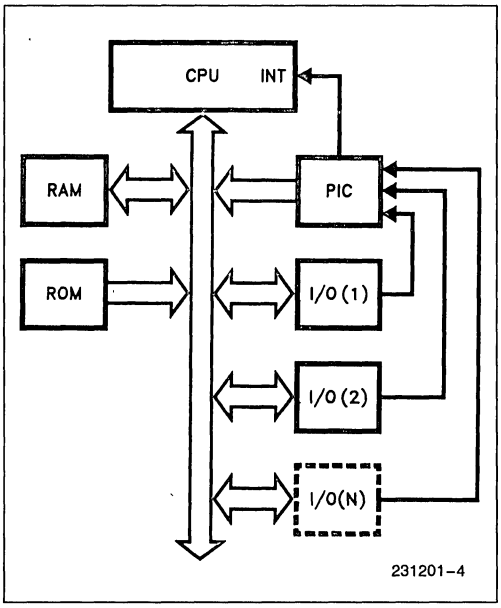


Figure 3b. Interrupt Method

5

tems. It manages eight levels or requests and has built-in features for expandability to other 82C59A-2's (up to 64 levels). It is programmed by the system's software as an I/O peripheral. A selection of priority modes is available to the programmer so that the manner in which the requests are processed by the 82C59A-2 can be configured to match system requirements. The priority modes can be changed or reconfigured dynamically at any time during the main program. This means that the complete interrupt structure can be defined as required, based on the total system environment.

INTERRUPT REQUEST REGISTER (IRR) AND IN-SERVICE REGISTER (ISR)

The interrupts at the IR input lines are handled by two registers in cascade, the Interrupt Request Register (IRR) and the In-Service Register (ISR). The IRR is used to store all the interrupt levels which are requesting service; and the ISR is used to store all the interrupt levels which are being serviced.

PRIORITY RESOLVER

This logic block determines the priorities of the bits set in the IRR. The highest priority is selected and strobed into the corresponding bit of the ISR during $\overline{INTA}$ pulse.

INTERRUPT MASK REGISTER (IMR)

The IMR stores the bits which mask the interrupt lines to be masked. The IMR operates on the IRR. Masking of a higher priority input will not affect the interrupt request lines of lower priority.

INT (INTERRUPT)

This output goes directly to the CPU interrupt input. The V_{OH} level on this line is designed to be fully compatible with the 8080A, 8085A, 80C88 and 80C86 input levels.

$\overline{INTA}$ (INTERRUPT ACKNOWLEDGE)

$\overline{INTA}$ pulses will cause the 82C59A-2 to release vectoring information onto the data bus. The format of this data depends on the system mode (μPM) of the 82C59A-2.

DATA BUS BUFFER

This 3-state, bidirectional 8-bit buffer is used to interface the 82C59A-2 to the system Data Bus. Control words and status information are transferred through the Data Bus Buffer.

READ/WRITE CONTROL LOGIC

The function of this block is to accept OUTput commands from the CPU. It contains the Initialization Command Word (ICW) registers and Operation Command Word (OCW) registers which store the various control formats for device operation. This function block also allows the status of the 82C59A-2 to be transferred onto the Data Bus.

$\overline{CS}$ (CHIP SELECT)

A LOW on this input enables the 82C59A-2. No reading or writing of the chip will occur unless the device is selected.

$\overline{WR}$ (WRITE)

A LOW on this input enables the CPU to write control words (ICWs and OCWs) to the 82C59A-2.

$\overline{RD}$ (READ)

A LOW on this input enables the 82C59A-2 to send the status of the Interrupt Request Register (IRR), In Service Register (ISR), the Interrupt Mask Register (IMR), or the Interrupt level onto the Data Bus.

A_0

This input signal is used in conjunction with $\overline{WR}$ and $\overline{RD}$ signals to write commands into the various command registers, as well as reading the various status registers of the chip. This line can be tied directly to one of the address lines.

THE CASCADE BUFFER/COMPARATOR

This function block stores and compares the IDs of all 82C59A-2's used in the system. The associated three I/O pins (CAS0–2) are outputs when the 82C59A-2 is used as a master and are inputs when the 82C59A-2 is used as a slave. As a master, the 82C59A-2 sends the ID of the interrupting slave device onto the CAS0–2 lines. The slave thus selected will send its preprogrammed subroutine address onto the Data Bus during the next one or two consecutive $\overline{INTA}$ pulses. (See section "Cascading the 82C59A-2".)

INTERRUPT SEQUENCE

The powerful features of the 82C59A-2 in a microcomputer system are its programmability and the interrupt routine addressing capability. The latter allows direct or indirect jumping to the specific interrupt routine requested without any polling of the interrupting devices. The normal sequence of events during an interrupt depends on the type of CPU being used.

The events occur as follows in an MCS-80/85 system:

1. One or more of the INTERRUPT REQUEST Lines (IR7–0) are raised high, setting the corresponding IRR bit(s).

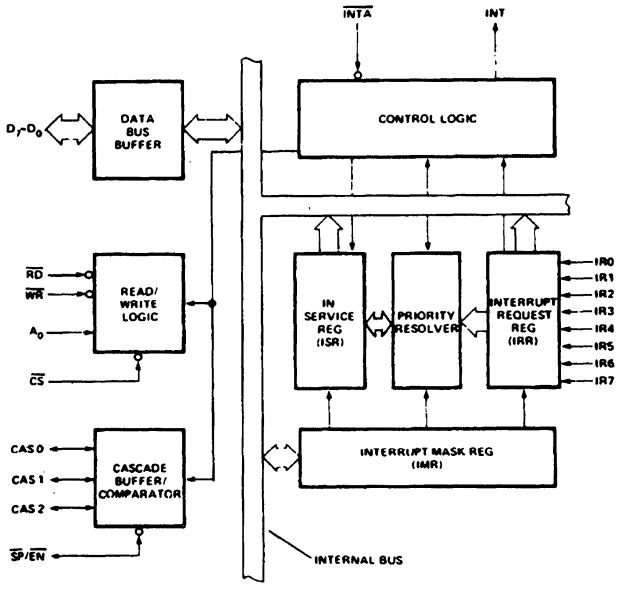


Figure 4. 82C59A-2 Block Diagram

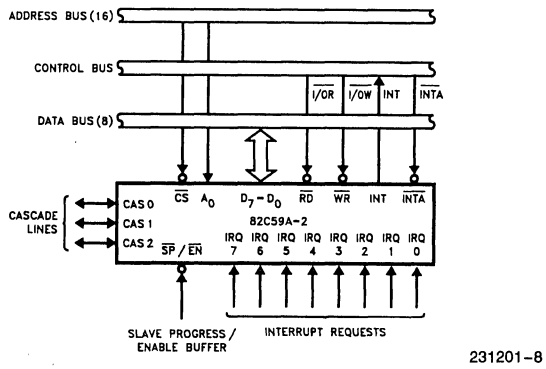


Figure 5. 82C59A-2 Interface to Standard System Bus

2. The 82C59A-2 evaluates these requests, and sends an INT to the CPU, if appropriate.
3. The CPU acknowledges the INT and responds with an $\overline{INTA}$ pulse.
4. Upon receiving an $\overline{INTA}$ from the CPU group, the highest priority ISR bit is set, and the corresponding IRR bit is reset. The 82C59A-2 will also release a CALL instruction code (11001101) onto the 8-bit Data Bus through its D7–0 pins.
5. This CALL instruction will initiate two more $\overline{INTA}$ pulses to be sent to the 82C59A-2 from the CPU group.
6. These two $\overline{INTA}$ pulses allow the 82C59A-2 to release its preprogrammed subroutine address onto the Data Bus. The lower 8-bit address is released at the first $\overline{INTA}$ pulse and the higher 8-bit address is released at the second $\overline{INTA}$ pulse.
7. This completes the 3-byte CALL instruction released by the 82C59A-2. In the AEOI mode the ISR bit is reset at the end of the third $\overline{INTA}$ pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt sequence.

The events occurring in an 80C86 system are the same until step 4.

4. Upon receiving an $\overline{INTA}$ from the CPU group, the highest priority ISR bit is set and the corresponding IRR bit is reset. The 82C59A-2 does not drive the Data Bus during this cycle.
5. The 80C86 will initiate a second $\overline{INTA}$ pulse. During this pulse, the 82C59A-2 releases an 8-bit pointer onto the Data Bus where it is read by the CPU.
6. This completes the interrupt cycle. In the AEOI mode the ISR bit is reset at the end of the second $\overline{INTA}$ pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt subroutine.

If no interrupt is present at step 4 of either sequence (i.e., the request was too short in duration) the 82C59A-2 will issue an interrupt level 7. Both the vectoring bytes and the CAS lines will look like an interrupt level 7 was requested.

When the 82C59A-2 PIC receives an interrupt, INT becomes active and an interrupt acknowledge cycle is started. If a higher priority interrupt occurs between the two INTA pulses, the INT line goes inactive immediately after the second INTA pulse. After an unspecified amount of time the INT line is activated again to signify the higher priority interrupt waiting for service. This inactive time is not specified and can vary between parts. The designer should be aware of this consideration when designing a system which uses the 82C59A-2. It is recommended that proper asynchronous design techniques be followed.

INTERRUPT SEQUENCE OUTPUTS

MCS®-80, MCS-85

This sequence is timed by three $\overline{INTA}$ pulses. During the first $\overline{INTA}$ pulse the CALL opcode is enabled onto the data bus.

Content of First Interrupt Vector Byte

	D7	D6	D5	D4	D3	D2	D1	D0
CALL CODE	1	1	0	0	1	1	0	1

During the second $\overline{INTA}$ pulse the lower address of the appropriate service routine is enabled onto the data bus. When Interval = 4 bits A_5–A_7 are programmed, while A_0–A_4 are automatically inserted by the 82C59A-2. When Interval = 8 only A_6 and A_7 are programmed, while A_0–A_5 are automatically inserted.

Content of Second Interrupt Vector Byte

IR	Interval = 4							
	D7	D6	D5	D4	D3	D2	D1	D0
7	A7	A6	A5	1	1	1	0	0
6	A7	A6	A5	1	1	0	0	0
5	A7	A6	A5	1	0	1	0	0
4	A7	A6	A5	1	0	0	0	0
3	A7	A6	A5	0	1	1	0	0
2	A7	A6	A5	0	1	0	0	0
1	A7	A6	A5	0	0	1	0	0
0	A7	A6	A5	0	0	0	0	0

IR	Interval = 8							
	D7	D6	D5	D4	D3	D2	D1	D0
7	A7	A6	1	1	1	0	0	0
6	A7	A6	1	1	0	0	0	0
5	A7	A6	1	0	1	0	0	0
4	A7	A6	1	0	0	0	0	0
3	A7	A6	0	1	1	0	0	0
2	A7	A6	0	1	0	0	0	0
1	A7	A6	0	0	1	0	0	0
0	A7	A6	0	0	0	0	0	0

During the third INTA pulse the higher address of the appropriate service routine, which was programmed as byte 2 of the initialization sequence (A_8 – A_{15}), is enabled onto the bus.

Content of Third Interrupt Vector Byte

D7	D6	D5	D4	D3	D2	D1	D0
A15	A14	A13	A12	A11	A10	A9	A8

80C86, 80C88

80C86, 80C88 mode is similar to MCS-80 mode except that only two Interrupt Acknowledge cycles are issued by the processor and no CALL opcode is sent to the processor. The first interrupt acknowledge cycle is similar to that of MCS-80, 85 systems in that the 82C59A-2 uses it to internally freeze the state of the interrupts for priority resolution and as a master it issues the interrupt code on the cascade lines at the end of the INTA pulse. On this first cycle it does not issue any data to the processor and leaves its data bus buffers disabled. On the second interrupt acknowledge cycle in 80C86, 80C88 mode the master (or slave if so programmed) will send a byte of data to the processor with the acknowledged interrupt code composed as follows (note the state of the ADI mode control is ignored and A_5–A_{11} are unused in 80C86, 80C88 mode):

Content of Interrupt Vector Byte for 80C86, 80C88 System Mode

	D7	D6	D5	D4	D3	D2	D1	D0
IR7	T7	T6	T5	T4	T3	1	1	1
IR6	T7	T6	T5	T4	T3	1	1	0
IR5	T7	T6	T5	T4	T3	1	0	1
IR4	T7	T6	T5	T4	T3	1	0	0
IR3	T7	T6	T5	T4	T3	0	1	1
IR2	T7	T6	T5	T4	T3	0	1	0
IR1	T7	T6	T5	T4	T3	0	0	1
IR0	T7	T6	T5	T4	T3	0	0	0

PROGRAMMING THE 82C59A-2

The 82C59A-2 accepts two types of command words generated by the CPU:

1. *Initialization Command Words (ICWs)*: Before normal operation can begin, each 82C59A-2 in the system must be brought to a starting point — by a sequence of 2 to 4 bytes timed by $\overline{WR}$ pulses.
2. *Operation Command Words (OCWs)*: These are the command words which command the 82C59A-2 to operate in various interrupt modes. These modes are:
   a. Fully nested mode
   b. Rotating priority mode
   c. Special mask mode
   d. Polled mode

The OCWs can be written into the 82C59A-2 anytime after initialization.

INITIALIZATION COMMAND WORDS (ICWS)

GENERAL

Whenever a command is issued with A0 = 0 and D4 = 1, this is interpreted as Initialization Command Word 1 (ICW1). ICW1 starts the initialization sequence during which the following automatically occur.

a. The edge sense circuit is reset, which means that following initialization, an interrupt request (IR) input must make a low-to-high transition to generate an interrupt.
b. The Interrupt Mask Register is cleared.
c. IR7 input is assigned priority 7.
d. The slave mode address is set to 7.
e. Special Mask Mode is cleared and Status Read is set to IRR.
f. If IC4 = 0, then all functions selected in ICW4 are set to zero. (Non-Buffered mode*, no Auto-EOI, MCS-80, 85 system).

***NOTE:**
Master/Slave in ICW4 is only used in the buffered mode.

INITIALIZATION COMMAND WORDS 1 AND 2 (ICW1, ICW2)

A_5–A_{15}: *Page starting address of service routines.* In an MCS 80/85 system, the 8 request levels will generate CALLs to 8 locations equally spaced in memory. These can be programmed to be spaced at intervals of 4 or 8 memory locations, thus the 8 routines will occupy a page of 32 or 64 bytes, respectively.

The address format is 2 bytes long (A_0–A_{15}). When the routine interval is 4, A_0–A_4 are automatically inserted by the 82C59A-2, while A_5–A_{15} are programmed externally. When the routine interval is 8, A_0–A_5 are automatically inserted by the 82C59A-2, while A_6–A_{15} are programmed externally.

The 8-byte interval will maintain compatibility with current software, while the 4-byte interval is best for a compact jump table.

In an 80C86, 80C88 system A_{15}–A_{11} are inserted in the five most significant bits of the vectoring

byte and the 82C59A-2 sets the three least significant bits according to the interrupt level. $A_{10}-A_5$ are ignored and ADI (Address Interval) has no effect:

LTIM: If LTIM = 1, then the 82C59A-2 will operate in the level interrupt mode. Edge detect logic on the interrupt inputs will be disabled.

ADI: CALL address interval. ADI = 1 then interval = 4; ADI = 0 then interval = 8.

SNGL: Single. Means that this is the only 82C59A-2 in the system. If SNGL = 1 no ICW3 will be issued.

IC4: If this bit is set — ICW4 has to be read. If ICW4 is not needed, set IC4 = 0.

INITIALIZATION COMMAND WORD 3 (ICW3)

This word is read only when there is more than one 82C59A-2 in the system and cascading is used, in which case SNGL = 0. It will load the 8-bit slave register. The functions of this register are:

a. In the master mode (either when SP = 1, or in buffered mode when M/S = 1 in ICW4) a "1" is set for each slave in the system. The master then will release byte 1 of the call sequence (for MCS-80/85 system) and will enable the corresponding slave to release bytes 2 and 3 (for 80C86, 80C88 only byte 2) through the cascade lines.

b. In the slave mode (either when $\overline{SP}$ = 0, or if BUF = 1 and M/S = 0 in ICW4) bits 2-0 identify the slave. The slave compares its cascade input with these bits and, if they are equal, bytes 2 and 3 of the call sequence (or just byte 2 for 80C86, 80C88 are released by it on the Data Bus.

