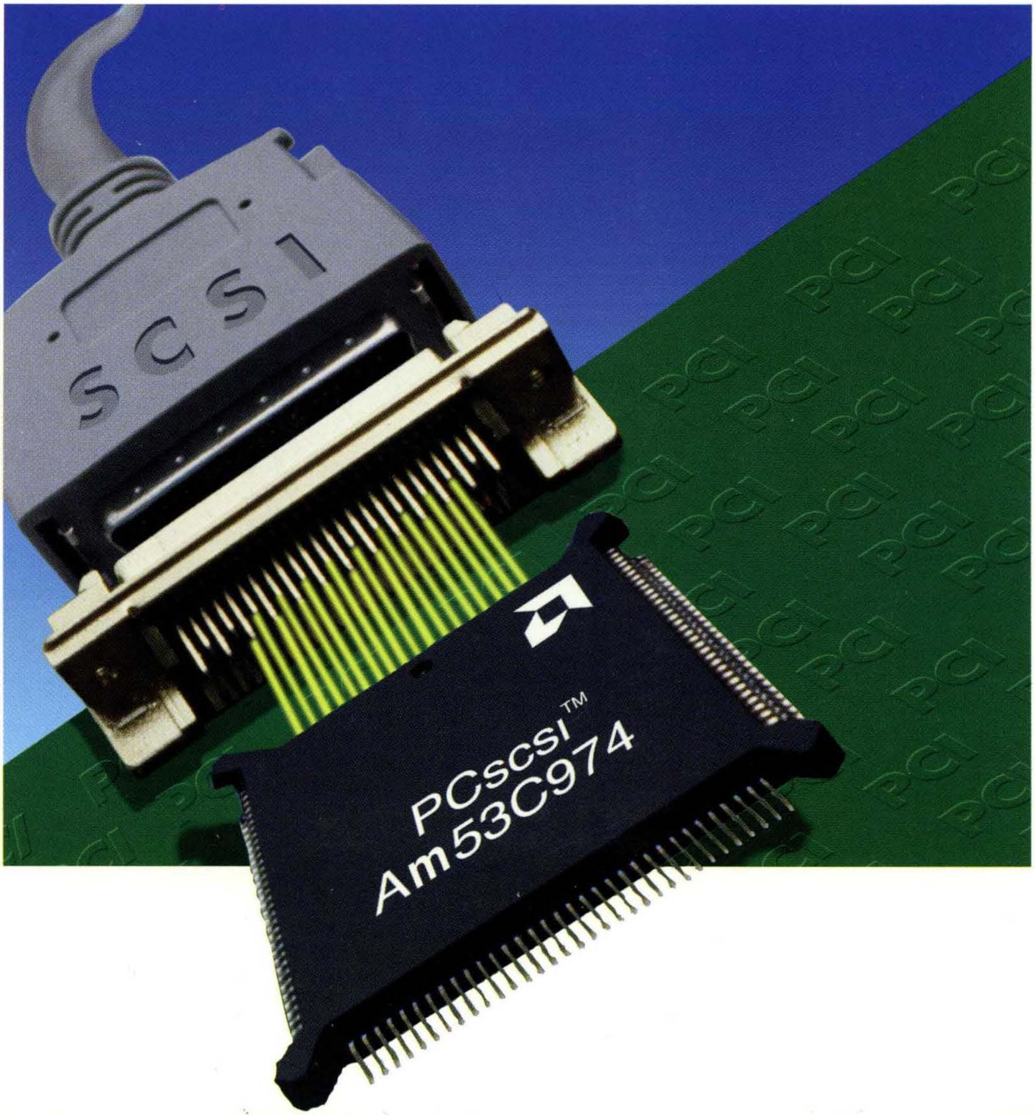
Am53C974 PCscsi™

Bus Mastering Fast SCSI Controller for PCI Systems
Technical Manual

Advanced
Micro
Devices



PCSCSI™
Bus Mastering Fast SCSI Controller for PCI Systems

Technical Manual

Revision 1.1

ADVANCED MICRO DEVICES

Advanced Micro Devices reserves the right to make changes in its products without notice in order to improve design or performance characteristics.

This publication neither states nor implies any warranty of any kind, including but not limited to implied warrants of merchantability or fitness for a particular application. AMD® assumes no responsibility for the use of any circuitry other than the circuitry in an AMD product.

The information in this publication is believed to be accurate in all respects at the time of publication, but is subject to change without notice. AMD assumes no responsibility for any errors or omissions, and disclaims responsibility for any consequences resulting from the use of the information included herein. Additionally, AMD assumes no responsibility for the functioning of undescribed features or parameters.

Trademarks

TABLE OF CONTENTS

PART I — INTRODUCTION

PART II — ARCHITECTURE

PART III — APPLICATIONS AND SOLUTIONS

PART IV — APPENDICES

SCSI TECHNOLOGY OVERVIEW

INTRODUCTION

This chapter is included in the technical specification for the PCI family of AMD SCSI solutions because of the nature of these devices. Each SCSI chip is effectively a complete solution for integrating SCSI onto the motherboard of a PCI based system. All of the logic required for a SCSI port is included in the PCI SCSI device. Because AMD also furnishes the software solution for PC operating systems, users are (for the first time) looking at SCSI as a standard I/O solution, but may have little experience in the basics of the standard. This introduction will provide a basic understanding of the relevant information so that users can quickly integrate and use the SCSI interface.

Before the SCSI standard was defined, in the 1979 time frame, the typical introduction of new peripheral technology required 18 to 24 months. The tasks consisted of:

1. Designing a disk controller board
2. Designing a host adapter board
3. Changing system software to accommodate new device characteristics
4. Testing the new configuration

Disk drive manufacturers were faced with a delay that severely impacted business. If disk development required 18 months, and integration took another 18 months, the total time to revenue was three years. Clearly, this technology bottleneck caused problems for the disk industry. Consequently, vendors began to define logical interfaces that could survive beyond each model, and allow integrators to preserve the majority of their development investment. SCSI was one of these definitions that survived. Originally defined by Shugart Associates, SASI (Shugart Associates Systems Interface) was offered as a public document by Shugart in hopes that other system vendors would use it and establish it as a defacto standard. Within a short time, an ANSI standards group was formed, the name was changed to Small Computer Systems Interface, and the standardization efforts began.

With support from peripheral vendors, the document defined a logical interface to a wide variety of peripherals. This interface would allow system vendors to attach peripherals easily and quickly to new systems, in order to meet a fast moving market window. The standard defined a physical level (mechanical and electrical) as well as a logical level (commands and messages). The actual command passing protocol and the set of peripheral command sets were defined by the standard. Using logical addressing mechanisms, rather than physical addressing (sector 347 vs. cylinder 13, head 2, sector 3), the standard allowed the physical details of the peripheral to be hidden from the system software. Therefore, after the standard was defined, a typical integration cycle was reduced to three months, and consisted of:

1. Replacing the current SCSI disk with a new model
2. Testing the new configuration for compliance

The process was complicated because new peripherals offered features that were enabled through the system software, and because vendors did not offer devices that were compatible with other vendors. But in general, the process was much simpler and faster than before.

Since 1979, the SCSI standard passed through several key stages as it matured into today's standard. These stages are:

SASI (1979 – 1982) — This was a DTC/Shugart collaboration that was a controller board shipped with Shugart drives. Although limited in function, it highlighted the advantages of a high level logical interface for disk drives.

SCSI-1 (1982 – 1986) — A group of companies (Shugart, Adaptec, NCR, & OMTI) banded together and approached ANSI for permission to develop a SCSI standard. The effort generated industry wide interest and was quickly written because of the participants' common interest (i.e. everyone desired quick time to market for their products). The ANSI specification (X3.131–1986) won final approval in May of 1986.