INITIALIZATION COMMAND WORD 4 (ICW4)

SFNM: If SFNM = 1 the special fully nested mode is programmed.

BUF: If BUF = 1 the buffered mode is programmed. In buffered mode $\overline{SP}/\overline{EN}$ becomes an enable output and the master/slave determination is by M/S.

M/S: If buffered mode is selected: M/S = 1 means the 82C59A-2 is programmed to be a master, M/S = 0 means the 82C59A-2 is programmed to be a slave. If BUF = 0, M/S has no function.

AEOI: If AEOI = 1 the automatic end of interrupt mode is programmed.

μPM: Microprocessor mode: μPM = 0 sets the 82C59A-2 for MCS-80, 85 system operation, μPM = 1 sets the 82C59A-2 for 80C86 system operation.

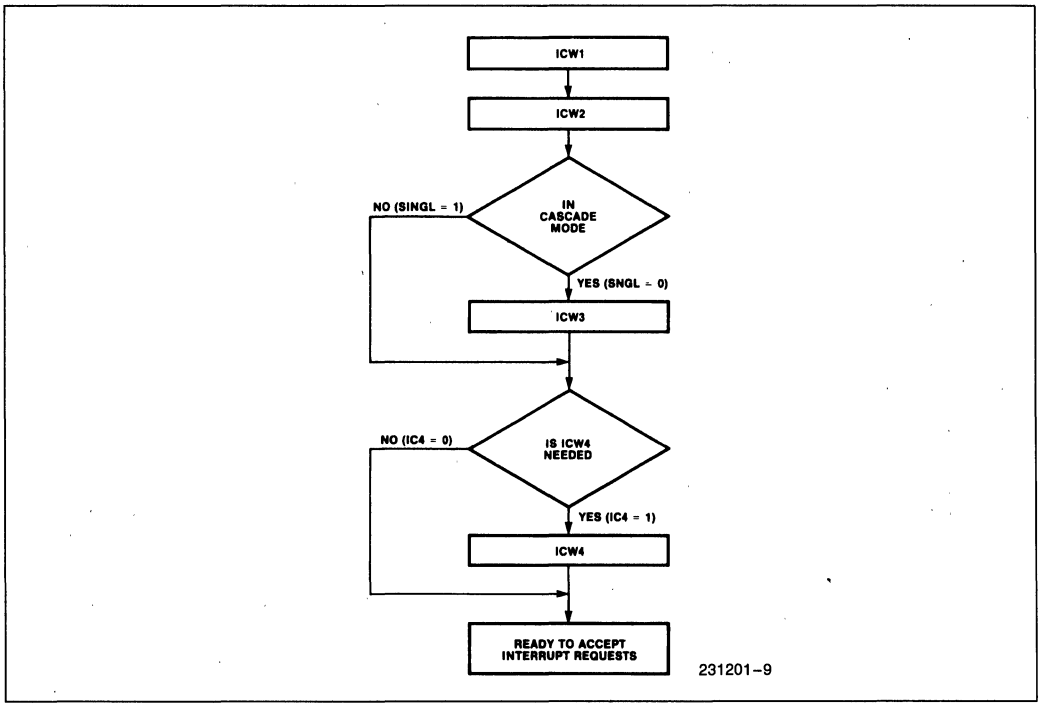
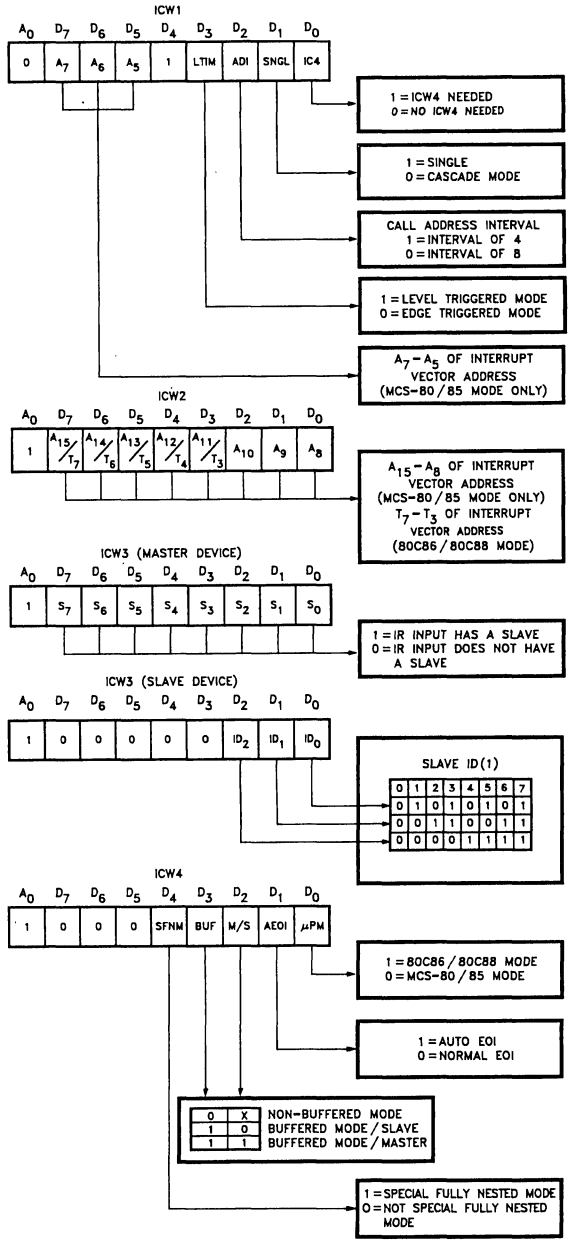


Figure 6. Initialization Sequence



NOTE:
Slave ID is equal to the corresponding master IR input.

Figure 7. Initialization Command Word Format

5

OPERATION COMMAND WORDS (OCWs)

After the initialization Command Words (ICWs) are programmed into the 82C59A-2, the chip is ready to accept interrupt requests at its input lines. However, during the 82C59A-2 operation, a selection of algorithms can command the 82C59A-2 to operate in various modes through the Operation Command Words (OCWs).

OPERATION CONTROL WORDS (OCWs)

OCW1

A0	D7	D6	D5	D4	D3	D2	D1	D0
1	M7	M6	M5	M4	M3	M2	M1	M0

OCW2

0	R	SL	EOI	0	0	L2	L1	L0

OCW3

0	0	ESMM	SMM	0	1	P	RR	RIS

OPERATION CONTROL WORD 1 (OCW1)

OCW1 sets and clears the mask bits in the interrupt Mask Register (IMR). M_7-M_0 represent the eight mask bits. M = 1 indicates the channel is masked (inhibited), M = 0 indicates the channel is enabled.

OPERATION CONTROL WORD 2 (OCW2)

R, SL, EOI — These three bits control the Rotate and End of Interrupt modes and combinations of the two. A chart of these combinations can be found on the Operation Command Word Format.

L_2, L_1, L_0—These bits determine the interrupt level acted upon when the SL bit is active.

OPERATION CONTROL WORD 3 (OCW3)

ESMM — Enable Special Mask Mode. When this bit is set to 1 it enables the SMM bit to set or reset the Special Mask Mode. When ESMM = 0 the SMM bit becomes a "don't care".

SMM — Special Mask Mode. If ESMM = 1 and SMM = 1 the 82C59A-2 will enter Special Mask Mode. If ESMM = 1 and SMM = 0 the 82C59A-2 will revert to normal mask mode. When ESMM = 0, SMM has no effect.

FULLY NESTED MODE

This mode is entered after initialization unless another mode is programmed. The interrupt requests are ordered in priority form 0 through 7 (0 highest). When an interrupt is acknowledged the highest priority request is determined and its vector placed on the bus. Additionally, a bit of the Interrupt Service register (ISO-7) is set. This bit remains set until the microprocessor issues an End of Interrupt (EOI) command immediately before returning from the service routine, or if AEOI (Automatic. End of Interrupt) bit is set, until the trailing edge of the last INTA. While the IS bit is set, all further interrupts of the same or lower priority are inhibited, while higher levels will generate an interrupt (which will be acknowledged only if the microprocessor internal interrupt enable flip-flop has been re-enabled through software).

After the initialization sequence, IR0 has the highest priority and IR7 the lowest. Priorities can be changed, as will be explained, in the rotating priority mode.

END OF INTERRUPT (EOI)

The In Service (IS) bit can be reset either automatically following the trailing edge of the last in sequence $\overline{\text{INTA}}$ pulse (when AEOI bit in ICW4 is set) or by a command word that must be issued to the 82C59A-2 before returning from a service routine (EOI command). An EOI command must be issued twice if in the Cascade mode, once for the master and once for the corresponding slave.

There are two forms of EOI command: Specific and Non-Specific. When the 82C59A-2 is operated in modes which preserve the fully nested structure, it can determine which IS bit to reset on EOI. When a Non-Specific EOI command is issued the 82C59A-2 will automatically reset the highest IS bit of those that are set, since in the fully nested mode the highest IS level was necessarily the last level acknowledged and serviced. A non-specific EOI can be issued with OCW2 (EOI = 1, SL = 0, R = 0).

When a mode is used which may disturb the fully nested structure, the 82C59A-2 may no longer be able to determine the last level acknowledged. In this case a Specific End of Interrupt must be issued which includes as part of the command the IS level to be reset. A specific EOI can be issued with OCW2 (EOI = 1, SL = 1, R = 0, and L0-L2 is the binary level of the IS bit to be reset).

It should be noted that an IS bit that is masked by an IMR bit will not be cleared by a non-specific EOI if the 82C59A-2 is in the Special Mask Mode.

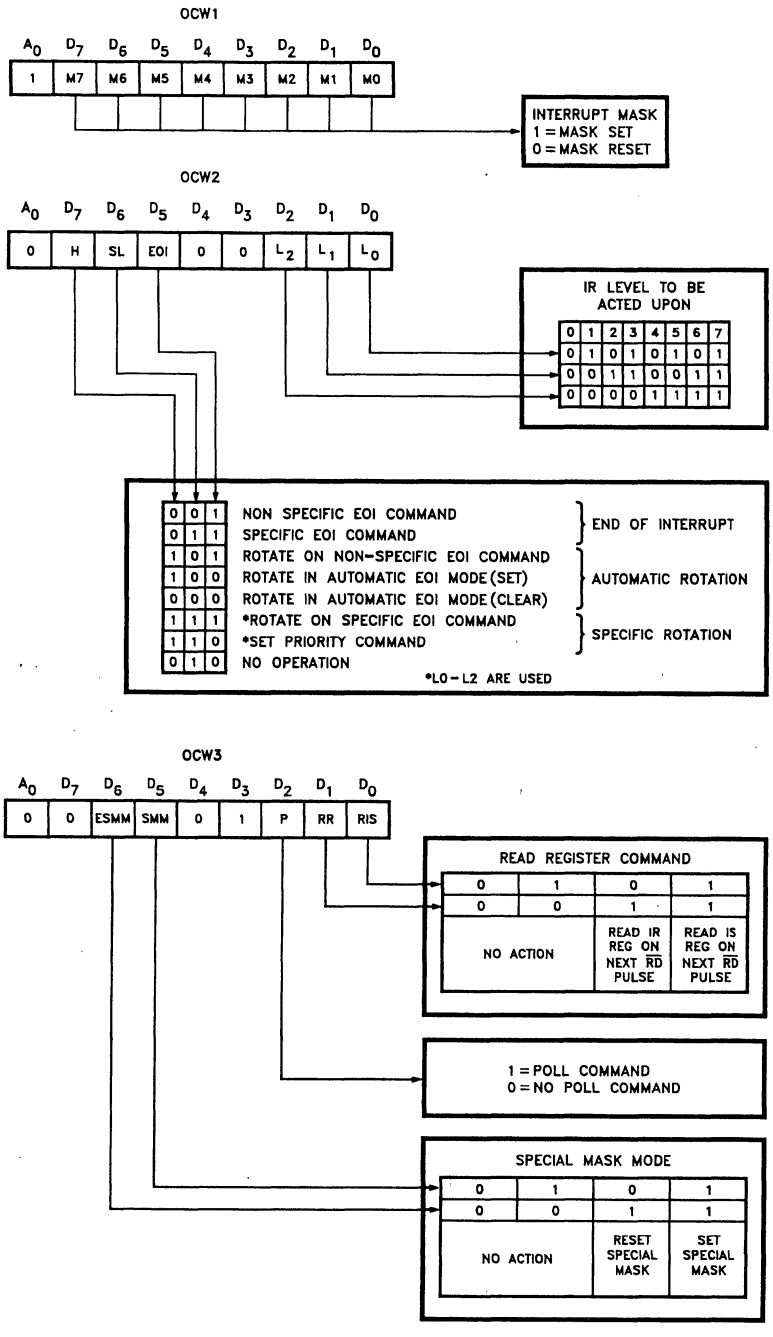


Figure 8. Operation Command Word Format

5

AUTOMATIC END OF INTERRUPT (AEOI) MODE

If AEOI = 1 in ICW4, then the 82C59A-2 will operate in AEOI mode continuously until reprogrammed by ICW4. In this mode the 82C59A-2 will automatically perform a non-specific EOI operation at the trailing edge of the last interrupt acknowledge pulse (third pulse in MCS-80/85, second in 80C86/88). Note that from a system standpoint, this mode should be used only when a nested multilevel interrupt structure is not required within a single 82C59A.

The AEOI mode can only be used in a master 82C59A and not a slave.

AUTOMATIC ROTATION

(Equal Priority Devices)

In some applications there are a number of interrupting devices of equal priority. In this mode a device, after being serviced, receives the lowest priority, so a device requesting an interrupt will have to wait, in the worst case until each of 7 other devices are serviced at most *once*. For example, if the priority and "in service" status is:

Before Rotate (IR4 the highest priority requiring service)

	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
"IS" Status	0	1	0	1	0	0	0	0

	Lowest Priority ↓							Highest Priority ↓
Priority Status	7	6	5	4	3	2	1	0

After Rotate (IR4 was serviced, all other priorities rotated correspondingly)

	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
"IS" Status	0	1	0	0	0	0	0	0

	Highest Priority ↓			Lowest Priority ↓				
Priority Status	2	1	0	7	6	5	4	3

There are two ways to accomplish Automatic Rotation using OCW2, the Rotation on Non-Specific EOI Command (R = 1, SL = 0, EOI = 1) and the Rotate in Automatic EOI Mode which is set by (R = 1, SL = 0, EOI = 0) and cleared by (R = 0, SL = 0, EOI = 0).

SPECIFIC ROTATION

(Specific Priority)

The programmer can change priorities by programming the bottom priority and thus fixing all other priorities; i.e., if IR5 is programmed as the bottom priority device, then IR6 will have the highest one.

The Set Priority command is issued in OCW2 where: R = 1, SL = 1; LO-L2 is the binary priority level code of the bottom priority device.

Observe that in this mode internal status is updated by software control during OCW2. However, it is independent of the End of Interrupt (EOI) command (also executed by OCW2). Priority changes can be executed during an EOI command by using the Rotate on Specific EOI command in OCW2 (R = 1, SL = 1, EOI = 1 and LO-L2 = IR level to receive bottom priority).

INTERRUPT MASKS

Each Interrupt Request input can be masked individually by the Interrupt Mask Register (IMR) programmed through OCW1. Each bit in the IMR masks one interrupt channel if it is set (1). Bit 0 masks IR0, Bit 1 masks IR1 and so forth. Masking an IR channel does not affect the other channels operation.

SPECIAL MASK MODE

Some applications may require an interrupt service routine to dynamically alter the system priority structure during its execution under software control. For example, the routine may wish to inhibit lower priority requests for a portion of its execution but enable some of them for another portion.

The difficulty here is that if an interrupt Request is acknowledged and an End of Interrupt command did not reset its IS bit (i.e., while executing a service routine), the 82C59A-2 would have inhibited all lower priority requests with no easy way for the routine to enable them.

That is where the Special Mask Mode comes in. In the special Mask Mode, when a mask bit is set in OCW1, it inhibits further interrupts at that level *and enables* interrupts from *all other* levels (lower as well as higher) that are not masked.

Thus, any interrupts may be selectivity enabled by loading the mask register.

The special Mask Mode is set by OCW3 where: SSMM = 1, SMM = 1, and cleared where SSMM = 1, SMM = 0.

POLL COMMAND

In Poll mode the INT output functions as it normally does. The microprocessor should ignore this output. This can be accomplished either by not connecting the INT output or by masking interrupts within the microprocessor, thereby disabling its interrupt input. Service to devices is achieved by software using a Poll command.

The Poll command is issued by setting P = "1" in OCW3. The 82C59A-2 treats the next $\overline{RD}$ pulse to the 82C59A-2 (i.e., $\overline{RD}$ = 0, $\overline{CS}$ = 0) as an interrupt acknowledge, sets the appropriate IS bit if there is a request, and reads the priority level. Interrupt is frozen from $\overline{WR}$ to $\overline{RD}$.

The word enabled onto the data bus during $\overline{RD}$ is:

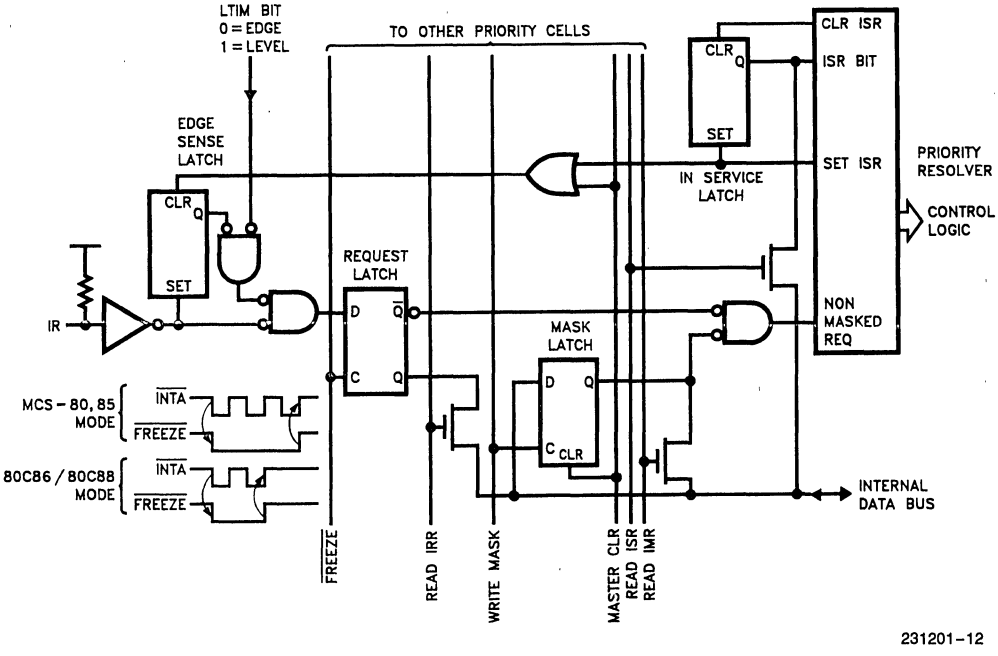
D7	D6	D5	D4	D3	D2	D1	D0
I	—	—	—	—	W2	W1	W0

WO–W2:

Binary code of the highest priority level requesting service.

I: Equal to a "1" if there is an interrupt.

This mode is useful if there is a routine command common to several levels so that the $\overline{INTA}$ sequence is not needed (saves ROM space). Another application is to use the poll mode to expand the number of priority levels to more than 64.



NOTES:
1. Master Clear active only during ICW1
2. Freeze/ is active during $\overline{INTA}$/and poll sequences only
3. Truth Table for D-Latch

C	D	Q	OPERATION
1	Di	Di	FOLLOW
0	X	Qn-1	HOLD

Figure 9. Priority Cell—Simplified Logic Diagram

READING THE 82C59A-2 STATUS

The input status of several internal registers can be read to update the user information on the system. The following registers can be read via OCW3 (IRR and ISR or OCW1 [IMR]).

Interrupt Request Register (IRR): 8-bit register which contains the levels requesting an interrupt to be acknowledged. The highest request level is reset from the IRR when an interrupt is acknowledged. (Not affected by IMR).

In-Service Register (ISR): 8-bit register which contains the priority levels that are being serviced. The ISR is updated when an End of Interrupt Command is issued.

Interrupt Mask Register: 8-bit register which contains the interrupt request lines which are masked.

The IRR can be read when, prior to the RD pulse, a Read Register Command is issued with OCW3 (RR = 1, RIS = 0.)

The ISR can be read when, prior to the RD pulse, a Read Register Command is issued with OCW3 (RR = 1, RIS = 1):

There is no need to write an OCW3 before every status read operation, as long as the status read corresponds with the previous one; i.e., the 82C59A-2 "remembers" whether the IRR or ISR has been previously selected by the OCW3. This is not true when poll is used.

After initialization the 82C59A-2 is set to IRR.

For reading the IMR, no OCW3 is needed. The output data bus will contain the IMR whenever $\overline{RD}$ is active and AO = 1 (OCW1).

Polling overrides status read when P = 1, RR = 1 in OCW3.

EDGE AND LEVEL TRIGGERED MODES

This mode is programmed using bit 3 in ICW1.

If LTIM = '0', an interrupt request will be recognized by a low to high transition on an IR input. The IR input can remain high without generating another interrupt.

If LTIM = '1', an interrupt request will be recognized by a 'high' level on IR Input, and there is no need for an edge detection. The interrupt request must be removed before the EOI command is issued or the CPU interrupt is enabled to prevent a second interrupt from occurring.

The priority cell diagram shows a conceptual circuit of the level sensitive and edge sensitive input circuitry of the 82C59A-2. Be sure to note that the request latch is a transparent D type latch.

In both the edge and level triggered modes the IR inputs must remain high until after the falling edge of the first INTA. If the IR input goes low before this time a DEFAULT IR7 will occur when the CPU acknowledges the interrupt. This can be a useful safeguard for detecting interrupts caused by spurious noise glitches on the IR inputs. To implement this feature the IR7 routine is used for "clean up" simply executing a return instruction, thus ignoring the interrupt. If IR7 is needed for other purposes a default IR7 can still be detected by reading the ISR. A normal IR7 interrupt will set the corresponding ISR bit, a default IR7 won't. If a default IR7 routine occurs during a normal IR7 routine, however, the ISR will remain set. In this case it is necessary to keep track of whether or not the IR7 routine was previously entered. If another IR7 occurs it is a default.

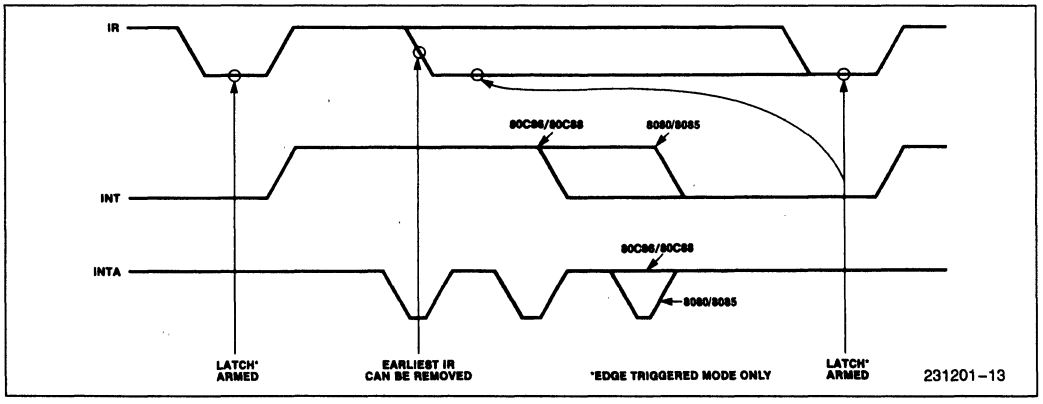


Figure 10. IR Triggering Timing Requirements

THE SPECIAL FULLY NESTED MODE

This mode will be used in the case of a big system where cascading is used, and the priority has to be conserved within each slave. In this case the fully nested mode will be programmed to the master (using ICW4). This mode is similar to the normal nested mode with the following exceptions:

a. When an interrupt request from a certain slave is in service this slave is not locked out from the master's priority logic and further interrupt requests from higher priority IR's within the slave will be recognized by the master and will initiate interrupts to the processor. (In the normal nestled mode a slave is masked out when its request is in service and no higher requests from the same slave can be serviced.)

b. When exiting the Interrupt Service routine the software has to check whether the interrupt serviced was the only one from that slave. This is done by sending a non-specific End of Interrupt (EOI) command to the slave and then reading its In-Service register and checking for zero. If it is empty, a non-specific EOI can be sent to the master too. If not, no EOI should be sent.

BUFFERED MODE

When the 82C59A-2 is used in a large system where bus driving buffers are required on the data bus and the cascading mode is used, there exists the problem of enabling buffers.

The buffered mode will structure the 82C59A-2 to send an enable signal on $\overline{SP}/\overline{EN}$ to enable the buffers. In this mode, whenever the 82C59A-2's data bus outputs are enabled, the $\overline{SP}/\overline{EN}$ output becomes active.

This modification forces the use of software programming to determine whether the 82C59A-2 is a master or a slave. Bit 3 in ICW4 programs the buffered mode, and bit 2 in ICW3 determines whether it is a master or a slave.

CASCADE MODE

The 82C59A-2 can be easily interconnected in a system of one master with up to eight slaves to handle up to 64 priority levels.

The master controls the slaves through the 3 line cascade bus. The cascade bus acts like chip selects to the slaves during the $\overline{INTA}$ sequence.

In a cascade configuration, the slave interrupt outputs are connected to the master interrupt request inputs. When a slave request line is activated and afterwards acknowledged, the master will enable the corresponding slave to release the device routine address during bytes 2 and 3 of INTA. (Byte 2 only for 80C86/80C88).

The cascade bus lines are normally low and will contain the slave address code from the trailing edge of the first INTA pulse to the trailing edge of the third pulse. Each 82C59A-2 in the system must follow a separate initialization sequence and can be programmed to work in a different mode. An EOI command must be issued twice: once for the master and once for the corresponding slave. An address decoder is required to activate the Chip Select (CS) input of each 82C59A-2.

The cascade lines of the Master 82C59A-2 are activated only for slave inputs, non slave inputs leave the cascade line inactive (low).

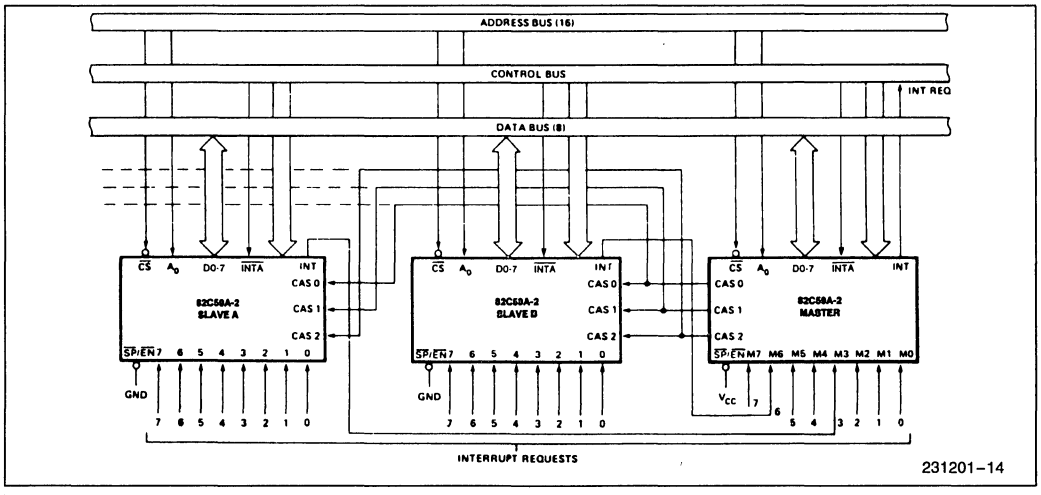


Figure 11. Cascading the 82C59A-2

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias .....0°C to 70°C
Storage Temperature .....−65°C to + 150°C
Supply Voltage (w.r.t. ground) .....−0.5 to 7.0V
Input Voltage (w.r.t. ground) ...−0.5 to V_{CC} + 0.5V
Output Voltage (w.r.t. ground)..−0.5 to V_{CC} + 0.5V
Power Dissipation.....0.9 Watt

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**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.*

D.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±10%

Symbol	Parameter	Min	Max	Units	Test Conditions
I_{CCS}	Standby Supply Current		10	μA	$V_{IN} = V_{CC}$ or GND All IR = GND Outputs Unloaded V_{CC} = 5.5V
I_{CC}	Operating Supply Current		5	mA	(Note)
V_{IH}	Input High Voltage	2.2	V_{CC} + 0.5	V	
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 2.5 mA
V_{OH}	Output High Voltage	3.0 V_{CC} −0.4		V	I_{OH} = −2.5 mA I_{OH} = −100 μA
I_{LI}	Input Leakage Current		±1.0	μA	$0V \le V_{IN} \le V_{CC}$
I_{LO}	Output Leakage Current		±10	μA	$0V \le V_{OUT} \le V_{CC}$
I_{LIR}	IR Input Leakage Current		−300 +10	μA	V_{IN} = 0 $V_{IN} = V_{CC}$

NOTE:
Repeated data input with 80C86-2 timings.

CAPACITANCE T_A = 25°C; V_{CC} = GND = 0V

Symbol	Parameter	Min	Max	Units	Test Conditions	
C_{IN}	Input Capacitance		7	pF		fc = 1 MHz
$C_{I/O}$	I/O Capacitance		20	pF	Unmeasured pins at GND	
C_{OUT}	Output Capacitance		15	pF		

A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±10%

TIMING REQUIREMENTS

Symbol	Parameter	82C59A-2		Units	Test Conditions
		Min	Max		
TAHRL	AO/$\overline{CS}$ Setup to $\overline{RD}$/$\overline{INTA}$ ↓	10		ns	
TRHAX	AO/$\overline{CS}$ Hold after $\overline{RD}$/$\overline{INTA}$ ↑	5		ns	
TRLRH	$\overline{RD}$/$\overline{INTA}$ Pulse Width	160		ns	
TAHWL	AO/$\overline{CS}$ Setup to $\overline{WR}$ ↓	0		ns	
TWHAX	AO/$\overline{CS}$ Hold after $\overline{WR}$ ↑	0		ns	
TWLWH	$\overline{WR}$ Pulse Width	190		ns	
TDVWH	Data Setup to $\overline{WR}$ ↑	160		ns	
TWHDX	Data Hold after $\overline{WR}$ ↑	0		ns	
TJLJH	Interrupt Request Width (Low)	100		ns	(See Note)
TCVIAL	Cascade Setup to Second or Third $\overline{INTA}$ ↓ (Slave Only)	40		ns	
TRHRL	End of $\overline{RD}$ to next $\overline{RD}$ End of $\overline{INTA}$ to next $\overline{INTA}$ within an $\overline{INTA}$ sequence only	160		ns	
TWHWL	End of $\overline{WR}$ to next $\overline{WR}$	190		ns	
*TCHCL	End of Command to next Command (Not same command type) End of $\overline{INTA}$ sequence to next $\overline{INTA}$ sequence.	400		ns	

*Worst case timing for TCHCL in an actual microprocessor system is typically much greater than 400 ns (i.e. 8085A = 1.6 μs, 8085-A2 = 1 μs, 80C86 = 1 μs, 80C86-2 = 625 ns)

NOTE:
This is the low time required to clear the input latch in the edge triggered mode.

5

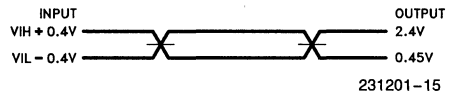
TIMING RESPONSES

Symbol	Parameter	8259A-2		Units	Test Conditions**
		Min	Max		
TRLDV	Data Valid from $\overline{RD}/\overline{INTA}\downarrow$		120	ns	1
TRHDZ	Data Float after $\overline{RD}/\overline{INTA}\uparrow$	10	85	ns	2
TJHIH	Interrupt Output Delay		300	ns	1
TIALCV	Cascade Valid from First $\overline{INTA}\downarrow$ (Master Only)		360	ns	1
TRLEL	Enable Active from $\overline{RD}\downarrow$ or $\overline{INTA}\downarrow$		110	ns	1
TRHEH	Enable Inactive from $\overline{RD}\uparrow$ or $\overline{INTA}\uparrow$		150	ns	1
TAHDV	Data Valid from Stable Address		200	ns	1
TCVDV	Cascade Valid to Valid Data		200	ns	1

**Test Condition Definition Table

TEST CONDITION	V1	R1	R2	C1
1	1.7V	523Ω	OPEN	100 pf
2	4.5V	1.8 kΩ	1.8 kΩ	30 pf

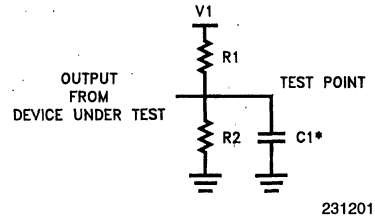
A.C. TESTING INPUT, OUTPUT WAVEFORM



231201-15

A.C. Testing: All input signals must switch between V_{IL} − 0.4V and V_{IH} + 0.4V. Input Rise and Fall Times must be ≤ 15 ns. All timing measurements are made at 2.4V and 0.45V.