CCS (1985 – 1986) — Upon finalization of the SCSI-1 specification, the participants barely had time to congratulate each other before the weakness of the document began to emerge. The number of options allowed vendors were to develop disks that worked fine, but which were not compatible with each other. Thus system vendors could not easily integrate disk drives from all vendors into their system. So, disk vendors met to define an extended subset of SCSI that would (if followed closely) permit vendors to produce compatible disk products. This document (Common Command Set) became a defacto standard and allowed further standardization of the SCSI market.

SCSI-2 (1986 – 199X) — As soon as the CCS specification was written, the complete SCSI community realized the benefit of these extensions and restarted the SCSI effort to bring the benefits of CCS into the complete SCSI standard document. The original goal was to quickly fold CCS into the disk section and expand other command sets to include CCS features. Unfortunately, the door of improvement, once open, allowed a flood of improvements to come in. Updates were cut off by 1990, but the final specification is still not out as of 6/93. The specification is finalized however, and is to scheduled for publication before the end of 1993.

SCSI-3 (1990 – 199X) — The stated goal of the first two SCSI specifications was downward compatibility, but with SCSI-3, backward compatibility was sacrificed. The goal was to define a protocol that accommodated the new serial interfaces and solved some of the problems of the parallel interface. The result is a family of documents being written for SCSI-3.

Basic SCSI

The standard SCSI parallel bus allows connection for 8 or 16 devices (computers or peripherals). Each has a unique identifier that allows selection and communication between any two devices on the SCSI bus. An Initiator is an entity on the bus that issues a command to a Target. Note that any device on the bus, peripheral or computer system, can initiate a command sequence. The Initiator arbitrates for and wins the SCSI bus, selects the desired Target, and sends a command to that Target. At this point, the Target controls the bus and directs the remainder of the command sequence. The Initiator can always regain control by use of the attention line that signals the Target that something is coming. But in general, the Target remains in control.

There are nine control signals on the SCSI bus that allow the complete handshake to happen. Target or Initiator control the Bus depending on the phase of the command sequence. There are 8, 16, or 32 data lines with each 8-bit set having a parity line.

The data width can be negotiated for, with 8 being the default, and 16 or 32 being options. 16 bit SCSI devices are available but are not in widespread use, while 32-bit devices are not yet available. There is considerable inertia resisting going beyond 8 bit cables because of the physical sizes involved. There is a Fast SCSI option that allows SCSI to transfer data at 10 mega transfers per second, thus allowing 10 MB/sec on 8-bit cables. This configuration is the clear winner in today's market, because it allows faster transfers with no change in the physical environment. Single ended SCSI is by far the most popular implementation and the specification allows 6 meter cables in this environment. Fast SCSI is not defined for single-ended cables, however, vendors are providing silicon that will allow data to be transferred at the higher speeds.

The SCSI protocol defines the phases required to complete a SCSI I/O. The Message phase allows short transfers of protocol related information to be sent across the bus to establish and control the logical connection between an Initiator and a Target. The Command phase allows the Initiator to tell the Target what action must be performed. Read, write, rewind, etc. are all SCSI commands. User data is transferred during the Data phase, allowing the data to be sent into or out from the Initiator. This data can be written to the media, or it can be control information that is used to communicate operating modes or sensed information to(from) the peripheral. Status information (one byte only) is sent to the Initiator from the Target at the end of the complete SCSI command sequence. If an error occurred, the Initiator must request the details of the problem with a subsequent SCSI command.

The basic SCSI command sequence follows:

- Bus is free and any device is allowed to arbitrate for ownership
- An Initiator arbitrates for the bus and selects a Target device
- The Target, after sensing selection, goes to Message Out phase and receives a one byte ID message from the Initiator that establishes which physical device the Initiator wants to communicate with
- After switching to Command phase, the Target receives the SCSI command.
- Changing to the appropriate Data phase (in/out) the Target handshakes the data across the bus
- A single status byte is sent to the Initiator
- A command complete message is sent to the Initiator to signal completion of the SCSI I/O
- The Target then physically disconnects from the SCSI bus, and it is free for the next device to use

Any time after the ID message is sent, the Target controls the sequence and directs the Initiator's next step. The Target is allowed to disconnect, temporarily get off the bus and reselect later. In this manner, data can be multiplexed from various peripherals to the host computer, in support of overlapped physical actions at the peripheral and multi-threaded I/O in the operating system.

SCSI-1 provided the basic protocol functionality that supported the original goals of device connectability, allowing 8 bit buses and a limited protocol flexibility. As noted above, the plug and play aspects of the specification were not quickly realized because of the large set of options. The CCS specification defined the extended subset of the SCSI-1 document that allowed multiple disk vendors to work in one system environment without compatibility issues in the host software. The main SCSI-2 features were:

- Compatibility — There must be an evolutionary growth, and mixed environments were permitted. Requirements in SCSI-2 were popular options in SCSI-1.
- High Performance — The original 5 MB/sec limit was pushed to 40 MB/sec using a combination of 32-bit buses and 10 Mega transfers per second. The maximum cable limits were still set at 6 (single ended) and 25 (differential) meters. Eight devices were allowed on the bus, but a working paper described how sixteen devices could be attached.
- Hardware Requirements — Parity on the bus is required for data integrity, and disconnect/reconnect is required for multiplexing the SCSI bus. SCSI-1 had no requirement for messages (all were optional) because of compatibility with SASI, but SCSI-2 required certain messages to make SCSI more usable in a complex systems environment.
- Logical Interface Improvements — A group of features were defined to generally improve SCSI. Features included were queueing of 256 commands at the peripheral, more detailed definition for all control data used to change device characteristics (or report information about the device), cache support for direct access devices, and tape partition support.
- New Command Sets — New device command sets were for scanners, optical memory devices, medium changers, and communication devices.

These changes dramatically increased the size of the SCSI-2 specification, and moved the definition into a more high performance system environment.

SCSI-3 has given SCSI a totally new set of capabilities. The basic desire was to increase the functionality to cover all the physical interfaces and allow the architecture to be extended to the future requirements with few restrictions for compatibility. The standard was divided into multiple documents to ease the pain of an editor. There are four transport layers:

- Standard parallel interface for 8- and 16-bit cables
- Fibre channel for 100 MB/sec, full duplex transfers
- Serial Storage Architecture in support of high performance (20 MB/sec) disks
- P1394 allows 400 Mb/sec isochronous transfers

Five protocols:

- Interlocked
- Fibre channel
- SSA
- Serial bus
- Generic packetized

Five command sets:

- Primary
- Block
- Stream
- Graphics
- Medium Changer

The target release date for these set of documents is 1994, but if the past is any indication, SCSI-3 will require many years of work before being ready for official release. The delay will not stop products from emerging however, because products with the new interfaces and features are already appearing in the marketplace.

Before SCSI could penetrate the PC market, a critical piece of technology was required: I/O software drivers to support the SCSI capabilities in the PC operating system environments. As the first vendors introduced host adapter boards that connected SCSI devices to the PC, the user quickly learned the importance of software. Consider the following scenario:

- User buys a SCSI disk with host adapter
- A tape drive is required to back up the disk
- User buys a tape drive
- The tape can be connected, but there is no software to drive the device
- User buys a host adapter to drive the tape
- The tape works now, but there is no software to move data to/from the disk

Point solutions became available, but general solutions were not available until 1991. Although the hardware was available and certain systems vendors were able to make SCSI work, there was no guarantee that a user could find an off-the-shelf solution for the DOS/Windows environment. Host adapter companies quickly began to offer limited software capabilities that allowed several devices to be used in the PC.

Several defacto interfaces were available and in the interest of standardization, a Common Access Method (CAM) committee was formed to standardize access to the SCSI interface for each PC operating system environment. Unfortunately, once software was running, users were reluctant to change. Therefore, the CAM specification, although technically sound, has never been widely accepted as a standard. The Advanced SCSI Programming Interface (ASPI) instead became a defacto standard interface for I/O application software such as CD-ROM and tape backup utilities. Consequently, ASPI seems to be entrenched as the interface of choice for DOS/ Windows.