A.C. TESTING LOAD CIRCUIT

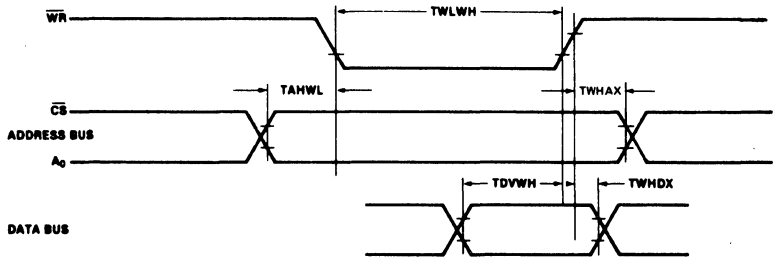


231201-16

*Includes Stray and Jig Capacitance

WAVEFORMS

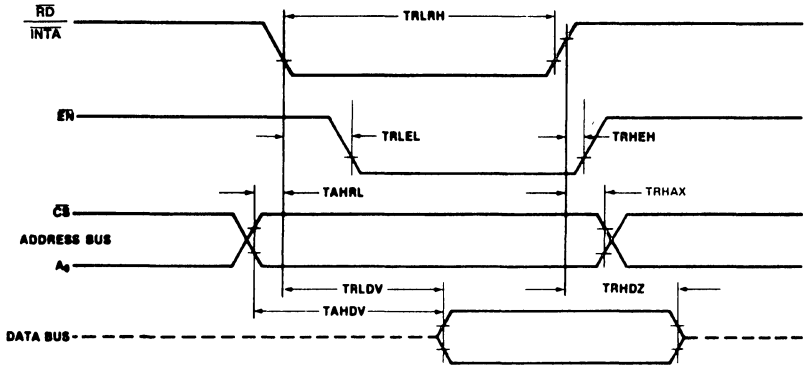
WRITE



231201-17

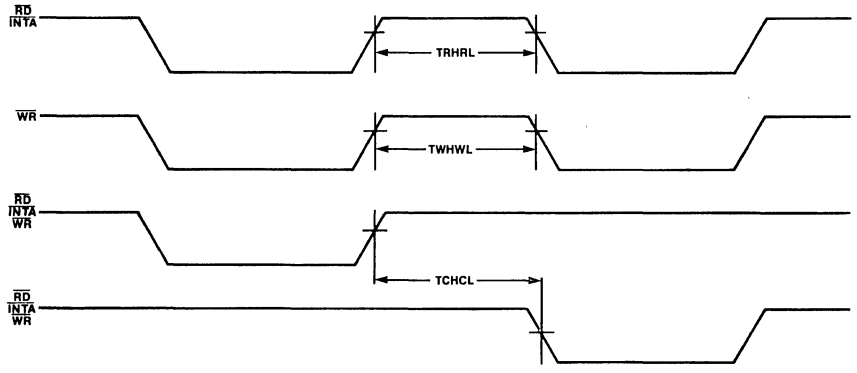
WAVEFORMS (Continued)

READ/INTA



231201-18

OTHER TIMING

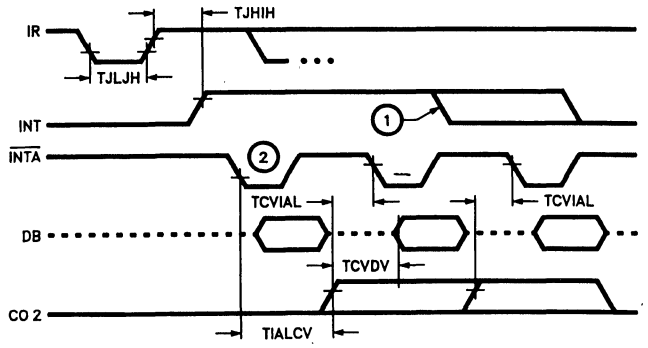


231201-19

5

WAVEFORMS (Continued)

$\overline{\text{INTA}}$ SEQUENCE



231201-20

NOTES:
1. Interrupt output must remain HIGH at least until leading edge of first INTA.
2. Cycle 1 in 80C86 and 80C88 systems, the Data Bus is not active.

DATA SHEET REVISION REVIEW

The following changes have been made since revision 003 of the 82C59A-2 data sheet.

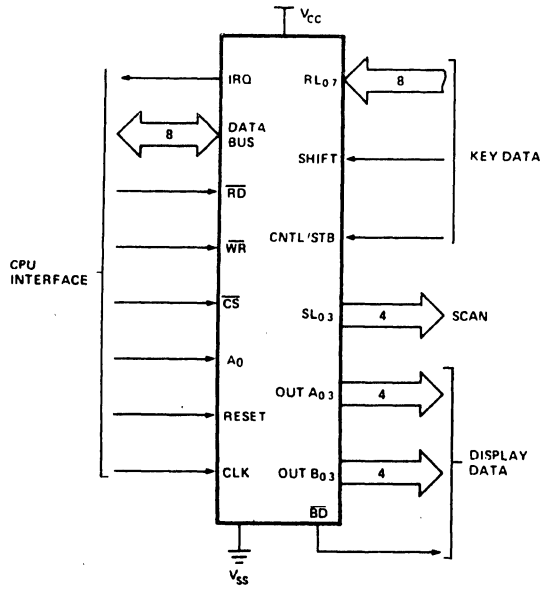
1. Preliminary was removed.
2. A reference to PLCC packaging was removed.
3. The first paragraph of the Poll Command section was rewritten to clarify the status of the INT pin.
4. A paragraph was added to the Interrupt Sequence section to indicate the status of the INT pin during multiple interrupts.

8279/8279-5
PROGRAMMABLE KEYBOARD/DISPLAY INTERFACE

- **Simultaneous Keyboard Display Operations**
- **Scanned Keyboard Mode**
- **Scanned Sensor Mode**
- **Strobed Input Entry Mode**
- **8-Character Keyboard FIFO**
- **2-Key Lockout or N-Key Rollover with Contact Debounce**
- **Dual 8- or 16-Numerical Display**
- **Single 16-Character Display**
- **Right or Left Entry 16-Byte Display RAM**
- **Mode Programmable from CPU**
- **Programmable Scan Timing**
- **Interrupt Output on Key Entry**
- **Available in EXPRESS**
  - **— Standard Temperature Range**
  - **— Extended Temperature Range**

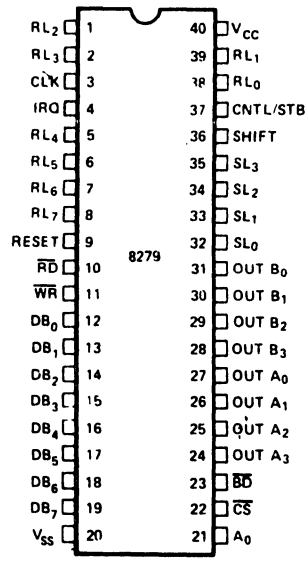
The Intel 8279 is a general purpose programmable keyboard and display I/O interface device designed for use with Intel microprocessors. The keyboard portion can provide a scanned interface to a 64-contact key matrix. The keyboard portion will also interface to an array of sensors or a strobed interface keyboard, such as the hall effect and ferrite variety. Key depressions can be 2-key lockout or N-key rollover. Keyboard entries are debounced and strobed in an 8-character FIFO. If more than 8 characters are entered, overrun status is set. Key entries set the interrupt output line to the CPU.

The display portion provides a scanned display interface for LED, incandescent, and other popular display technologies. Both numeric and alphanumeric segment displays may be used as well as simple indicators. The 8279 has 16x8 display RAM which can be organized into dual 16x4. The RAM can be loaded or interrogated by the CPU. Both right entry, calculator and left entry typewriter display formats are possible. Both read and write of the display RAM can be done with auto-increment of the display RAM address.



290123-1

Figure 1. Logic Symbol



290123-2

Figure 2. Pin Configuration

5

The complete document for this product is available from Intel's Literature Center at 1-800-548-4725.

September 1987
Order Number: 290123-002

NORTH AMERICAN SALES OFFICES

ARIZONA

Intel Corp.
410 North 44th Street
Suite 470
Phoenix 85008
Tel: (800) 628-8686
FAX: (602) 244-0446

CALIFORNIA

Intel Corp.
26707 W. Agoura Road
Suite 203
Calabasas, CA 91302
Tel: (800) 628-8686
FAX: (818)-880-1820

Intel Corp.
3550 Watt Avenue
Suite 140
Sacramento 95821
Tel: (800) 628-8686
FAX: (916) 979-7011

Intel Corp.
9655 Granite Ridge Drive
3rd Floor
Suite 4A
San Diego 92123
Tel: (800) 628-8686
FAX: (619) 467-2460

Intel Corp.
1781 Fox Drive
San Jose 95131
Tel: (800) 628-8686
FAX: (408) 441-9540

Intel Corp.
1551 North Tustin Avenue
Suite 800
Santa Ana 92701
Tel: (800) 628-8686
TWX: (910) 595-1114
FAX: (714) 541-9157

Intel Corp.
514 Via de la Valle
Suite 208-RCO
Solana Beach 92075

Intel Corp.
1960 E. Grand Avenue
Suite 150
El Segundo, CA 90245
Tel: (800) 628-8686
FAX: (310) 640-7133

COLORADO

Intel Corp.
600 South Cherry Street
Suite 700
Denver 80222
Tel: (800) 628-8686
TWX: 910-931-2289
FAX: (303) 322-8670

CONNECTICUT

Intel Corp.
40 Old Ridgebury Road
Suite 311
Danbury 06811
Tel: (800) 628-8686
FAX: (203) 778-2168

FLORIDA

Intel Corp.
600 West Hillsboro Blvd.
Suite 348
Deerfield Beach 33441
Tel: (800) 628-8686
FAX: (305) 421-2444

Intel Corp.
2250 Lucien Way
Suite 100
Suite 8
Maitland 32751
Tel: (800) 628-8686
FAX: (407) 660-1283

GEORGIA

Intel Corp.
20 Technology Park
Suite 150
Norcross 30092
Tel: (800) 628-8686
FAX: (404) 448-0875

IDAHO

Intel Corp.
910 W. Main Street
Suite 236
Boise 83702
Tel: (800) 628-8686
FAX: (208) 331-2295

ILLINOIS

Intel Corp.
300 North Martingale Road
Suite 400
Schaumburg 60173
Tell: (800) 628-8686
FAX: (708) 605-9762

INDIANA

Intel Corp.
8041 Knue Road
Indianapolis 46250
Tel: (800) 628-8686
FAX: (317) 577-4939

MARYLAND

Intel Corp.
131 National Bus. Pkwy
Suite 200
Annapolis Junction 20701
Tel: (800) 628-8686
FAX: (301) 206-3678

MASSACHUSETTS

Intel Corp.
Nagog Park
125 Nagog Park
Acton 01720
Tel: (800) 628-8686
FAX: (508) 266-3867

MICHIGAN

Intel Corp.
32255 North Western Hwy.
Suite 212, Tri Atria
Farmington Hills 48334
Tel: (800) 628-8686
FAX: (313) 851-8770

MINNESOTA

Intel Corp.
3500 West 80th Street
Suite 360
Bloomington 55431
Tel: (800) 628-8686
TWX: 910-576-2867
FAX: (612) 831-6497

NEW JERSEY

Intel Corp.
2001 Route 46
Suite 310
Parsippany 07054
Tel: (800) 628-8686
FAX: (201) 402-4893

Intel Corp.
Lincroft Center
125 Half Mile Road
Red Bank 07701
Tel: (800) 628-8686
FAX: (908) 747-0983

NEW YORK

Intel Corp.
850 Cross Keys Office Pk
Fairport 14450
Tel: (800) 628-8686
TWX: 510-253-7391
FAX: (716) 223-2561

Intel Corp.
2950 Expressway Drive
Islandia 11722
Tel: (800) 628-8686
TWX: 510-227-6236
FAX: (516) 348-7939

OHIO

Intel Corp.
56 Milford Drive
Suite 205
Hudson 44236
Tel: (800) 628-8686
FAX: (216) 528-1026

†Intel Corp.
3401 Park Center Drive
Suite 220
Dayton 45414
Tel: (800) 628-8686
TWX: 810-450-2528
FAX: (513) 890-8658

OKLAHOMA

Intel Corp.
6801 North Broadway
Suite 115
Oklahoma City 73162
Tel: (800) 628-8686
FAX: (405) 840-9819

OREGON

Intel Corp.
15254 NW Greenbrier
Pkwy
Building B
Beaverton 97006
Tel: (800) 628-8686
TWX: 910-467-8741
FAX: (503) 645-8181

PENNSYLVANIA

Intel Corp.
925 Harvest Drive
Suite 200
Blue Bell 19422
Tel: (800) 628-8686
FAX: (215) 641-0785

SOUTH CAROLINA

Intel Corp.
7403 Parklane Road
Suite 4
Columbia 29223
Tel: (800) 628-8686
FAX: (803) 788-7999

Intel Corp.
100 Executive Center Dr
Suite 109, B183
Greenville 29615
Tel: (800) 628-8686
FAX: (803) 297-3401

TEXAS

Intel Corp.
8911 Capital of Texas Hwy
Suite 4230
Austin 78759
Tel: (800) 628-8686
FAX: (512) 338-9335

Intel Corp.
5000 Quorum Drive
Suite 750
Dallas 75240
Tel: (800) 628-8686
FAX: (214) 233-1325

Intel Corp.
20405 State Hwy 249
Suite 880
Houston 77070
Tel: (800) 628-8686
TWX: 910-881-2490
FAX: (713) 376-2891

UTAH

Intel Corp.
428 East 6400 South
Suite 135
Murray 84107
Tel: (800) 628-8686
FAX: (801) 268-1457

WASHINGTON

Intel Corp.
2800 156th Avenue SE
Suite 105
Bellevue 98007
Tel: (800) 628-8686
FAX: (206) 746-4495

WISCONSIN

Intel Corp.
400 North Executive Drive
Suite 401
Brookfield 53005
Tel: (800) 628-8686
FAX: (414) 789-2746

CANADA

BRITISH COLUMBIA

Intel of Canada, Ltd.
999 Canada Place
Suite 404
Suite 11
Vancouver V6C 3E2
Tel: (800) 628-8686
FAX: (604) 844-2813

ONTARIO

Intel of Canada, Ltd.
2650 Queensview Drive
Suite 250
Ottawa K2B 8H6
Tel: (800) 628-8686
FAX: (613) 820-5936

Intel of Canada, Ltd.
190 Attwell Drive
Suite 500
Rexdale M9W 6H8
Tel: (800) 628-8686
FAX: (416) 675-2438

QUEBEC

Intel of Canada, Ltd.
1 Rue Holiday, Tour West
Suite 320
Pt. Claire H9R 5N3
Tel: (800) 628-8686
FAX: 514-694-0064

NORTH AMERICAN DISTRIBUTORS

ALABAMA

Anthem Electronics
600 Boulevard South
Suite 104F & H
Huntsville 35802
Tel: (205) 890-0302

Arrow/Schweber Electronics
1015 Henderson Road
Huntsville 35805
Tel: (205) 837-6955
FAX: (205) 721-1581

Hall-Mark Computer
4890 University Square
Huntsville 35816
Tel: (800) 409-1483

Hamilton Hallmark
4890 University Square
Suite 1
Huntsville 35816
Tel: (205) 837-8706
FAX: (205) 830-2565

MTI Systems Sales
4950 Corporate Drive
Suite 120
Huntsville 35805
Tel: (205) 830-9526
FAX: (205) 830-9557

Pioneer Technologies Group
4835 University Square
Suite 5
Huntsville 35805
Tel: (205) 837-9300
FAX: (205) 837-9358

Wyle Electronics
7800 Governers Dr., W.
Tower Building, 2nd Floor
Huntsville 35807
Tel: (205) 830-1119
FAX: (205) 830-1520

ARIZONA

Alliance Electronics
7550 East Redfield Rd
Scottsdale 85260
Tel: (602) 261-7988

Anthem Electronics
1555 West 10th Place
Suite 101
Tempe 85281
Tel: (602) 966-6600
FAX: (602) 966-4826

Arrow/Schweber Electronics
2415 West Erie Drive
Tempe 85282
Tel: (602) 431-0030
FAX: (602) 431-9555

Avnet Computer
1626 South Edwards Dr
Tempe 85281
Tel: (800) 426-7999

Hall-Mark Computer
4637 South 37th Place
Phoenix 85040
Tel: (800) 409-1483

Pioneer Standard
1438 West Broadway
Suite B-140
Tempe 85282
Tel: (602) 350-9335

Hamilton Hallmark
4637 South 36th Place
Phoenix 85040
Tel: (602) 437-1200
FAX: (602) 437-2348

Wyle Electronics
4141 East Raymond
Phoenix 85040
Tel: (602) 437-2088
FAX: (602) 437-2124

CALIFORNIA

Anthem Electronics
9131 Oakdale Avenue
Chatsworth 91311
Tel: (818) 775-1333
FAX: (818) 775-1302

Anthem Electronics
1 Oldfield Drive
Irvine 92718-2809
Tel: (714) 768-4444
FAX: (714) 768-6456

Anthem Electronics
580 Menlo Drive
Suite 8
Rocklin 95677
Tel: (916) 624-9744
FAX: (916) 624-9750

Anthem Electronics
9369 Carroll Park Drive
San Diego 92121
Tel: (619) 453-9005
FAX: (619) 546-7893

Anthem Electronics
1160 Ridder Park Drive
San Jose 95131
Tel: (408) 453-1200
FAX: (408) 441-4504

Arrow/Schweber Electronics
26707 West Agoura Road
Calabasas 91302
Tel: (818) 880-9686
FAX: (818) 772-8930

Arrow/Schweber Electronics
48834 Kato Road
Suite 103
Fremont 94538
Tel: (510) 490-9477
FAX: (510) 490-1084

Arrow/Schweber Electronics
6 Cromwell
Suite 100
Irvine 92718
Tel: (714) 581-4622
FAX: (714) 454-4206

Arrow/Schweber Electronics
9511 Ridgehaven Court
San Diego 92123
Tel: (619) 565-4800
FAX: (619) 279-8062

Arrow/Schweber Electronics
1180 Murphy Avenue
San Jose 95131
Tel: (408) 441-9700
FAX: (408) 453-4810

Avnet Computer
1 Mauchley
Irvine 92718
Tel: (800) 426-7999

Avnet Computer
371 Van Ness Way
Torrance 90501
Tel: (800) 426-7999

Avnet Computer
15950 Bernardo Ctr Dr
Suite 6
San Diego 92127
Tel: (800) 426-7999

Avnet Computer
1175 Bordeaux Drive
Suite A
Sunnyvale 94089
Tel: (800) 426-7999

Hall-Mark Computer
21150 Califa Street
Woodland Hills 91367
Tel: (800) 409-1483

Hall-Mark Computer
15950 Bernardo Ctr Dr
Suite C
San Diego 92127
Tel: (800) 409-1483

Hall-Mark Computer
1175 Bordeaux Drive
Sunnyvale 94089
Tel: (800) 409-1483

Hall-Mark Computer
1 Mauchly
Irvine 92718
Tel: (800) 409-1483

Hall-Mark Computer
580 Menlo Drive
Suite 2
Rocklin 95765
Tel: (800) 409-1483

Hamilton Hallmark
3170 Pullman Street
Costa Mesa 92626
Tel: (714) 641-4100
FAX: (714) 641-4122

Hamilton Hallmark
2105 Lundy Avenue
San Jose 95131
Tel: (408) 435-3500
FAX: (408) 435-3720

Hamilton Hallmark
4545 Viewridge Avenue
San Diego 92123
Tel: (619) 571-7540
FAX: (619) 277-6136

Hamilton Hallmark
21150 Califa Street
Woodland Hills 91367
Tel: (818) 594-0404
FAX: (818) 594-8234

Hamilton Hallmark
580 Menlo Drive
Suite 2
Rocklin 95762
Tel: (916) 624-9781
FAX: (916) 961-0922

Pioneer Standard
5126 Clareton Drive
Suite 106
Agoura Hills 91301
Tel: (818) 865-5800

Pioneer Standard
217 Technology Drive
Suite 110
Irvine 92718
Tel: (714) 753-5090

Pioneer Technologies Group
134 Rio Robles
San Jose 95134
Tel: (408) 954-9100
FAX: (408) 954-9113

Pioneer Standard
4370 La Jolla Village Drive
San Diego 92122
Tel: (619) 546-4906

Wyle Electronics
15370 Barranca Pkwy
Irvine 92713
Tel: (714) 753-9953
FAX: (714) 753-9877

Wyle Electronics
15360 Barranca Pkwy
Suite 200
Irvine 92713
Tel: (714) 753-9953
FAX: (714) 753-9877

Wyle Electronics
2951 Sunrise Blvd.
Suite 175
Rancho Cordova 95742
Tel: (916) 638-5282
FAX: (916) 638-1491

Wyle Electronics
9525 Chesapeake Dr.
San Diego 92123
Tel: (619) 565-9171
FAX: (619) 365-0512

Wyle Electronics
3000 Bowers Avenue
Santa Clara 95051
Tel: (408) 727-2500
FAX: (408) 727-5896

Wyle Electronics
17872 Cowan Avenue
Irvine 92714
Tel: (714) 863-9953
FAX: (714) 263-0473

Wyle Electronics
26010 Mureau Road
Suite 150
Calabasas 91302
Tel: (818) 880-9000
FAX: (818) 880-5510

Zeus Arrow Electronics
6276 San Ignacio Avenue
Suite E
San Jose 95119
Tel: (408) 629-4689
FAX: (408) 629-4792

Zeus Arrow Electronics
6 Cromwell Street
Suite 100
Irvine 92718
Tel: (714) 581-4622
FAX: (714) 454-4355

COLORADO

Anthem Electronics
373 Inverness Dr. S.
Englewood 80112
Tel: (303) 790-4500
FAX: (303) 790-4532

Arrow/Schweber Electronics
61 Inverness Dr East
Suite 105
Englewood 80112
Tel: (303) 799-0258
FAX: (303) 799-0730

Avnet Computer
9605 Maroon Circle
Englewood 80111
Tel: (800) 426-7999

Hall-Mark Computer
9605 Maroon Circle
Englewood 80111
Tel: (800) 409-1483

Hamilton Hallmark
12503 East Euclid Dr
Suite 20
Englewood 80111
Tel: (303) 790-1662
FAX: (303) 790-4991

Hamilton Hallmark
710 Wooten Road
Suite 28
Colorado Springs 80915
Tel: (719) 637-0055
FAX: (719) 637-0088

Pioneer Technologies
5600 Greenwood Plaza Blvd.
Suite 201
Englewood 80111
Tel: (303) 773-8090

Wyle Electronics
451 East 124th Avenue
Thornton 80241
Tel: (303) 457-9953
FAX: (303) 457-4831

CONNECTICUT

Anthem Electronics
61 Mattatuck Heights Road
Waterburg 06705
Tel: (203) 575-1575
FAX: (203) 596-3232

Arrow/Schweber Electronics
860 N. Main St. Ext.
Wallingford 06492
Tel: (203) 265-7741
FAX: (203) 265-7988

Hall-Mark Computer
Still River Corporate Ctr
55 Federal Road
Danbury 06810
Tel: (800) 409-1483

Hamilton Hallmark
125 Commerce Court, Unit 6
Cheshire 06410
Tel: (203) 271-2844
FAX: (203) 272-1704

Pioneer Standard
2 Trap Falls Road
Shelton 06484
Tel: (203) 929-5600

FLORIDA

Anthem Electronics
5200 NW 3rd Avenue
Suite 206
Ft. Lauderdale 33309
Tel: (305) 484-0990

Anthem Electronics
598 S. Northlake Blvd.
Suite 1024
Altamonte Sprgs 32701
Tel: (813) 797-2900
FAX: (813) 796-4880

Arrow/Schweber Electronics
400 Fairway Drive
Suite 102
Deerfield Beach 33441
Tel: (305) 429-8200
FAX: (305) 428-3991

Arrow/Schweber Electronics
37 Skyline Drive
Suite 3101
Lake Mary 32746
Tel: (407) 333-9300
FAX: (407) 333-9320

Arrow/Schweber Electronics
4010 Boy Scout Dr.
Suite 295
Tampa 33607
Tel: (813) 873-1030
FAX: (813) 873-0077

Avnet Computer
541 S. Orlando Ave.
Suite 203
Maitland 32751
Tel: (800) 426-7999

Hall-Mark Computer
10491 72nd St. North
Largo 34647
Tel: (800) 409-1483

Hall-Mark Computer
13700 58th St. North
Suite 206
Clearwater 34620
Tel: (800) 409-1483

Hamilton Hallmark
3350 N.W. 53rd Street
Suite 105-107
Ft. Lauderdale 33309
Tel: (305) 484-5482
FAX: (305) 484-2995

NORTH AMERICAN DISTRIBUTORS (Cont'd)

Hamilton Hallmark
10491 72nd St. North
Largo 34647
Tel: (813) 541-7440
FAX: (813) 544-4394

Hamilton Hallmark
7079 University Blvd.
Winter Park 32792
Tel: (407) 657-3300
FAX: (407) 678-4414

Pioneer Technologies Group
337 Northlake Blvd
Suite 1000
Alta Monte Spgs 32701
Tel: (407) 834-9090
FAX: (407) 834-0865

Pioneer Technologies Group
674 South Military Trail
Deerfield Beach 33442
Tel: (305) 428-8877
FAX: (305) 481-2950

Wyle Electronics
1000 112th Circle North
St. Petersburg 33716
Suite 800
Tel: (813) 579-1518
FAX: (813) 579-1518

Zeus Arrow Electronics
37 Skyline Drive
Bldg D., Suite 3101
Lake Mary 32746
Suite 800
Tel: (407) 333-3055
FAX: (407) 333-9681

GEORGIA

Anthem Electronics
2400 Pleasant Hill Rd
Suites 9 & 10
Duluth 30136
Tel: (404) 931-3900
FAX: (404) 931-3902

Arrow/Schweber Electronics
4250 E Rivergreen Pkwy
Suite E
Duluth 30136
Tel: (404) 497-1300
FAX: (404) 476-1493

Avnet Computer
3425 Corporate Way
Suite G
Duluth 30136
Tel: (800) 426-7999

Hall-Mark Computer
3425 Corporate Way
Suite G
Duluth 30136
Tel: (800) 409-1483

Hamilton Hallmark
3425 Corporate Way
Suite G & A
Duluth 30136
Tel: (404) 623-5475
FAX: (404) 623-5490

Pioneer Technologies Group
4250C Rivergreen Pkwy
Duluth 30136
Tel: (404) 623-1003
FAX: (404) 623-0665

Wyle Electronics
6025 The Comers Pkwy
Suite 111
Norcross 30092
Tel: (404) 441-9045
FAX: (404) 441-9086

ILLINOIS

Anthem Electronics
1300 Remington Road
Suite A
Schaumberg 60173
Tel: (708) 884-0200
FAX: (708) 885-0480

Arrow/Schweber Electronics
1140 W Thorndale Rd
Itasca 60143
Tel: (708) 250-0500

Avnet Computer
1124 Thorndale Ave
Bensenville 60106
Tel: (800) 426-7999

Hall-Mark Computer
1124 Thorndale Ave
Bensenville 60106
Tel: (800) 409-1483

Hamilton Hallmark
1130 Thorndale Ave
Bensenville 60106
Tel: (800) 426-7999

MTI Systems Sales
1140 West Thorndale Avenue
Itasca 60143
Tel: (708) 250-8222
FAX: (708) 250-8275

Pioneer Standard
2171 Executive Drive
Suite 200
Addison 60101
Tel: (708) 495-9680
FAX: (708) 495-9831

Wyle Electronics
2055 Army Trail Road
Suite 140
Addison 60101
Tel: (800) 853-9953

Zeus Arrow Electronics
1140 W Thorndale Ave
Itasca 60143
Tel: (708) 250-0500

INDIANA

Arrow/Schweber Electronics
7108 Lakeview
Parkway West Drive
Indianapolis 46268
Tel: (317) 299-2071
FAX: (317) 299-2379

Avnet Computer
655 West Carmel Drive
Suite 160
Carmel 46032
Tel: (800) 426-7999

Hall-Mark Computer
655 West Carmel Drive
Carmel 46032
Tel: (800) 409-1483

Hamilton Hallmark
655 West Carmel Drive
Suite 160
Carmel 46032
Tel: (317) 575-3500
FAX: (317) 575-3535

Pioneer Standard
9350 Priority Way W Dr
Indianapolis 46250
Tel: (317) 573-0880
FAX: (317) 573-0979

KANSAS

Arrow/Schweber Electronics
9801 Legler Road
Lenexa 66219
Tel: (913) 541-9542
FAX: (913) 541-0328

Hall-Mark Computer
10809 Lakeview Ave
Lenexa 66219
Tel: (800) 409-1483

Hamilton Hallmark
10809 Lakeview Avenue
Lenexa 66215
Tel: (913) 888-4747
FAX: (913) 888-0523

MARYLAND

Anthem Electronics
7168A Columbia
Gateway Drive
Columbia 21046
Tel: (800) 239-6039

Arrow/Schweber Electronics
9800J Patuxent Woods Drive
Columbia 21046
Tel: (301) 596-7800
FAX: (301) 596-7821

Avnet Computer
7172 Columbia
Gateway Drive
Suite G
Columbia 21045
Tel: (800) 426-7999

Hall-Mark Computer
7172 Columbia
Gateway Drive
Suite G
Columbia 21046
Tel: (800) 409-1483

Hamilton Hallmark
10240 Old Columbia Road
Columbia 21046
Tel: (410) 988-9800
FAX: (410) 381-2036

North Atlantic Industries
Systems Division
7125 River Wood Drive
Columbia 21046
Tel: (301) 312-5800
FAX: (301) 312-5850

Pioneer Technologies Group
15810 Gaither Road
Gaithersburg 20877
Tel: (301) 921-0660
FAX: (301) 670-6746

Wyle Electronics
9101 Guilford Road
Suite 120
Columbia 21046
Tel: (301) 490-2170
FAX: (301) 490-2190

MASSACHUSETTS

Anthem Electronics
200 Research Drive
Wilmington 01887
Tel: (508) 657-5170
FAX: (508) 657-6008

Arrow/Schweber Electronics
25 Upton Drive
Wilmington 01887
Tel: (508) 658-0900
FAX: (508) 694-1754

Avnet Computer
10 D Centennial Drive
Peabody 01960
Tel: (800) 426-7999

Hall-Mark Computer
10 D Centennial Drive
Peabody 01960
Tel: (800) 409-1483

Hamilton Hallmark
10 D Centennial Drive
Peabody 01960
Tel: (508) 531-7430
FAX: (508) 532-9802

Pioneer Standard
44 Hartwell Avenue
Lexington 02173
Tel: (617) 861-9200
FAX: (617) 863-1547

Wyle Electronics
5 Oak Park Drive
Bedford 01803
Tel: (617) 271-9953
FAX: (617) 275-3809

Zeus Arrow Electronics
25 Upton Drive
Wilmington 01887
Tel: (508) 658-4776
FAX: (508) 694-2199

MICHIGAN

Arrow/Schweber Electronics
44720 Helm Street
Plymouth 48170
Tel: (313) 462-2290
FAX: (313) 462-2686

Avnet Computer
41650 Garden Brk Rd
Suite 120
Novi 48375
Tel: (800) 426-7999

Hall-Mark Computer
41650 Garden Brk Rd
Suite 120
Novi 48375
Tel: (800) 409-1483

Hamilton Hallmark
44191 Plymouth Oaks Blvd.
Suite 1300
Plymouth 48170
Tel: (313) 416-5806
FAX: (313) 416-5811