However, more sophisticated operating systems already have a logical I/O interface defined and do not share the same limitations as DOS. As the new operating systems become more widely used in PCs and DOS I/O gradually becomes buried in an emulation mode, the critical driver issues for the PC will disappear. Until then (as Yogi Berra says, "Tomorrow isn't here yet"), highly integrated SCSI chips must be supplied with SCSI I/O drivers for the operating systems that execute on a PC. The drivers must be architected carefully so as to require minimal change when the SCSI device changes. Known as a two layer software interface, a good architecture will isolate changes in the operating system from changes in the SCSI interface. Software technology like this allows SCSI to be commonly available in the PC environment.

Another key element in the I/O performance saga is the availability of bus mastering in the PC environment. Rather inexpensive 32 bit bus mastering SCSI chips with FIFO's in the chip have taken high performance I/O to new levels. The combination of bus mastering on the chip, extremely high performance system processors and increasing levels of silicon integration have combined recently to result in high performance low cost solutions in a single chip.

Performance that was once only available as a host adapter board (with a separate processor, SCSI chip, RAM, and system interface) is now achieved using a single chip. System processors offer a level of MIPS on the motherboard that can easily support I/O silicon on the motherboard, thus simplifying the SCSI port dramatically. This new level of cost effective silicon, combined with a complete software solution will no doubt enable SCSI in a new environment and drive the industry to another level of capability.

CHAPTER

1 GENERAL INFORMATION

1.1 INTRODUCTION

The Am53C974 from Advanced Micro Devices was developed in response to the PC industry's need for a hardware solution which harnessed the speed and flexibility of high bandwidth local and I/O buses. Combining the performance of the PCI local bus with the intelligence of the SCSI I/O bus, Advanced Micro Devices offers this Bus Mastering SCSI Controller for PCI systems. The Am53C974 is one member of a family of AMD's plug compatible PCI products which share a common software solution. These products are compliant with PCI specification rev 2.0; the Am53C974 also complies with ANSI standards X3.131–1986 (SCSI-1) and X3.131–199X (SCSI-2).

The Am53C974 offers a glueless interface to the PCI Bus, making it an ideal choice for motherboard designs. The on-chip state machine which controls SCSI sequences in hardware is coupled with a bus-mastering DMA engine to eliminate the need for an additional RISC processor. The result is a PCI-SCSI controller with a superior price/performance advantage. Furthermore, AMD's value-added proposition for the Am53C974 offering is a complete, licensable solution to minimize your time to market.

Distinctive Features:

PCI Bus Interface Unit

- PCI specification rev 2.0 compliant interface
- 32-bit address/data bus master DMA Host interface
- Glueless interface to 33 MHz 32-bit PCI Bus
- 96-byte DMA FIFO for low bus latency

SCSI Controller Features

- Single chip PCI to SCSI interface
- SCSI-2 compliant, 8-bit Fast SCSI interface
- Supports single-ended SCSI bus
- 48 mA SCSI drivers

Performance

- 10 MByte/s synchronous and 7 MByte/s asynchronous transfer rates on the SCSI bus
- 132 MByte/s burst DMA transfer rate
- Supports Scatter-Gather data transfers

Power Management

- Sleep mode capability for Fast SCSI block — Power down for SCSI receivers
- SCSI activity monitoring pin
- Fully static design for low frequency operation

Reliability

- AMD's patented Programmable GLITCH EATER™ Circuitry on $\overline{\text{REQ}}$, $\overline{\text{ACK}}$, and data lines
- Programmable Active Negation on $\overline{\text{REQ}}$, $\overline{\text{ACK}}$ and data lines

Other Features

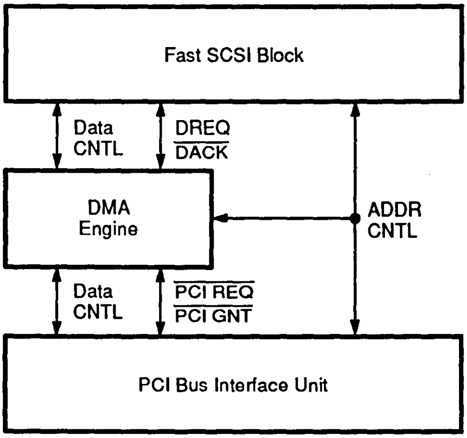
- 132-pin PQFP
- Uses state of the art advanced CMOS technology

The Am53C974 hardware solution is complemented by an extensive software package for all major operating systems. The software solution incorporates AMD's portable SCSI software which is based in part on the Microsoft® Windows NT Miniport™ model.

1.2 HARDWARE

The Am53C974 is a high performance PCI local bus-SCSI controller. It is comprised of the PCI Bus Interface Unit (BIU), a Fast SCSI block, and a bus master DMA engine. The PCI BIU consists of configuration space and a PCI master/slave interface as defined in rev 2.0 of the PCI specification. Figure 1-1 shows the basic interface block diagram of the Am53C974.

Figure 1-1 PCI-DMA-SCSI Interface Block Diagram



18264B-1

1.2.1 Fast SCSI Block

The Am53C974's Fast SCSI block supports SCSI transfer rates of up to 10 MBytes/s synchronously, and up to 7 MByte/s asynchronously. The Am53C974 combines this functionality with features such as programmable Active Negation, and a 24-bit transfer counter. AMD's proprietary features such as power-down mode for SCSI receivers and programmable GLITCH EATER Circuitry are also included for improved product performance.

The Am53C974 has an 8-bit SCSI data interface and operates as an Initiator to support host applications. The SCSI block is designed to minimize host intervention by implementing common SCSI sequences in hardware. An on-chip state machine reduces protocol overhead by performing the required sequences in response to a single command from the host.

The 16-byte SCSI FIFO further assists in minimizing host involvement. The FIFO provides a temporary storage for all command, data, status and message bytes as they are transferred between the 32-bit host data bus and the 8-bit SCSI data bus. Parity on the DMA bus is optional, however, parity is generated in the SCSI block as data is loaded into the SCSI FIFO. Data transfers between the SCSI bus and the SCSI FIFO are internal processes that the Am53C974 handles without microprocessor intervention.

1.2.1.1 Features

Key features in the Am53C974 Fast SCSI block are highlighted below:

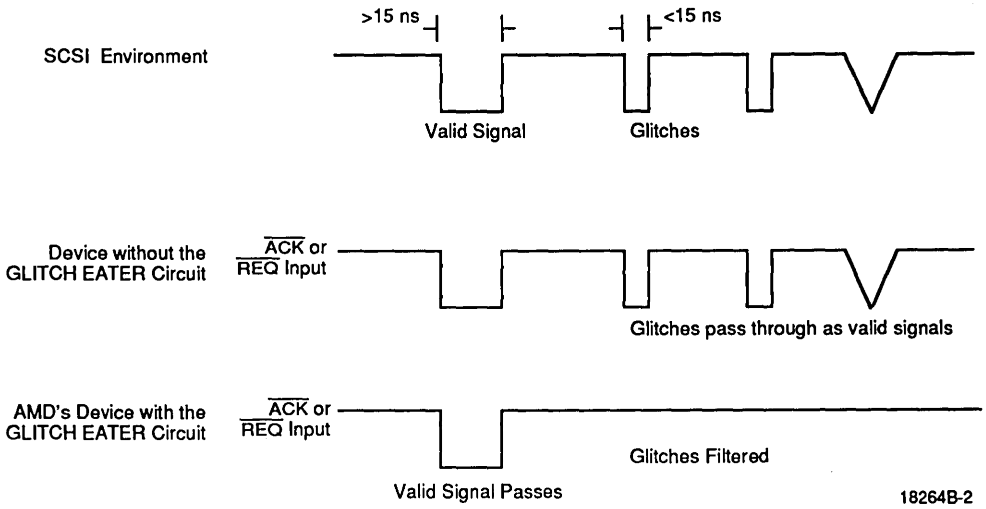
1.2.1.1.1 Reduced Power Mode

AMD's exclusive power-down feature can be enabled to help reduce power consumption. The receivers on the SCSI bus may be turned off to eliminate current flow due to termination power (~3 V) near the trip point of the input buffers.