Hamilton Hallmark
41650 Garden Brk Rd
Suite 100
Novi 49418
Tel: (313) 347-4271
FAX: (313) 347-4021

Pioneer Standard
4505 Broadmoor S.E.
Grand Rapids 49512
Tel: (616) 698-1800
FAX: (616) 698-1831

Pioneer Standard
44190 Plymouth Oaks Blvd.
Plymouth 48170
Tel: (313) 525-1800
FAX: (313) 427-3720

MINNESOTA

Anthem Electronics
7646 Golden Triangle Drive
Eden Prairie 55344
Tel: (612) 944-5454
FAX: (612) 944-3045

Arrow/Schweber Electronics
10100 Viking Drive
Suite 100
Eden Prairie 55344
Tel: (612) 941-5280
FAX: (612) 942-7803

Avnet Computer
9800 Bren Roaqd East
Suite 410
Minnetonka 55343
Tel: (800) 426-7999

Hall-Mark Computer
9800 Bren Road East
Suite 410
Minnetonka 55343
Tel: (800) 409-1483

Hamilton Hallmark
9401 James Ave South
Suite 140
Bloomington 55431
Tel: (612) 881-2600
FAX: (612) 881-9461

Pioneer Standard
7625 Golden Triangle Drive
Suite G
Eden Prairie 55344
Tel: (612) 944-3355
FAX: (612) 944-3794

Wyle Electronics
1325 East 79th Street
Suite 1
Bloomington 55425
Tel: (612) 853-2280
FAX: (612) 853-2298

MISSOURI

Arrow/Schweber Electronics
2380 Schuetz Road
St. Louis 63141
Tel: (314) 567-6888
FAX: (314) 567-1164

Avnet Computer
3783 Rider Train South
Earth City 63045
Tel: (800) 426-7999

Hall-Mark Computer
3783 Rider Trail South
Earth City 63045
Tel: (800) 409-1483

Hamilton Hallmark
3783 Rider Trail South
Earth City 63045
Tel: (314) 291-5350
FAX: (314) 291-0362

NEW HAMPSHIRE

Avnet Computer
2 Executive Park Drive
Bedford 03102
Tel: (800) 426-7999

NEW JERSEY

Anthem Electronics
26 Chapin Road, Unit K
Pine Brook 07058
Tel: (201) 227-7960
FAX: (201) 227-9246

Arrow/Schweber Electronics
4 East Stow Road
Unit 11
Marlton 08053
Tel: (609) 596-8000
FAX: (609) 596-9632

Arrow/Schweber Electronics
43 Route 46 East
Pine Brook 07058
Tel: (201) 227-7880
FAX: (201) 227-2064

Avnet Computer
1-B Keystone Avenue
Building 36
Cherry Hill 08003
Tel: (800) 426-7999

Hall-Mark Computer
1-B Keystone Avenue
Building 36
Cherry Hill 08003
Tel: (800) 409-1483

Hall-Mark Computer
10 Lanidex Plaza West
Parsippany 07054
Tel: (800) 409-1483

Hamilton Hallmark
1 Keystone Avenue
Building 36
Cherry Hill 08003
Tel: (609) 424-0110
FAX: (609) 751-2552

Hamilton Hallmark
10 Lanidex Plaza West
Parsippany 07054
Tel: (201) 515-5300
FAX: (201) 515-1601

MTI Systems Sales
43 Route 46 East
Pinebrook 07058
Tel: (201) 882-8780
FAX: (201) 539-6430

NORTH AMERICAN DISTRIBUTORS (Cont'd)

PioneerStandard
14-A Madison Road
Fairfield 07006
Tel: (201) 575-3510
FAX: (201) 575-3454

Wyle Electronics
115 Route 46, Bldg F
Mountain Lakes 07046
Tel: (201) 402-4970

NEW MEXICO

Alliance Electronics, Inc.
3411 Bryn Mawr N.E.
Albuquerque 87101
Tel: (505) 292-3360
FAX: (505) 275-6392

Avnet Computer
7801 Academy Road
Building 1, Suite 204
Albuquerque 87109
Tel: (800) 426-7999

NEW YORK

Anthem Electronics
47 Mall Drive
Commack 11725
Tel: (516) 864-6600
FAX: (516) 493-2244

Arrow/Schweber Electronics
3375 Brighton Henrietta Townline Road
Rochester 14623
Tel: (716) 427-0300
FAX: (716) 427-0735

Arrow/Schweber Electronics
20 Oser Avenue
Hauppauge 11788
Tel: (516) 231-1000
FAX: (516) 231-1072

Avnet Computer
2 Penn Plaza
Suite 1245
New York 10121
Tel: (800) 426-7999

Avnet Computer
1057 E. Henrietta Road
Rochester 14623
Tel: (800) 426-7999

Hall-Mark Computer
2 Penn Plaza
New York 10121
Tel: (800) 409-1483

Hall-Mark Computer
1057 E Henrietta Road
Rochester 14623
Tel: (800) 409-1483

Hamilton Hallmark
933 Motor Parkway
Hauppauge 11788
Tel: (516) 434-7470
FAX: (516) 434-7491

Hamilton Hallmark
1057 E Henrietta Road
Rochester 14623
Tel: (716) 475-9130
FAX: (716) 475-9119

Hamilton Hallmark
3075 Veterans Memorial Hwy.
Ronkonkoma 11779
Tel: (516) 737-0600
FAX: (516) 737-0838

MTI Systems Sales
1 Penn Plaza
250 West 34th Street
New York 10119
Tel: (212) 643-1280
FAX: (212) 643-1288

Pioneer Standard
68 Corporate Drive
Binghamton 13904
Tel: (607) 722-9300
FAX: (607) 722-9562

Pioneer Standard
60 Crossway Pk West
Woodbury, Long Island 11797
Tel: (516) 921-8700
FAX: (516) 921-2143

Pioneer Standard
840 Fairport Park
Fairport 14450
Tel: (716) 381-7070
FAX: (716) 381-5955

Zeus Arrow Electronics
100 Midland Avenue
Port Chester 10573
Tel: (914) 937-7400
FAX: (914) 937-2553

NORTH CAROLINA

Anthem Electronics
4805 Greenwood
Suite 100
Raleigh 27604
Tel: (919) 782-3550

Arrow/Schweber Electronics
5240 Greensdairy Road
Raleigh 27604
Tel: (919) 876-3132
FAX: (919) 878-9517

Avnet Computer
4421 Stuart Andrew Boulevard
Suite 600
Charlotte 28217
Tel: (800) 426-7999

Hall-Mark Computer
3510 Spring Forest Rd
Suite B
Raleigh 27604
Tel: (800) 409-1483

Hamilton Hallmark
3510 Spring Forest Rd
Suite B
Raleigh 27604
Tel: (800) 409-1483

Hamilton Hallmark
5234 Greens Dairy Rd
Raleigh 27604
Tel: (919) 878-0819

Pioneer Technologies Group
2200 Gateway Ctr. Blvd
Suite 215
Morrisville 27560
Tel: (919) 460-1530

OHIO

Arrow/Schweber Electronics
6573 Cochran Road
Suite E
Solon 44139
Tel: (216) 248-3990
FAX: (216) 248-1106

Arrow/Schweber Electronics
8200 Washington Village Drive
Centerville 45458
Tel: (513) 435-5563
FAX: (513) 435-2049

Avnet Computer
7764 Washington Village Drive
Dayton 45459
Tel: (800) 426-7999

Avnet Computer
2 Summit Park Drive
Suite 520
Independence 44131
Tel: (800) 426-7999

Hall-Mark Computer
5821 Harper Road
Solon 44139
Tel: (800) 409-1483

Hall-Mark Computer
777 Dearborn Pk Lane
Suite L
Worthington 43085
Tel: (800) 409-1483

Hamilton Hallmark
5821 Harper Road
Solon 44139
Tel: (216) 498-1100
FAX: (216) 248-4803

Hamilton Hallmark
777 Dearborn Pk Lane
Suite L
Worthington 43085
Tel: (614) 888-3313
FAX: (614) 888-0767

MTI Systems Sales
23404 Commerce Pk Road
Beachwood 44122
Tel: (216) 464-6688
FAX: (216) 464-3564

Pioneer Standard
4433 Interpoint Blvd
Dayton 45424
Tel: (513) 236-9900
FAX: (513) 236-8133

Pioneer Standard
4800 East 131st Street
Cleveland 44105
Tel: (216) 587-3600
FAX: (216) 663-1004

Wyle Electronics
6835 Cochran Rd.
Solon 44139
Tel: (216) 248-9996

OKLAHOMA

Arrow/Schweber Electronics
12101 East 51st Street
Suite 106
Tulsa 74146
Tel: (918) 252-7537
FAX: (918) 254-0917

Hamilton Hallmark
5411 S. 125th E. Ave
Suite 305
Tulsa 74146
Tel: (918) 254-6110
FAX: (918) 254-6207

Pioneer Standard
9717 East 42nd Street
Suite 105
Tulsa 74146
Tel: (918) 665-7840
FAX: (918) 665-1891

OREGON

Almac Arrow Electronics
9500 S.W. Nimbus Ave
Suite E
Beaverton 97008
Tel: (503) 629-8090
FAX: (503) 645-0611

Anthem Electronics
9090 SW Gemini Drive
Beaverton 97005
Tel: (503) 643-1114
FAX: (503) 626-7928

Avnet Computer
9750 SW Nimbus Ave.
Beaverton 97005
Tel: (800) 426-7999

Hall-Mark Computer
9750 SW Nimbus Ave.
Beaverton 97005
Tel: (800) 409-1483

Hamilton Hallmark
9750 SW Nimbus Ave.
Beaverton 97005
Tel: (503) 526-6200
FAX: (503) 641-5939

Pioneer Technologies
8905 Southwest Nimbus Ave.
Suite 160
Beaverton 97005
Tel: (503) 626-7300
FAX: (503) 626-5300

Wyle Electronics
9640 Sunshine Court
Building G
Suite 200
Beaverton 97005
Tel: (503) 643-7900
FAX: (503) 646-5466

PENNSYLVANIA

Anthem Electronics
355 Business Ctr Drive
Horsham 19044
Tel: (215) 443-5150
FAX: (215) 675-9875

Avnet Computer
213 Executive Drive
Suite 320
Mars 16046
Tel: (800) 426-7999

Arrow/Schweber Electronics
2681 Mosside Blvd
Suite 204
Monroeville 15146
Tel: (412) 856-9490

Pioneer Technologies Group
259 Kappa Drive
Pittsburgh 15238
Tel: (412) 782-2300
FAX: (412) 963-8255

Pioneer Technologies Group
500 Enterprise Road
Keith Valley Bus.Ctr
Horsham 19044
Tel: (215) 674-4000

Wyle Electronics
1 Eves Drive
Suite 111
Marlton 08053-3185
Tel: (609) 985-7953
FAX: (609) 985-8757

TEXAS

Anthem Electronics
651 N. Plano Road
Suite 401
Richardson 75081
Tel: (214) 238-7100
FAX: (214) 238-0237

Anthem Electronics
14050 Summit Drive
Suite 119
Tel: (512) 388-0049
FAX: (512) 388-0271

Arrow/Schweber Electronics
Brake Ctr III, Bldg M1
11500 Metric Boulevard
Suite 160
Austin 78758
Tel: (512) 835-4180
FAX: (512) 832-5921

Arrow/Schweber Electronics
3220 Commander Drive
Carrollton 75006
Tel: (214) 380-6464
FAX: (214) 248-7208

Arrow/Schweber Electronics
19416 Park Row
Suite 190
Houston 77084
Tel: (713) 647-6868
FAX: (713) 492-8722

Avnet Computer
4004 Beltline
Suite 200
Dallas 75244
Tel: (800) 426-799

Avnet Computer
1235 North Loop West
Suite 525
Houston 77008
Tel: (800) 426-7999

Hall-Mark Computer
12211 Technology Blvd
Austin 78727
Tel: (800) 409-1483

Hall-Mark Computer
4004 Beltline Road
Suite 200
Dallas 75244
Tel: (800) 409-1483

Hall-Mark Computer
1235 North Loop West
Houston 77008
Tel: (800) 409-1483

Hamilton Hallmark
12211 Technology Boulevard
Austin 78727
Tel: (512) 258-8848
FAX: (512) 258-3777

Hamilton Hallmark
11420 Page Mill Road
Dallas 75243
Tel: (214) 553-4300
FAX: (214) 553-4395

Hamilton Hallmark
8000 Westglen
Houston 77063
Tel: (713) 781-6100
FAX: (713) 953-8420

Pioneer Standard
1826D Kramer Lane
Austin 78758
Tel: (512) 835-4000
FAX: (512) 835-9829

Pioneer Standard
13765 Beta Road
Dallas 75244
Tel: (214) 263-3168
FAX: (214) 490-6419

Pioneer Standard
10530 Rockley Road
Suite 100
Houston 77099
Tel: (713) 495-4700
FAX: (713) 495-5642

Wyle Electronics
1810 Greenville Ave
Richardson 75081
Tel: (214) 235-9953
FAX: (214) 644-5064

Wyle Electronics
9208 Waterford Center Blvd
Suite 150
Austin 78750
Tel: (512) 345-8853
FAX: (512) 345-9330

Wyle Electronics
2901 Wilcrest
Suite 120
Houston 77099
Tel: (713) 879-9953
FAX: (713) 879-9953

Zeus Arrow Electronics
3220 Commander Dr
Carrollton 75006
Tel: (214) 380-4330
FAX: (214) 447-2222

UTAH

Anthem Electronics
1279 West 2200 South
Salt Lake City 84119
Tel: (801) 973-8555
FAX: (801) 973-8909

Arrow/Schweber Electronics
1946 West Parkway Boulevard
Salt Lake City 84119
Tel: (801) 973-6913
FAX: (801) 972-0200

intel®

NORTH AMERICAN DISTRIBUTORS (Cont'd)

Avnet Computer
1100 East 6600 South
Suite 150
Salt Lake City 84121
Tel: (800) 426-7999

Hall-Mark Computer
1100 East 6600 South
Suite 150
Salt Lake City
Tel: (800) 409-1483

Hamilton Hallmark
1100 East 6600 South
Suite 120
Salt Lake City 84121
Tel: (801) 266-2022
FAX: (801) 263-0104

Wyle Electronics
1325 West 2200 South
Suite E
West Valley 84119
Tel: (801) 974-9953
FAX: (801) 972-2524

WASHINGTON

Almac Arrow Electronics
14360 S.E. Eastgate Way
Bellevue 98007
Tel: (206) 643-9992
FAX: (206) 643-9709

Anthem Electronics
19017 120th Ave N.E.
Suite 102
Bothell 98011
Tel: (206) 483-1700
FAX: (206) 486-0571

Avnet Computer
8630 154th Ave, NE
Redmond 98052
Tel: (800) 426-7999

Hamilton Hallmark
8630 154th Avenue
Redmond 98052
Tel: (206) 881-6697
FAX: (206) 867-0159

Pioneer Technologies
2800 156th Ave S.E.
Suite 100
Bellevue 98007
Tel: (206) 644-7500

Wyle Electronics
15385 NE 90th St
Redmond 98052
Tel: (206) 881-1150
FAX: (206) 881-1567

WISCONSIN

Arrow/Schweber Electronics
200 N. Patrick
Suite 100
Brookfield 53045
Tel: (414) 792-0150
FAX: (414) 792-0156

Avnet Computer
2440 South 179th St
New Berlin 53416
Tel: (800) 426-7999

Hall-Mark Computer
2440 South 179th St
New Berlin 53146
Tel: (800) 409-1483

Hamilton Hallmark
2440 South 179th St
New Berlin 53146
Tel: (414) 797-7844
FAX: (414) 797-9259

Pioneer Standard
120 Bishops Way
Suite 163
Brookfield 53005
Tel: (414) 780-3600
FAX: (414) 780-3613

Wyle Electronics
150 North Patrick
Building 7, Suite 150
Brookfield 53045
Tel: (414) 879-0434
FAX: (414) 879-0474