1.2.1.1.2 Programmable GLITCH EATER Circuitry

The patented GLITCH EATER Circuitry in the Am53C974 PCscsi Controller can be programmed to filter glitches with widths up to 35 ns. It is designed to dramatically increase system reliability by detecting and removing glitches that may cause system failure. The GLITCH EATER Circuitry is implemented on the $\overline{\text{REQ}}$, $\overline{\text{ACK}}$, and data lines since they are most susceptible to electrical anomalies such as reflections and voltage spikes. Such signal inconsistencies can trigger false $\overline{\text{REQ}}/\overline{\text{ACK}}$ handshaking, false data transfers, addition of random data, and double clocking. AMD's GLITCH EATER Circuitry therefore maintains system performance and improves reliability. The following diagram illustrates this circuit's operation.

Figure 1-2 GLITCH EATER Circuitry Operation


>15 ns
<15 ns
SCSI Environment
Valid Signal
Glitches
Device without the GLITCH EATER Circuit
$\overline{\text{ACK}}$ or $\overline{\text{REQ}}$ Input
Glitches pass through as valid signals
AMD's Device with the GLITCH EATER Circuit
$\overline{\text{ACK}}$ or $\overline{\text{REQ}}$ Input
Glitches Filtered
Valid Signal Passes
18264B-2

By default, this feature is enabled and will filter glitches with widths up to 12 ns. When this feature is implemented, the setup and hold times for the following parameters are modified. However, they are still compliant with the SCSI-2 specification and have no effect on the Am53C974's ability to meet Fast SCSI timings.

	0 ns Window	**12 ns Window**
Data to $\overline{\text{REQ}}$ or $\overline{\text{ACK}}$ Setup Time:		
Fast SCSI ANSI requirement:	25 ns	
Normal SCSI ANSI requirement:	55 ns	
Data to $\overline{\text{REQ}}$ Set-Up Time	0 ns	12 ns
Data to $\overline{\text{ACK}}$ Set-Up Time	0 ns	12 ns
Data to $\overline{\text{REQ}}$ or $\overline{\text{ACK}}$ Set-Up Time	5 ns	12 ns

1.2.1.1.3 Programmable Active Negation

AMD offers programmable active negation, a feature which, when implemented, will actively drive the $\overline{\text{REQ}}$, $\overline{\text{ACK}}$ and SCSI Data lines to a high state. This feature is especially helpful for reducing SCSI Bus noise and improving data reliability. By actively driving these signals to their high state, Active Negation eliminates unwanted signal transitions and associated data double-clocking.

This feature is controlled by bits 3:2 in the SCSI Control Register Four ((B)+34h), and may be implemented on $\overline{\text{REQ}}$ and $\overline{\text{ACK}}$, or on $\overline{\text{REQ}}$, $\overline{\text{ACK}}$, and data lines during all SCSI Bus phases except Arbitration and Selection. For more information on programming options, refer to Control Register Four ((B)+34h) bit level descriptions.

1.2.2 DMA Engine

The Am53C974 bridges the PCI and SCSI buses by providing a buffer for these buses. A 96-byte DMA FIFO (24 Double Words) internally interfaces with the 16-byte SCSI FIFO to provide temporary storage for command, data, status, and message bytes as they are transferred between the two buses.

The DMA engine is also capable of handling block type transfers (4 KB pages) during scatter-gather operations. Odd/even boundary conditions are handled through hardware to minimize software overhead.

1.3 Software

To minimize your time to market, AMD offers a complete software solution for the Am53C974. This combination represents a powerful PCI systems solution which enhances the flexibility of your system.

1.3.1 AMD's PCscsi Software Solution

AMD's PCscsi Software maximizes reusability and portability of SCSI protocol chip and device driver source code across multiple operating system platforms. The software architecture was based in part on the Microsoft® Windows NT SCSI Miniport™ Driver model.

AMD's SCSI Software architecture supports the following features:

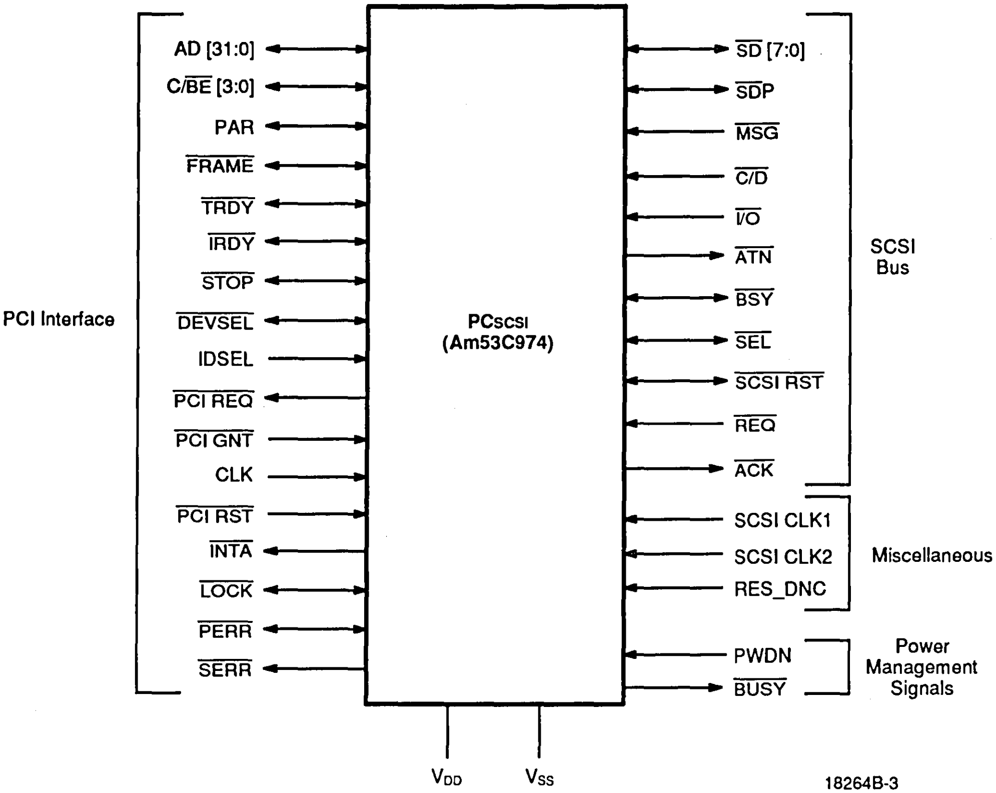
- Device level overlapped/multithreaded operation
- Tagged-queuing
- Automatic request sense
- Scatter-gather operations
- Synchronous transfers (including Fast SCSI)

CHAPTER

2 SIGNAL DESCRIPTIONS

2.1 LOGIC SYMBOL

Figure 2-1 Am53C974 Logic Symbol



2.2 QUICK REFERENCE PIN DESCRIPTIONS

Pin Name	Pin Type	Description
PCI		
AD [31:00]	IN/OUT	Address/Data Bus
C/$\overline{\text{BE}}$ [3:0]	IN/OUT	Command/Byte Enable signals
PAR	IN/OUT	Parity Signal
$\overline{\text{FRAME}}$	IN/OUT	Cycle Frame
$\overline{\text{TRDY}}$	IN/OUT	Target Ready
$\overline{\text{IRDY}}$	IN/OUT	Initiator Ready
$\overline{\text{STOP}}$	IN/OUT	Stop
$\overline{\text{LOCK}}$	IN/OUT	Lock
IDSEL	IN	Initialization Device Select
$\overline{\text{DEVSEL}}$	IN/OUT	Device Select
$\overline{\text{PCI REQ}}$	OUT	PCI Request
$\overline{\text{PCI GNT}}$	IN	PCI Grant
CLK	IN	PCI Clock
$\overline{\text{PCI RST}}$	IN	PCI Reset
$\overline{\text{PERR}}$	IN/OUT	Parity Error
$\overline{\text{SERR}}$	OUT	System Error
$\overline{\text{INTA}}$	OUT	Interrupt
SCSI Interface		
$\overline{\text{SD}}$ [7:0]	IN/OUT	SCSI Data
$\overline{\text{SDP}}$	IN/OUT	SCSI Data Parity
$\overline{\text{MSG}}$	IN	Message
$\overline{\text{C/D}}$	IN	Command/Data
$\overline{\text{I/O}}$	IN	Input/Output
$\overline{\text{ATN}}$	OUT	Attention
$\overline{\text{BSY}}$	IN/OUT	Busy
$\overline{\text{SEL}}$	IN/OUT	Select
$\overline{\text{SCSI RST}}$	IN/OUT	SCSI Bus Reset
$\overline{\text{REQ}}$	IN	Request
$\overline{\text{ACK}}$	OUT	Acknowledge
Miscellaneous		
SCSI CLK1	IN	SCSI Core Clock
SCSI CLK2	IN	Chip Reset Clock
RES_DNC	IN	Reserved, DO NOT CONNECT
Power Management		
PWDN	IN	Power Down Indicator
$\overline{\text{BUSY}}$	OUT	SCSI Bus Activity Pin
Power Supply		
V_{DD}		+5 V
V_{SS}		GND
V_{DDB}		+5 V (Buffer)
V_{SSB}		GND (Buffer)
V_{DD3B}		+5 V (PCI)
V_{SS3B}		GND (5 V PCI)

2.3 SIGNAL DESCRIPTIONS

2.3.1 Address and Data Pins

AD (31:00)
Address/Data (Input/Output, Active High)

These signals are multiplexed on the same PCI pins. During the first clock of a transaction the AD (31:00) contains the physical byte address (32 bits). During the subsequent clocks AD (31:00) contains data. Little-endian byte ordering is used. AD (07:00) is defined as least significant byte and AD (31:24) is defined as the most significant byte.

When $\overline{\text{PCI RST}}$ is active, AD(31:00) are inputs for NAND tree testing.

C/$\overline{\text{BE}}$ (3:0)
Bus Command/Byte Enable (Input/Output, Active Low)

These signals are multiplexed on the same PCI pins. During the address phase of the transaction, C/$\overline{\text{BE}}$(3:0) define the bus command. During the data phase C/$\overline{\text{BE}}$ [3:0] are used as Byte Enables. The Byte Enables define which byte lanes carry meaningful data. C/$\overline{\text{BE}}$(0) applies to byte 0 and C/$\overline{\text{BE}}$(3) applies to byte 3.

When $\overline{\text{PCI RST}}$ is active, C/$\overline{\text{BE}}$ (3:0) are inputs for NAND tree testing.

PAR
Parity (Input/Output, Active High)

Parity is even across AD(31:00) and C/$\overline{\text{BE}}$ (3:0). Parity is generated and driven during Master Address Cycle, Memory Write, I/O Read, and Configuration Read cycles. Parity is checked during Slave Address Cycle, Memory Read, I/O Write, and Configuration Write cycles.

When $\overline{\text{PCI RST}}$ is active, PAR is an input for NAND tree testing.

2.3.2 PCI Interface Control Pins

$\overline{\text{FRAME}}$
Cycle Frame (Input/Output, Active Low)

This signal is driven by the Am53C974 when it is the bus master to indicate the beginning and duration of the access. $\overline{\text{FRAME}}$ is asserted to indicate that bus transaction is beginning. $\overline{\text{FRAME}}$ is asserted while data transfers continue. $\overline{\text{FRAME}}$ is driven low when the transaction is in the final data phase.

When $\overline{\text{PCI RST}}$ is active, $\overline{\text{FRAME}}$ is an input for NAND tree testing.

$\overline{\text{TRDY}}$
Target Ready (Input/Output, Active Low)

When the Am53C974 is selected as a slave, it will drive(low) this signal to indicate its ability to complete the current data phase of the transaction. As a master, this signal is an input to the Am53C974 from the selected (slave) device.

$\overline{\text{TRDY}}$ is used in conjunction with $\overline{\text{IRDY}}$ to indicate completion of the data phase. The data phase is complete (on any clock) when both $\overline{\text{TRDY}}$ and $\overline{\text{IRDY}}$ are sampled asserted. During a read transaction, $\overline{\text{TRDY}}$ is asserted when valid data is present on AD (31:00), while during a write transaction, $\overline{\text{TRDY}}$ asserted indicates the target is prepared to accept data. Wait cycles are inserted until both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted together.

When $\overline{\text{PCI RST}}$ is active, $\overline{\text{TRDY}}$ is an input for NAND tree testing.

IRDY
Initiator Ready (Input/Output, Active Low)

When the Am53C974 is the initiator (master), it will drive (low) this signal to indicate its ability to complete the current data phase of the transaction. As a slave, this signal is an input to the Am53C974 from the initiating (master) device.

IRDY is used in conjunction with TRDY to indicate completion of the data phase. The data phase is complete (on any clock) when both IRDY and TRDY are sampled asserted. During a read transaction, IRDY asserted indicates the master is prepared to accept data, while during a write transaction, IRDY is asserted to indicate that valid data is present on AD (31:00). Wait cycles are inserted until both IRDY and TRDY are asserted together.

When PCI RST is active, IRDY is an input for NAND tree testing.

STOP
Stop (Input/Output, Active Low)

In the slave role the Am53C974 drives the STOP signal to indicate to the bus master to stop the current transaction. In the bus master role the Am53C974 receives the STOP signal and stops the current transaction.

When PCI RST is active, STOP is an input for NAND tree testing.

LOCK
Lock (Input/Output, Active Low)

In the master role the Am53C974 drives the LOCK signal to indicate to the slave device that multiple transactions may be necessary to complete an operation. When LOCK is asserted, non-exclusive transactions may proceed. Control of LOCK is obtained under its own protocol in conjunction with PCI GNT. In the slave role the Am53C974 receives the LOCK signal from the master.

When PCI RST is active, LOCK is an input for NAND tree testing.

Note: *In the current implementation, the Am53C974 as a master will never generate a LOCK. However in slave role, the chip will respond to a LOCK asserted by a master.*

IDSEL
Initialization Device Select (Input, Active High)

This signal is used as a chip select for the Am53C974 in lieu of the 24 address lines during configuration read and write transaction.

When PCI RST is active, IDSEL is an input for NAND tree testing.

DEVSEL
Device Select (Input/Output, Active Low)

This signal when actively driven by the Am53C974 as a slave device signals to the master device that it has decoded its address as the target of the current access. As an input it indicates whether any device on the bus has been selected.

When PCI RST is active, DEVSEL is an input for NAND tree testing.

2.3.3 Arbitration Pins

$\overline{\text{PCI REQ}}$
PCI Request (Output, Active Low, Tristate)

This signal indicates to the arbiter that the Am53C974 desires use of the bus. This is a point to point signal. Every master has its own equivalent of $\overline{\text{PCI REQ}}$, which will be tristated after a power-up or a chip reset.

When $\overline{\text{PCI RST}}$ is active, $\overline{\text{PCI REQ}}$ is an input for NAND tree testing.

$\overline{\text{PCI GNT}}$
PCI Grant (Input, Active Low)

This signal indicates that the access to the bus has been granted to the Am53C974. This is a point to point signal. Every master has its own equivalent of $\overline{\text{PCI GNT}}$.

When $\overline{\text{PCI RST}}$ is active, $\overline{\text{PCI GNT}}$ is an input for NAND tree testing.