ALASKA

Avnet Computer
1400 W Benson Blvd
Suite 400
Anchorage 99503
Tel: (800) 426-7999

CANADA

ALBERTA

Avnet Computer
1144 29th Avenue NE
Suite 108
Calgary T2E 7P1
Tel: (800) 387-3406

Pioneer/Pioneer
560, 1212-31 Ave. NE
Calgary T2E 7S8
Tel: (403) 291-1988
FAX: (403) 295-8714

BRITISH COLUMBIA

Almac Arrow Electronics
8544 Baxter Place
Burnaby V5A 4T8
Tel: (604) 421-2333
FAX: (604) 421-5030

Hamilton Hallmark
8610 Commerce Court
Burnaby V5A 4N6
Tel: (604) 420-4101
FAX: (604) 420-5376

Pioneer/Pioneer
4455 North 6 Road
Rochmond V6V 1P6
Tel: (604) 273-5575
FAX: (604) 273-2413

MANITOBA

Pioneer/Pioneer
540 Marjorie Street
Winnipeg R3H 0S9

ONTARIO

Arrow/Schweber Electronics
36 Antares Drive
Unit 100
Nepean K2E 7W5
Tel: (613) 226-6903
FAX: (613) 723-2018

Arrow/Schweber Electronics
1093 Meyerside, Unit 2
Mississauga L5T 1M4
Tel: (416) 670-2010
FAX: (416) 670-5863

Avnet Computer
Canada System Engineering Group
151 Superior Blvd.
Mississuaga L5T 2L1
Tel: (800) 387-3406

Avnet Computer
190 Colonade Road
Nepean K2E 7J5
Tel: (800) 387-3406

Canada System Engineering Group
151 Superior Boulevard
Mississauga L5T 2L1
Tel: (800) 387-3406

Hamilton Hallmark
151 Superior Blvd.,
Unit 1-6
Mississauga L5T 2L1
Tel: (416) 564-6060
FAX: (416) 564-6033

Hamilton Hallmark
190 Colonade Road
Nepean K2E 7J5
Tel: (613) 226-1700
FAX: (613) 226-1184

Pioneer/Pioneer
3415 American Drive
Mississauga L4V 1T6
Tel: (416) 507-2600
FAX: (416) 507-2831

Pioneer/Pioneer
155 Colonnade Rd., S.
Suite 17
Nepean K2E 7K3
Tel: (613) 226-8840
FAX: (613) 226-6352

QUEBEC

Arrow/Schweber Electronics
1100 Street Regis Blvd
Dorval H9P 2T5
Tel: (514) 421-7411
FAX: (514) 421-7430

Gates Arrow Electronics
500 Boul.
St-Jean-Baptiste Ave.
Quebec H2E 5R9
Tel: (418) 871-7500
FAX: (418) 871-6816

Avnet Computer
7575 Trans Canada
Suite 601
St. Laurent H4T 1V6
Tel: (800) 265-1135

Hamilton Hallmark
7575 Transcanada Hwy
Suite 600
Street Laurent H4T 2V6
Tel: (514) 335-1000
FAX: (514) 335-2481

Pioneer/Pioneer
520 McCaffrey
Street Laurent H4T 1N1
Tel: (514) 737-9700
FAX: (514) 737-5212

EUROPEAN DISTRIBUTORS/REPRESENTATIVES

AUSTRIA

Elbatex GmbH
Eitnergasse 6
A-1231 Wien
Tel: (43) 1 866 420
FAX: (43) 1 866 42372

Spoerle Elektronic
Heiligentstädter Str. 52
A-1190 Wien
Tel: (43) 1 363 6500
FAX: (43) 1 369 2273

BELGIUM

Ineico
Ave Croix de Guerre 94
1120 Bruxelles
Tel: (32) 2 244 2811
FAX: (32) 2 216 3304

Diode
Keiberg 2
Minervastraat, 14/B2
1930 Zaventem
Tel: (32) 2 725 4660
FAX: (32) 2 725 4511

CHS
Budasteenweg 2
1830 Machelen
Tel: (32) 2 255 0700
FAX: (32) 2 252 5900

CZECH REPUBLIC

Elbatex
Prechodni 11
CS-14000 Praha 4
Tel: (42) 2 692 8087
FAX: (42) 2 471 8203

DENMARK

Avnet Nortec A/S
Transformervej 17
DK-2730 Herlev
Tel: (45) 4284 2000
FAX: (45) 4492 1552

Farnell Elect Svcs AS
Naverland 29
DK-2600 Glostrup
Tel: (45) 5254 6645
FAX: (45) 4245 7624

ESTONIA

Avnet Baltronic AS
Akadeemia tee 21F,
EE0026
Tallinn
Tel: (372) 2 527 349
FAX: (372) 2 527 556

FINLAND

Computer 2000
Pynyntitte 3
P.O. Box 44
SF-02231 Espoo
Tel: (358) 0 887 331
FAX: (358) 0 887 33343

Avnet Nortec OY
Italahdenkatu 22
SF-00210 Helsinki
Tel: (358) 0 670 277
FAX: (358) 0 692 2326

Farnell Elect Svcs O.Y.
PL. 25
Tyopaiakatu 5
SF-00581 Helsinki
Tel: (358) 0 793 100
FAX: (358) 0 701 9892

FRANCE

Arrow Electronique
73-79 Rue des Solets
Silic 585
94663 Rungis Cedex
Tel: (33) 1 4978 4978
FAX: (33) 1 4978 0596

Avnet EMG SA
79, Rue Pierre Sémard
92310 Chatillon
Tel: (33) 1 4965 2500
FAX: (33) 1 4965 2769

Métrologie
Tour d'Asnières
4, Avenue Laurent Cely
92606 Asnières Cedex
Tel: (33) 1 4080 9000
FAX: (33) 1 4791 0561

Tekelec
Cité des Bruyères
5, Rue Carle Vernet-BP2
92310 Sévres
Tel: (33) 1 4623 2425
FAX: (33) 1 4507 2191

Ineico
135 Ave Louis Roche
92621 Gennevilliers
Tel: (33) 1 4185 4949
FAX: (33) 1 4798 0543

CHS
11 Rue de Cambrai
75019 Paris
Tel: (33) 1 4005 2800
FAX: (33) 1 4034 3734

GERMANY

Avnet E2000
Stahlgruberring 12
81829 München
Tel: (49) 89 451 1001
FAX: (49) 89 45110129

Jermyn GmbH
Im Dachsstück 9
65549 Limburg
Tel: (49) 6431 5080
FAX: (49) 6431 5082 89

Metrologie GmbH
Steinerstrasse 15
81369 München
Tel: (49) 89 742 170
FAX: (49) 89 7421 7111

CHS GmbH
Ohepark 2
21222 Rosengarten
Tel: (49) 4108 120
FAX: (49) 4108 1297

Raab Karcher Elektronik GmbH
Loetscher Weg 66
41334 Nettetal
Tel: (49) 2153 7330
FAX: (49) 2153 7335 13

Spoerle Electronic
Max-Planck Strausse 1-3
63303 Dreieich
Tel: (49) 6103 30 4343
FAX: (49) 6103 30 4425

GREECE

Ergodata
Algiroupoleos 2a
176 76 Kalithea
Tel: (30) 1 951 0922
FAX: (30) 1 959 3160

Pouliadis Assoc. Corp.
Aristotelous Street 3
Sygrou Av. 150
17671 Athens
Tel: (30) 1 924 2072
FAX: (30) 1 924 1066

HUNGARY

Elbatex
Gabor Takacs
Vaci u. 202
H-1138 Budapest
Tel: (36) 1 140 9194
FAX: (36) 1 120 9478

IRELAND

Micro Marketing
Taney Hall
Eglinton Terr. Dundrum
Dublin 14
Tel: (353) 1 298 9400
FAX: (353) 1 298 9828

Arrow Electronics
Unit 7 - Newland Bus Pk
Nass Road - Condalkin
Dublin 22
Tel: (353) 1 627 1949
FAX: (353) 1 459 5490

ISRAEL

Eastronics Limited
Rozanis 11-POB 39300
Tel Baruch
Tel-Aviv 61392
Tel: (972) 3 6458 777
FAX: (972) 3 6458 666

ITALY

Avnet EMG
Via Novara, 570
20100 Milano
Tel: (39) 2 38 103 100
FAX: (39) 2 38 002 988

Farnell Elect Svcs SPA
Viale Milanofiori E/5
I-20090 Assago
Tel: (39) 2 824 701
FAX: (30) 2 824 2631

Lasi Eletronica
P.I. 00839000155
Viale Fulvio Testi, N.280
20126 Milano
Tel: (39) 2 661 431
FAX: (39) 2 6610 1385

Telcom
Via Lorenteggio 270/A
20152 Milano
Tel: (39) 2 4830 2640
FAX: (39) 2 4830 2010

Lifeboat
Via Galileo Ferraris 2
20147 Saronno
Tel: (39) 2 9670 1592
FAX: (39) 2 9670 3113

LATVIA

Avnet Baltronic
Maskavas iela 40/42
New Bldg - Room 513
LV 1018 Riga
Tel: (371) 2 211 109
FAX: (371) 2 211 109

NETHERLANDS

Datelcom B.V.
Meidoornkdae 22
3993 AE Houten
Tel: (31) 3403 57222
FAX: (31) 3403 57220

Diode Components
Coltbaan 17
3439 NG Nieuwegein
Tel: (31) 3402 9 1234
FAX: (31) 3402 3 5924

Koning en Hartman
Energieweg 1
2627 AP Delft
Tel: (31) 15 609 906
FAX: (31) 15 619 194

NORWAY

Avnet Nortec A/S
Postbox 123
N-1364 Hvalstad
Tel: (47) 66 846 210
FAX: (47) 66 846 545

Computer System Integrated A/S
Postbox 198
Hvamsvingen 24
N-2013 Skjetten
Tel: (47) 63 845 411
FAX: (47) 63 845 310

Farnell Elect Svcs A/S
Postbox 120 Furuset
Karihaugvei 89
N-1001 Oslo
Tel: (47) 22 321 270
FAX: (47) 66 846 545

POLAND

Elbatex
ul. Hoza 29/31 M6
PL-00-681 Warszawa
Tel: (48) 2 623 060209
FAX: (48) 2 623 0605

PORTUGAL

Arrow/ATD Elect. LDA
Quinta Grande,
Lote 20 - R/C DTO
2700 Amadora
Tel: (351) 1 471 4182
FAX: (351) 1 471 5886

RUSSIA

Marvel
49 Spalernaya Street
193015 St. Petersburg
Tel: (7) 812 274 3210
FAX: (7) 812 274 3708

Merisel
3 Kroutitskiy Val Street
Section 2
109044 Moscow
Tel: (7) 095 276 4718
FAX: (7) 095 276 4714

Novoiet
53 Russkaya Street
630058 Novosibirsk
Tel: (7) 383 235 6672
FAX: (7) 383 233 1788

Stins Coman
126 Pervomaiskaya St
105203 Moscow
Tel: (7) 095 465 4763
FAX: (7) 095 465 9034

SAUDI ARABIA

Hoshanco
Airport Rd., PO Box 382
Riyadh 11411
Tel: (966) 1 477 2323
FAX: (966) 1 479 2588

SLOVAKIA

Elbatex
Topololanska 23
SK-85105 Bratislava
Tel: (42) 7 831 320
FAX: (42) 7 831 320

SLOVENIA

Elbatex
Stegna 19
SLO-61117 Lubljana
Tel: (30) 61 191 126
FAX: (30) 61 192 398

SOUTH AFRICA

EBE
178 Erasmus Street
Meyerspark
Pretoria 0184
Tel: (27) 12 803 7680
FAX: (27) 12 803 8294

SPAIN

Arrow/ATD Electronica
Albasanz 75-3
28037 Madrid
Tel: (34) 1 304 1534
FAX: (34) 1 327 2778

Metrologia
Avda. Industria, 32-2
28100 Alcobendas
Madrid
Tel: (34) 1 661 1142
FAX: (34) 1 661 5755

Diode
c/Orense, 34
28080 Madrid
Tel: (34) 1 555 3686
FAX: (34) 1 556 7159

SWEDEN

Avnet Computer AB
Box 1392
Englundavägen 7
S-171 41 Solna
Tel: (46) 8 629 1400
FAX: (46) 8 627 5165

Avnet Nortec AB
Box 1830
S-171 27 Solna
Tel: (46) 8 705 1800
FAX: (46) 8 836 918

Farnell Elect Svcs AB
Ankdammsgatan 32
Box 1330
S-171 26 Solna
Tel: (46) 8 830 020
FAX: (46) 8 825 770

SWITZERLAND

Elbatex AG
Hardstrasse 7
CH-5430 Wettingen
Tel: (41) 56 275 000
FAX: (41) 56 271 240

Fabrimex AG
Cherstrasse 4
CH-8152 Opfikon-Glattbrugg
Tel: (41) 1 874 6262
FAX: (41) 1 874 6200

IMITEC
Zurichstrasse
CH-8185 Winkel-Ruti
Tel: (41) 1 862 0055
FAX: (41) 1 862 0266

Indústrade AG
Hertistrasse 31
CH-8304 Wallisellen
Tel: (41) 1 832 8111
FAX: (41) 1 830 7550

TURKEY

Empa Electronic
Besyol Mah E-5
Karayolu Yani
Florya Is Merkezi
No. 5 34630 Sefakoy
Istanbul
Tel: (90) 212 599 3050
FAX: (90) 212 598 5353

Info
Buyukdere Cad. 107/3
Bangun Han Gayerttepe
80300 Istanbul
Tel: (902) 1 2 275 0780
FAX: (902) 1 2 272 3427

UNITED KINGDOM

Arrow/MMD
3 Bennet Court
Bennet Road
Reading RG2 0QX
Tel: (44) 734 313 232
FAX: (44) 734 313 255

Avnet Access
Jubilee House
Jubilee Road
Letchworth SG6 1QH
Tel: (44) 462 488 500
FAX: (44) 462 488 567

Bytech Systems
5 The Sterling Centre
Eastern Road
Bracknell RG12 2PW
Tel: (44) 344 55 333
FAX: (44) 344 867 270

Bytech Electronics
12a Cedarwood
Chineham Park -
Crockford Lane
Basingstoke RG12 1RW
Tel: (44) 256 707 107
FAX: (44) 256 707 162

EUROPEAN DISTRIBUTORS/REPRESENTATIVES (Cont'd)

Datrontech PLC
42-44 Birchett Road
Aldershot
Hants GU11 1LU
Tel: (44) 252 303 333
FAX: (44) 252 303 444

Metrologie UK Ltd.
Metrologie House
Oxford Road
High Wycombe HP11 2EE
Tel: (44) 494 526 271
FAX: (44) 494 521 860

CHS
Copse Road
St Johns, Woking
Surrey GU21 1SX
Tel: (44) 483 723 411
FAX: (44) 483 729 974

UKRAINE
Kvasar Micro
Popudrenko Str. 52
253094 Kiev
Tel: (7) 044 516 8496
FAX: (7) 044 516 8608

UNITED ARAB EMIRATES

Graytech
P.O. Box 50718
Dubai
Tel: (971) 434 6952
FAX: (971) 434 6546

Jumbo Electronics Co. Ltd.
Al Wahaibi Building
Al Garhoud
P.O. Box 3426
Dubai
Tel: (971) 4 882 4888
FAX: (971) 4 821 839

EUROPEAN SALES OFFICES

BELGIUM

Intel Corporation S.A.
Avenue Winston Churchill 11
1180 Brussels
Tel: 32-2-347-0666
FAX: 32-2-347-0585

BELORUSSIA

Intel Microelectronics Belarus Ltd.
Prospect Scoriny, 11
Hotel Minsk Admin Bldg
App. 246, "RANS"
200042, Minsk
Tel: 375-172-20057
FAX: 375-172-2000257

FINLAND

Intel Finland OY
Albertinkatu 36A, 5th Floor
00181 Helsinki
Tel: (358) 0 400 2179
FAX: (358) 0 400 3787

FRANCE

Intel Corp. S.A.R.L.
1 Quai de Grenelle
BP 543
75725 Paris Decex 15
Tel: (33) (1) 45717171
FAX: (33) (1) 45717000

GERMANY

Intel GmbH
Dornacher Strasse 1
D-85622 Feldkirchen Muenchen
Tel: (49) 89/99143-0
FAX: (49) 89/9043948

Intel GmbH
Wandersmannstrasse 64
65205 Wiesbaden
Tel: (49) 611-977030
FAX: (49) 611-731543