2.3.4 System Pins

CLK
Clock (Input)

This signal provides timing for all the transactions on the PCI bus and all PCI devices on the bus including the Am53C974. All signals are sampled on the rising edge of CLK and all parameters are defined with respect to this edge. The Am53C974 operates up to 33 MHz.

When $\overline{\text{PCI RST}}$ is active, CLK is an input for NAND tree testing.

$\overline{\text{PCI RST}}$
PCI Reset (Input, Active Low)

This signal forces the Am53C974 sequencer to a known state. All Three-State bidirectional signals are forced to a high impedance state and all Sustained Open Drain signals are allowed to float high. The Am53C974 will tristate $\overline{\text{PCI REQ}}$, and completely reset the Am53C974. $\overline{\text{PCI RST}}$ may be asynchronous to the CLK when asserted or driven low. It is recommended that the deassertion be synchronous to guarantee a clean and bounce free edge.

When $\overline{\text{PCI RST}}$ is active, NAND tree testing is enabled. All PCI interface pins are input mode. The result of the NAND tree testing can be observed on the $\overline{\text{BUSY}}$ output (pin 62).

2.3.5 Error Reporting Pins

$\overline{\text{PERR}}$
Parity Error (Input/Output, Active Low, Open Drain)

This signal may be pulsed by the Am53C974 when it detects a parity error during any data phase when its AD (31:00) and C/$\overline{\text{BE}}$ (3:0) lines are inputs. The Am53C974 monitors the PERR input during a bus master write cycle. It will assert the Data Parity Reported bit in the Status Register of the PCI Configuration Space when a parity error is reported by the target device.

When $\overline{\text{PCI RST}}$ is active, $\overline{\text{PERR}}$ is an input for NAND tree testing.

$\overline{\text{SERR}}$

System Error (Output, Active Low, Open Drain)

This signal may be pulsed by the Am53C974 for reporting address parity errors when AD(31:00) are inputs.

When $\overline{\text{PCI RST}}$ is active, $\overline{\text{SERR}}$ is an input for NAND tree testing.

2.3.6 Interrupt Request Pins

$\overline{\text{INTA}}$

Interrupt Request (Output, Active Low, Open Drain)

This signal combines the interrupt request from both the DMA engine and the SCSI block. The interrupt source can be determined by reading the DMA Status Register. Interrupts caused by the DMA engine will be cleared when the DMA Status Register is read. However, for those interrupts generated by the SCSI block, the SCSI interrupt register must be serviced in order to clear the interrupt.

When $\overline{\text{PCI RST}}$ is active, $\overline{\text{INTA}}$ is an input for NAND tree testing.

2.3.7 SCSI Interface Signals

$\overline{\text{SD}}$ (7:0)

SCSI Data (Input/Output, Active Low, Open Drain/Active Negation, Schmitt Trigger)

These pins are defined as bi-directional SCSI data bus.

$\overline{\text{SDP}}$

SCSI Data Parity

(Input/Output, Active Low, Open Drain/Active Negation, Schmitt Trigger)

This pin is defined as bi-directional SCSI data parity.

$\overline{\text{MSG}}$

Message (Input, Active Low, Schmitt Trigger)

This is a Schmitt trigger input.

$\overline{\text{C/D}}$

Command/Data (Input, Schmitt Trigger)

This is a Schmitt trigger input.

$\overline{\text{I/O}}$

Input/Output (Input, Schmitt Trigger)

This is a Schmitt trigger input.

$\overline{\text{ATN}}$

Attention (Output, Active Low, Schmitt Trigger)

This signal is an 48 mA output. It will be asserted when the device detects a parity error or it can be asserted via certain commands.

$\overline{\text{BSY}}$

Busy (Input/Output, Active Low, Schmitt Trigger/Open Drain)

As a SCSI input signal it has a Schmitt trigger and as an output signal it has a 48 mA drive.

$\overline{\text{SEL}}$
Select (Input/Output, Active Low, Schmitt Trigger/Open Drain)

As a SCSI input signal it has a Schmitt trigger and as an output signal it has a 48 mA drive.

$\overline{\text{SCSI RST}}$
Reset (Input /Output, Active Low, Schmitt Trigger/Open Drain)

As a SCSI input signal it has a Schmitt trigger and as an output signal it has a 48 mA drive. The Reset SCSI command causes the device to drive $\overline{\text{SCSIRST}}$ active for 25 µs – 40 µs, depending on the CLK frequency and conversion factor.

$\overline{\text{REQ}}$
Request (Input, Active Low, Schmitt Trigger)

This is a SCSI input signal with a Schmitt trigger.

$\overline{\text{ACK}}$
Acknowledge (Output, Active Low, Open Drain/Active Negation)

This is a SCSI output signal with a 48 mA drive.

2.3.8 Power Management Signals

PWDN
Power Down Indicator (Input, Active High)

This signal, when asserted, sets the PWDN status bit in the DMA status register and sends an interrupt to the host.

When $\overline{\text{PCI RST}}$ is active, PWDN is an input for NAND tree testing.

$\overline{\text{BUSY}}$
SCSI Devices Busy (Output, Active Low)

This signal is the logical equivalent of the SCSI bus $\overline{\text{BSY}}$ signal. It is duplicated so that external logic can be connected to monitor SCSI bus activity.

When $\overline{\text{PCI RST}}$ is active, the results of the NAND tree testing can be observed on $\overline{\text{BUSY}}$. When $\overline{\text{PCI RST}}$ is driven low, $\overline{\text{BUSY}}$ will function as described above.

2.3.9 Miscellaneous Signals

SCSI CLK1
SCSI Clock (Input)

The SCSI clock signal is used to generate all internal device timings. The maximum frequency of this input is 40 MHz, while the minimum is 10 MHz to maintain SCSI bus timing requirements. To achieve Fast SCSI timings, a 40 MHz clock must be supplied to this input.

SCSI CLK2
SCSI Clock (Input)

This clock is required to properly reset the Am53C974. A minimum of 30 clock cycles are needed to reset the chip properly. The clock value must be between 16 MHz and 40 MHz. For board layout convenience, SCSI CLK1 and SCSI CLK2 may be connected to a common clock source.

RES_DNC
Reserved _Do Not Connect (Input)

This pin (#116) is reserved for factory testing. To ensure proper chip operation, **It must not be connected.**

2.3.10 Power Supply Pins

V_{DD}
+5 V *Power* (Input)

These inputs provide power necessary to operate the Am53C974. All V_{DD} pins must be connected to a +5 V source.

V_{DDB}
+5 V *Power* (Input)

These inputs are for SCSI Buffers. These pins can be connected to the V_{DD} pins.

V_{DD3B}
+5 V *Power* (Input)

These inputs provide power for the PCI Interface block. These pins must be connected to a +5 V source.

$V_{SS}/V_{SSB}/V_{SS3B}$
Ground (Input)

These inputs provide the necessary grounds to operate the Am53C974. The V_{SSB} and V_{SS3B} can be connected to V_{SS} provided there is a decoupling capacitor between V_{SSB} and V_{DDB} and V_{SS3B} and V_{DD3B}.

2.4 CONNECTION DIAGRAM TABLES

2.4.1 Listed by Pin Number

Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
1	V_{DD3B}	34	PAR	67	V_{SSB}	100	V_{SS}
2	AD27	35	C/$\overline{\text{BE}}$1	68	$\overline{\text{SD}}$0	101	NC
3	AD26	36	AD15	69	$\overline{\text{SD}}$1	102	NC
4	V_{SS3B}	37	V_{SS3B}	70	$\overline{\text{SD}}$2	103	V_{DD}
5	AD25	38	AD14	71	$\overline{\text{SD}}$3	104	NC
6	AD24	39	AD13	72	V_{SSB}	105	NC
7	C/$\overline{\text{BE}}$3	40	AD12	73	$\overline{\text{SD}}$4	106	NC
8	V_{DD}	41	AD11	74	$\overline{\text{SD}}$5	107	NC
9	IDSEL	42	AD10	75	$\overline{\text{SD}}$6	108	V_{DD}
10	NC	43	V_{SS3B}	76	V_{DDB}	109	V_{DD}
11	V_{SS}	44	AD9	77	$\overline{\text{SD}}$7	110	NC
12	AD23	45	AD8	78	$\overline{\text{SDP}}$	111	NC
13	AD22	46	V_{DD3B}	79	V_{SS}	112	NC
14	V_{SS3B}	47	C/$\overline{\text{BE}}$0	80	$\overline{\text{SEL}}$	113	V_{SS}
15	AD21	48	AD7	81	$\overline{\text{REQ}}$	114	NC
16	AD20	49	AD6	82	V_{SSB}	115	NC
17	V_{DD3B}	50	V_{SS3B}	83	$\overline{\text{ACK}}$	116	RES_DNC
18	AD19	51	AD5	84	V_{DD}	117	$\overline{\text{INTA}}$
19	AD18	52	AD4	85	$\overline{\text{MSG}}$	118	NC
20	V_{SS3B}	53	AD3	86	$\overline{\text{C/D}}$	119	V_{SS}
21	AD17	54	AD2	87	$\overline{\text{I/O}}$	120	$\overline{\text{PCI RST}}$
22	AD16	55	V_{SS3B}	88	V_{SS}	121	CLK
23	C/$\overline{\text{BE}}$2	56	AD1	89	NC	122	V_{DD}
24	$\overline{\text{FRAME}}$	57	AD0	90	NC	123	NC
25	$\overline{\text{IRDY}}$	58	PWDN	91	V_{DD}	124	$\overline{\text{PCI GNT}}$
26	$\overline{\text{TRDY}}$	59	V_{DD}	92	NC	125	V_{SS}
27	$\overline{\text{DEVSEL}}$	60	SCSICLK1	93	NC	126	NC
28	$\overline{\text{STOP}}$	61	V_{SS}	94	NC	127	$\overline{\text{PCI REQ}}$
29	$\overline{\text{LOCK}}$	62	$\overline{\text{BUSY}}$	95	NC	128	AD31
30	V_{SS}	63	V_{SS}	96	V_{DD}	129	AD30
31	$\overline{\text{PERR}}$	64	$\overline{\text{BSY}}$	97	SCSICLK2	130	V_{SS3B}
32	$\overline{\text{SERR}}$	65	$\overline{\text{ATN}}$	98	V_{SS}	131	AD29
33	V_{DD3B}	66	$\overline{\text{SCSI RST}}$	99	NC	132	AD28

NC = No Connect

RES_DNC = Reserved_DO NOT CONNECT.