ISRAEL

Intel Semiconductor
Atidim Industrial Park
P.O. Box 43202
Tel-Aviv 61430
Tel: (972) 3 548-3222
FAX: (972) 3 491870

ITALY

Intel Corp. Italia S.P.A.
Milanofiori Palazzo E/4
20094 Assago
Milano
Tel: (39) (2) 575441
FAX: (39) (2) 57501221

POLAND

Intel Poland Developments Inc.
UL Hankiewicza 2
02-103 Warszawa
Tel: 48 2 658 42 46
FAX: 48 2 658 45 47

RUSSIA

Intel Technologies
Krementshugskaya 6/7
121357 Moscow
Tel: 7-503-956-4900
FAX: 7-503-956-4905

Intel Technologies
Klenovaya 2
191011 St. Petersburg
121357 Moscow
Tel: 7-812-119-6072
FAX: 7-812-119-6213

Intel Technologies
PR Lavrentyeva 6, App. 407
630090, Novosibirsk, Siberia
Tel: 7-3832-39-6231
FAX: 7-3832-39-6231

SPAIN

Intel Corp. Iberia
C/Zubaran, 28-1 IZQDA
28010 Madrid
Tel: 341 308 2552
FAX: 341) 310 5460

SWEDEN

Intel Sweden A.B.
Box 18384
Centralvagen 16
171 28 Solna
Tel: (46) 8 705 5600
FAX: (46) 8 278085

UKRAINE

Intel Ukraine Microelectronics Ltd.
Krasnoarmeiskaya 9/2
App. 5 Entrance 1
252004 Keiv
Tel: 380-44-244-3644
FAX: 380-44-244-3642

UNITED KINGDOM

Intel Corp. (U.K.) Ltd.
Pipers Way
Swindon, Wiltshire SN3 1RJ
Tel: (44) (01793) 696000
FAX: (44) (01793) 641440

SWEDEN

Intel Sweden A.B.
Dalvagen 24
171 36 Solna
Tel: (46) 8 705 5600
FAX: (46) 8 278085

UNITED KINGDOM

Intel Corp. (U.K.) Ltd.
Pipers Way
Swindon, Wiltshire SN3 1RJ
Tel: 44 (01793) 696000
FAX: 44 (01793) 641440

INTERNATIONAL SALES OFFICES

AUSTRALIA

Intel Australia Pty.
Level 2, 8 West St
North Sydney, NSW, 2060
Tel: 61 2 9937-5800

BRAZIL

Intel Semicondutores
Rua Florida 1703-2 CJ22
CEP 04565-001
Sao Paulo, SP
Tel: 55-11 505-2296
FAX: 55 11 505-1737

HONG KONG

Intel Semicon. Ltd.
32/F Two Pacific Place
88 Queensway Central
Hong Kong
Tel: 852 2844 4555
FAX: 852 2868 1989

INDIA

Intel Asia Electron., Inc.
4/2, Samrah Plaza
St. Mark's Road
Bangalore 560001
Tel: 91 80 221 5065
FAX: 91 80 227 5067

JAPAN

Intel Japan K.K.
Hachioji Office
Hachioji on Bldg. 4f
4-17-14 Myojincho
Hachioji-sji, Tokyo
Tel: 81 426 48 8770
FAX: 81 426 48 8775

Intel Japan K.K.
Osaka Sales Office
Byokuchi-Eki Bldg.
2-4-1 Terauchi
Toyonaka-shi, Osaka, 560
Tel: 81 6 863 1091
FAX: 81 6 863 1084

Intel Japan K.K.
Tokyo HQ
Hotel New Otani
Garden Court 10f
4-1 Kiolcho Chiyadaku
Tokyo, 102
Tel: 81 3 5210 0670
FAX: 81 3 5210 0678

Intel Japan K.K.
Shinyokohama Office
Kawaasa Bldg.
11-5, Shin-Yokohama 2-Chome
Kohoku-Ky, Yokohama
Kanagawa-ken, 222
Tel: 81 45 474 7679
FAX: 81 45 474 7660

Intel Japan K.K.
Meiji Seimei Kurosaki Bldg. 8F
3-9-18 Kurosaki,
Yahata Nishi-ku
Kita Kyushu-shi
Fukuoka-ken, 806
Tel: 81 93 622 8700
FAX: 81 93 622 8720

Intel Japan K.K.
c/o NPNX
1580 Yamamoto
Tateyama-shi
Chiba-ken, 294
Tel: 81 849 45 8121
FAX: 81 849 45 8124

KOREA

Intel Korea, Ltd.
4th Floor, Dae Han Inbestements & Trust Co. Bldg
27-3, Yoido-dong
Youngdeungpo-ku
Seoul 150-010
Tel: 822 767 2500
FAX: 822 767 2509

MEXICO

Intel Tecnologia de Mexico,S A. de C.V.
Av. Mexico No. 2798-9B, S.H.
Guadalajara, Jal. 44680
Tel: 523-640-1259
FAX: 523-642-7661

Intel Tecnologia de Mexico,S A. de C.V.
Prolog. Paseo de la Reforma
No. 600 PH 312
Edificio Plaza Reforma
Col. Pena Blanca Santa Fe
01210 Mexico, D.F..
Tel. 525-259-7531

SINGAPORE

Intel Singapore Technology, Ltd.
391 Orchard Road
#14-02 Tower A
Ngee Ann City
Singapore 0923
Tel: 65 735 3811
FAX: 65 252 1097

SOUTH AFRICA

Intel South Africa
P.O. Box 1162
Auckland Park 2006
Tel: 27-11-482-8154
FAX: 27-11-482-8153

TAIWAN

Intel Technology (Far East) Ltd.
8th Flr,Bank Twr Bldg
No. 205 Tun Hwa N. Rd
Taipei
Tel: 886 2 716 9660
FAX: 886 2 717 2455

INTERNATIONAL DISTRIBUTORS/REPRESENTATIVES

ARGENTINA

DAFSYS Consulting SA
Chacabuco, 90-6 Piso
1069 Buenos Aires
Tel: 54 1 342 7726
FAX: 54 1 334 1871

Reycom Electronics SA
Bernardo de Irigoyen
972-2B
1304 Buenos Aires
Tel: 54 1 300 2013
FAX: 54 1 304 2018

AUSTRALIA

Hardie Technologies
205 Middleborough Road
Box Hill, Victoria 3128

BRAZIL

Hitech
Av. Eng. Luiz Carlos
Berrini
801-12 andar
Sao Paulo/SP
Cep: 04571 901
Tel: 55 11 505 5277
FAX: 55 11 505 2914

Itaucom
Av. Dr. Hugo Beolchi
900-Terraco-Predio D
Sao Paulo - SP Brazil
Cep: 13213 000
Tel: 55 11 584 3704
FAXL 55 11 584 3719

CHILE

DTS
Rosas 1444
Santiago
Tel: 562 697 0991
FAX: 562 699 3316

COLOMBIA

Dimser Ingenieria Ltda
Avenida El Dorado No.
84A-55 Loc. 130
Dorado Plaza
Santa Fe de Bogota D.C.
Tel: 571 295 3642
FAX: 571 295 5998

ETEK Electronics Corp
6353 W. Rogers Circle
Suite 3
Boca Raton 33487
Tel: 407 997 6277
FAX: 407 997 5467

HONG KONG

Novel Precision Machinery Co., Ltd.
Room 728 Trade Square
681 Cheung Sha Wan Rd
Kowloon, Hong Kong
Tel: 852 360 8999
TLX: 32032 NVTNL HX
FAX: 852 725 3695
Cable: NVTNCINDHK

H.K.Components Agent Ltd.
36/F Metroplaza, Tower 1
Hing Fong Road
Kwai Chung
New Territories
Tel: 852 487 8826
TLX: 30398 COMAG HX
FAX: 852 487 1268

Novel Precision Machinery Co., Ltd.
Rm 728Trade Square
681 Cheung Sha Wan
Road
Kowloon
Tel: 852 360 8999
TLX: 32032 NVTNL HX
FAX: 852 725 3695
Cable: NVTNCINDHK
H.K.

INDIA

SES Computers and Technologies Pvt., Ltd.
11/18 SNS Chambers
239 Palace Upper
Orchards
Sankey Roadd,
Sadashivnagar
Bangalore 560 080
Tel: 91 80 334 8481
FAX: 91 80 334 3685

SAI Svcs Int'l Pvt., Ltd.
19-01A, Topng Eng Bldg
101 Cecil Street
Singapore 0106
Tel: 65 227 1962
FAX: 65 227 1963

PEAC Electronics and Chemicals Pvt., Ltd.
120 Hill View Ave.
03-05 Kewalram Hill View
Singapore 2366
Tel: 65 763 6889
FAX: 65 763 6819

Priya International, Ltd.
D-6 2nd Floor
Devatha Plaza
131/132 Residency Road
Bangalore 560 025
Tel: 91 80 221 4027
FAX: 91 80 221 4150

JAMAICA

M.C. Systems
10-12 Grenada Crescent
Kingston 5
Tel: 809-926-0104
FAX: 809-929-5678

JAPAN

Asahi Electronics Co. Ltd.
KMM Building 2-14-1
Asano
Kokurakita-ku
Kitakyushu-shi 802
Tel: 81 93 511 6471
FAX: 81 93 551 7861

C. Itoh Techno-Science Co., Ltd.
C. Itoh Bldg.
2-5-1 Kita-Aoyama
Minato-ku, Tokyo 107
Tel: 81 3 497 4900
FAX: 81 3 497 4879

Dia Semicon Systems, Inc.
Wacore 64
1-37-8 Sangenjaya
Setagaya-ku, Tokyo 154
Tel: 81 3 487 0386
FAX: 81 3 487 8088

Okaya Koki
2-4-18 Sakae
Naka-ku, Nagoya-shi 460
Tel: 81 52 204 2916
FAX: 81 52 204 2901

Ryoyo Electro Corp.
Konwa Bldg.
1-12-22 Tsukiji
Chuo-ku, Tokyo 104
Tel: 81 3 3546 5011
FAX: 81 3 3546 5044

KOREA

Samsung Electronics
15th Flr, Severance Bldg.
84-11, 5-KA,
Namdaemoon Ro
Chung-Ku
Seoul 100-095
Tel: 82 2 259 4755
FAX: 82 2 259 2482

Tong Baek Electronics Co., Ltd.
Yong San Electronic
Office
3F/RM 303
16-58, 3-KA, Han Gang
Ru
Yong San-Ku, Seoul

MEXICO

Dicopel, SouthA. de C.V.
Fco. Pimentel No. 98
Col. San Rafael
Mexico D.F., C.P. 06470
Tel: 525 705 7422
Fax: 525 703 1772

Dicopel Inc.
2256 Main Street
Suite 18
Chula Vista, CA 91911
U.S.A.
Te:: (619) 423-3392
FAX: (619) 423-3394

NEW ZEALAND

Email Electronics
36 Olive Road
Penrose, Auckland
Attn: Dean Danford
Tel: 64 9 591 155
FAX: 64 9 592 681

PAKISTAN

Global Systems
R.B.S II Floor
Awami Complex
Garden Town
Tel: 42 865 173
FAX: 42 871 528

SAUDI ARABIA

AAE Systems, Inc.
642 N. Pastoria Avenue
Sunnyvale, CA 94086
USA
Tel: 408 732 1710
FAX: 408 732 3095
TLX: 494 3405 AAE SYS

SINGAPORE

PEAC Electronics and Chemicals, PTE, Ltd.
120 Hillview Ave.
03-05 Kewalram Hill View
Singapore 2366

SAI Services International PTE, Ltd.
19-01A, Tong Eng
Building
101 Cecil Street
Singapore 0106

Electronic Resources, PTE, Ltd.
17 Harvey Road #04-01
Singapore 1336
Tel: 65 283 0888
TWX: 56541 FRELS
FAX: 65 289 5327

SOUTH AFRICA

Electronic Building Elements
178 Erasmus Street
Meyerspark, Pretoria
Tel: 2712 803 7680
FAX: 2712 803 8294

TAIWAN

Acer Sertek Inc.
11-15/F, 135 Section 2
Chien Kuo North Road
Taipei 10479

Synntex Technology Inter Corp.
11/F, 75 Sec. 3
Ming-Sheng East Road
Taipei

URUGUAY

Interfase SouthA.
Bulevar Espana 2094
11200 Montevideo
Tel: 598 249 4600
FAX: 598 249 3040

YUGOSLAVIA

H.R. Microelectronics Corp.
2005 de la Cruz Blvd
Suite 190
Santa Clara, CA 95050
U.S.A.
Tel: (408) 988-0286
TLX: 387452
FAX: (408) 988-0306

MOUNTAIN VIEW, CA (covers rest of Latin America)

Intectra
2629 Terminal Boulevard
Mt. View, CA 94043
U.S.A.
Tel: (415) 967-8818
FAX: (415) 967-8838

NORTH AMERICAN SERVICE OFFICES

Computervision

Intel Corporation's North American Preferred Service Provider

Central Dispatch: 1-800-876-SERV (1-800-876-7378)

ALABAMA
- Birmingham
- Huntsville

ALASKA
- Anchorage

ARIZONA
- Phoenix
- Tucson

ARKANSAS
- North Little Rock

CALIFORNIA
- Concord
- Los Angeles
- Ontario
- Orange
- Redwood City
- Sacramento
- San Diego
- San Francisco
- Van Nuys

COLORADO
- Colorado Springs
- Denver

CONNECTICUT
- E. Windsor
- Middlebury

FLORIDA
- Ft. Lauderdale
- Jacksonville
- Miami
- Orlando
- Pensacola
- Tampa

GEORGIA
- Atlanta

HAWAII
- Honolulu

ILLINOIS
- Chicago
- Wood Dale

INDIANA
- Carmel
- Evansville
- Ft. Wayne
- South Bend

IOWA
- Cedar Rapids
- Davenport
- West Des Moines

KANSAS
- Kansas City
- Wichita

KENTUCKY
- Louisville
- Madisonville

LOUISIANA
- Baton Rouge
- New Orleans

MAINE
- Auburn

MARYLAND
- Baltimore

MASSACHUSETTS
- Bedfoird
- S. Easton

MICHIGAN
- Detroit
- Flint
- Grand Rapids
- Lansing
- Troy

MINNESOTA
- Minneapolis

MISSOURI
- Springfield
- Street Louis

MISSISSIPPI
- Jackson

NEW HAMPSHIRE
- Manchester*

MONTANA
- Butte

NEBRASKA
- Omaha

NEW JERSEY
- Cherry Hill
- Hamilton Township
- Westfield

NEW MEXICO
- Albuquerque

NEW YORK
- Albany
- Binghampton
- Buffalo
- Farmingdale
- New York City
- Rochester
- Dryden

NORTH CAROLINA
- Ashville
- Charlotte
- Greensboro
- Raleigh
- Wilmington

OHIO
- Cincinnati
- Cleveland
- Columbus
- Dayton

OKLAHOMA
- Oklahoma City
- Tulsa

OREGON
- Beaverton

PENNSYLVANIA
- Camp Hill
- Erie
- Pittsburgh
- Wayne

SOUTH CAROLINA
- Charleston
- Columbia
- Greenville

TENNESSEE
- Chattanooga
- Knoxville
- Memphis
- Nashville

TEXAS
- Austin
- Houston
- Dallas
- Tyler

UTAH
- Salt Lake City

VERMONT
- White River Junction

VIRGINIA
- Charlottesville
- Richmond
- Roanoke
- Virginia Beach

WASHINGTON
- Renton
- Richland

WASHINGTON D.C.*

WEST VIRGINIA
- Charleston

WISCONSIN
- Milwaukee

CANADA
- Calgary
- Edmonton
- Fredericton
- Halifax
- Mississauga
- Montreal
- Ottawa
- Toronto
- Vancouver, BC*
- Winnipeg
- Quebec City
- Regina
- St.John's

Peripheral Components

Peripheral components range from chip sets for PCI Local Bus and Extended Industry Standard Architecture (EISA) to controllers for Cache, Direct Memory Access (DMA), Floppy Disk and Keyboard.

This databook contains datasheets which include comprehensive charts covering symbols and functions, block diagrams and operational and functional descriptions.

ISBN 1-55512-255-8

Order Number 296467-007
Printed in USA/0196/19k/RRD/TS
Peripheral Components

intel®

Printed on Recycled Material