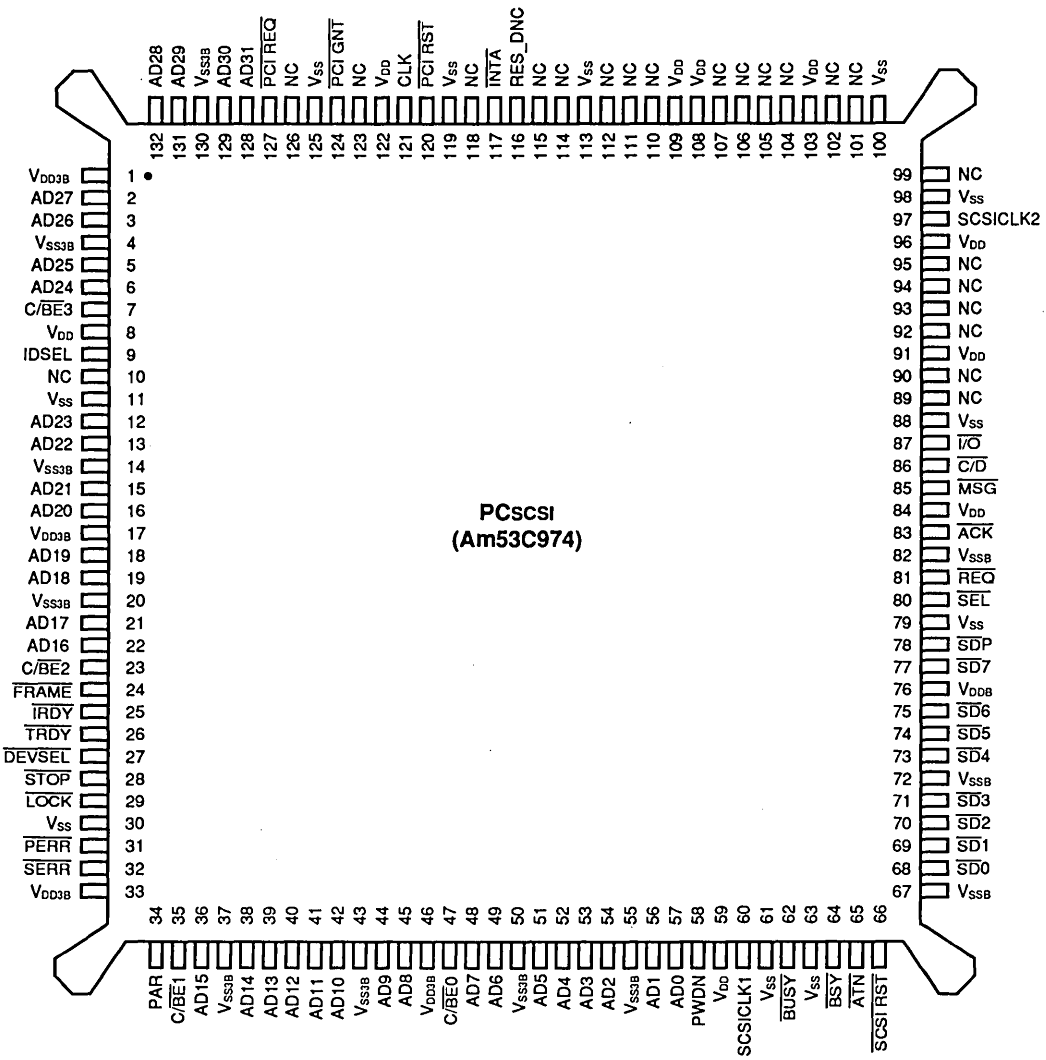
2.4.2 Listed by Pin Name

Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.
$\overline{\text{ACK}}$	83	$\overline{\text{ATN}}$	65	NC	112	V_{DD}	91
AD0	57	$\overline{\text{BSY}}$	64	NC	114	V_{DD}	96
AD1	56	$\overline{\text{BUSY}}$	62	NC	115	V_{DD}	103
AD2	54	C/$\overline{\text{BE}}$0	47	NC	118	V_{DD}	108
AD3	53	C/$\overline{\text{BE}}$1	35	NC	123	V_{DD}	109
AD4	52	C/$\overline{\text{BE}}$2	23	NC	126	V_{DD}	122
AD5	51	C/$\overline{\text{BE}}$3	7	PAR	34	V_{DD3B}	1
AD6	49	$\overline{\text{C/D}}$	86	$\overline{\text{PCI GNT}}$	124	V_{DD3B}	17
AD7	48	CLK	121	$\overline{\text{PCI REQ}}$	127	V_{DD3B}	33
AD8	45	$\overline{\text{DEVSEL}}$	27	$\overline{\text{PCI RST}}$	120	V_{DD3B}	46
AD9	44	$\overline{\text{FRAME}}$	24	$\overline{\text{PERR}}$	31	V_{DDB}	76
AD10	42	$\overline{\text{I/O}}$	87	PWDN	58	V_{SS}	11
AD11	41	IDSEL	9	$\overline{\text{REQ}}$	81	V_{SS}	30
AD12	40	$\overline{\text{INTA}}$	117	RES_DNC	116	V_{SS}	61
AD13	39	$\overline{\text{IRDY}}$	25	$\overline{\text{SCSI RST}}$	66	V_{SS}	63
AD14	38	$\overline{\text{LOCK}}$	29	SCSICLK1	60	V_{SS}	79
AD15	36	$\overline{\text{MSG}}$	85	SCSICLK2	97	V_{SS}	88
AD16	22	NC	10	$\overline{\text{SD0}}$	68	V_{SS}	98
AD17	21	NC	89	$\overline{\text{SD1}}$	69	V_{SS}	100
AD18	19	NC	90	$\overline{\text{SD2}}$	70	V_{SS}	113
AD19	18	NC	92	$\overline{\text{SD3}}$	71	V_{SS}	119
AD20	16	NC	93	$\overline{\text{SD4}}$	73	V_{SS}	125
AD21	15	NC	94	$\overline{\text{SD5}}$	74	V_{SS3B}	4
AD22	13	NC	95	$\overline{\text{SD6}}$	75	V_{SS3B}	14
AD23	12	NC	99	$\overline{\text{SD7}}$	77	V_{SS3B}	20
AD24	6	NC	101	$\overline{\text{SDP}}$	78	V_{SS3B}	37
AD25	5	NC	102	$\overline{\text{SEL}}$	80	V_{SS3B}	43
AD26	3	NC	104	$\overline{\text{SERR}}$	32	V_{SS3B}	50
AD27	2	NC	105	$\overline{\text{STOP}}$	28	V_{SS3B}	55
AD28	132	NC	106	$\overline{\text{TRDY}}$	26	V_{SS3B}	130
AD29	131	NC	107	V_{DD}	8	V_{SSB}	67
AD30	129	NC	110	V_{DD}	59	V_{SSB}	72
AD31	128	NC	111	V_{DD}	84	V_{SSB}	82

NC = No Connect

RES_DNC = Reserved_DO NOT CONNECT.

2.5 PIN OUT MAP


PCscsi
(Am53C974)
1 V_{DD3B}
2 AD27
3 AD26
4 V_{SS3B}
5 AD25
6 AD24
7 C/$\overline{\text{BE3}}$
8 V_{DD}
9 IDSEL
10 NC
11 V_{SS}
12 AD23
13 AD22
14 V_{SS3B}
15 AD21
16 AD20
17 V_{DD3B}
18 AD19
19 AD18
20 V_{SS3B}
21 AD17
22 AD16
23 C/$\overline{\text{BE2}}$
24 $\overline{\text{FRAME}}$
25 $\overline{\text{IRDY}}$
26 $\overline{\text{TRDY}}$
27 $\overline{\text{DEVSEL}}$
28 $\overline{\text{STOP}}$
29 $\overline{\text{LOCK}}$
30 V_{SS}
31 $\overline{\text{PERR}}$
32 $\overline{\text{SERR}}$
33 V_{DD3B}
34 PAR
35 C/$\overline{\text{BE1}}$
36 AD15
37 V_{SS3B}
38 AD14
39 AD13
40 AD12
41 AD11
42 AD10
43 V_{SS3B}
44 AD9
45 AD8
46 V_{DD3B}
47 C/$\overline{\text{BE0}}$
48 AD7
49 AD6
50 V_{SS3B}
51 AD5
52 AD4
53 AD3
54 AD2
55 V_{SS3B}
56 AD1
57 AD0
58 PWDN
59 V_{DD}
60 SCSICLK1
61 V_{SS}
62 $\overline{\text{BUSY}}$
63 V_{SS}
64 $\overline{\text{BSY}}$
65 $\overline{\text{ATN}}$
66 $\overline{\text{SCSI RST}}$
67 V_{SSB}
68 $\overline{\text{SD0}}$
69 $\overline{\text{SD1}}$
70 $\overline{\text{SD2}}$
71 $\overline{\text{SD3}}$
72 V_{SSB}
73 $\overline{\text{SD4}}$
74 $\overline{\text{SD5}}$
75 $\overline{\text{SD6}}$
76 V_{DDB}
77 $\overline{\text{SD7}}$
78 $\overline{\text{SDP}}$
79 V_{SS}
80 $\overline{\text{SEL}}$
81 $\overline{\text{REQ}}$
82 V_{SSB}
83 $\overline{\text{ACK}}$
84 V_{DD}
85 $\overline{\text{MSG}}$
86 $\overline{\text{C/D}}$
87 $\overline{\text{I/O}}$
88 V_{SS}
89 NC
90 NC
91 V_{DD}
92 NC
93 NC
94 NC
95 NC
96 V_{DD}
97 SCSICLK2
98 V_{SS}
99 NC
100 V_{SS}
101 NC
102 NC
103 V_{DD}
104 NC
105 NC
106 NC
107 NC
108 V_{DD}
109 V_{DD}
110 NC
111 NC
112 NC
113 V_{SS}
114 NC
115 NC
116 RES_DNC
117 $\overline{\text{INTA}}$
118 NC
119 V_{SS}
120 $\overline{\text{PCI RST}}$
121 CLK
122 V_{DD}
123 NC
124 $\overline{\text{PCI GNT}}$
125 V_{SS}
126 NC
127 $\overline{\text{PCI REQ}}$
128 AD31
129 AD30
130 V_{SS3B}
131 AD29
132 AD28

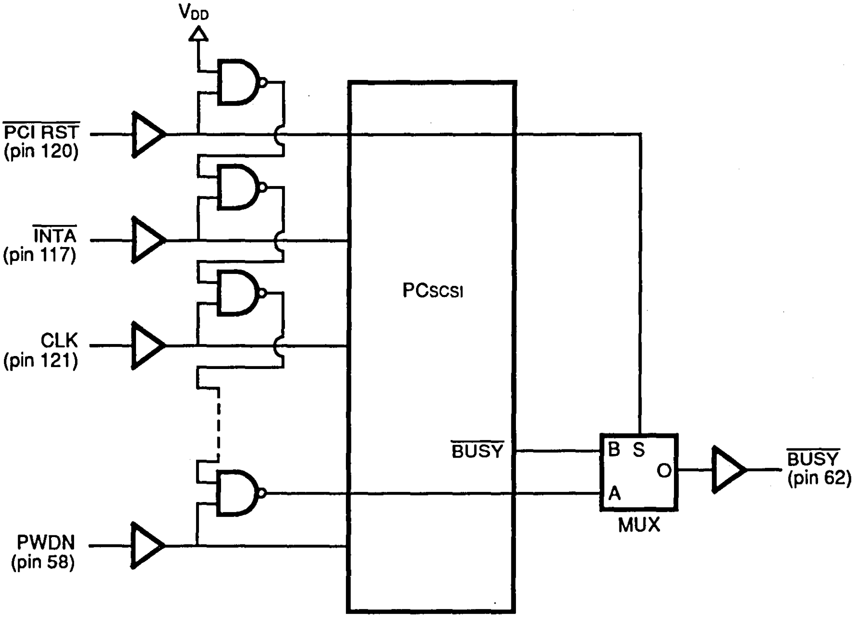
18264B-4

2.6 NAND TREE TESTING

The Am53C974 PCscsi controller provides a NAND tree test mode to allow connectivity checking to the device on a printed circuit board. The NAND tree is built on all PCI bus signals.

The NAND tree test is enabled by asserting $\overline{\text{PCI RST}}$. All PCI signals will become inputs when $\overline{\text{PCI RST}}$ is asserted. The result of the NAND tree test can be observed on the $\overline{\text{BUSY}}$ pin.

Figure 2-2 NAND Tree



18264B-5

Pin 120 ($\overline{\text{PCI RST}}$) is the first input to the NAND tree. Pin 117 ($\overline{\text{INTA}}$) is the second input to the NAND tree, followed by pin 121 (CLK). All other PCI bus signals follow, counterclockwise, with pin 58 (PWDN) being the last. Pins labeled NC and power supply pins are not part of the NAND tree. The table below shows the complete list of pins connected to the NAND tree.

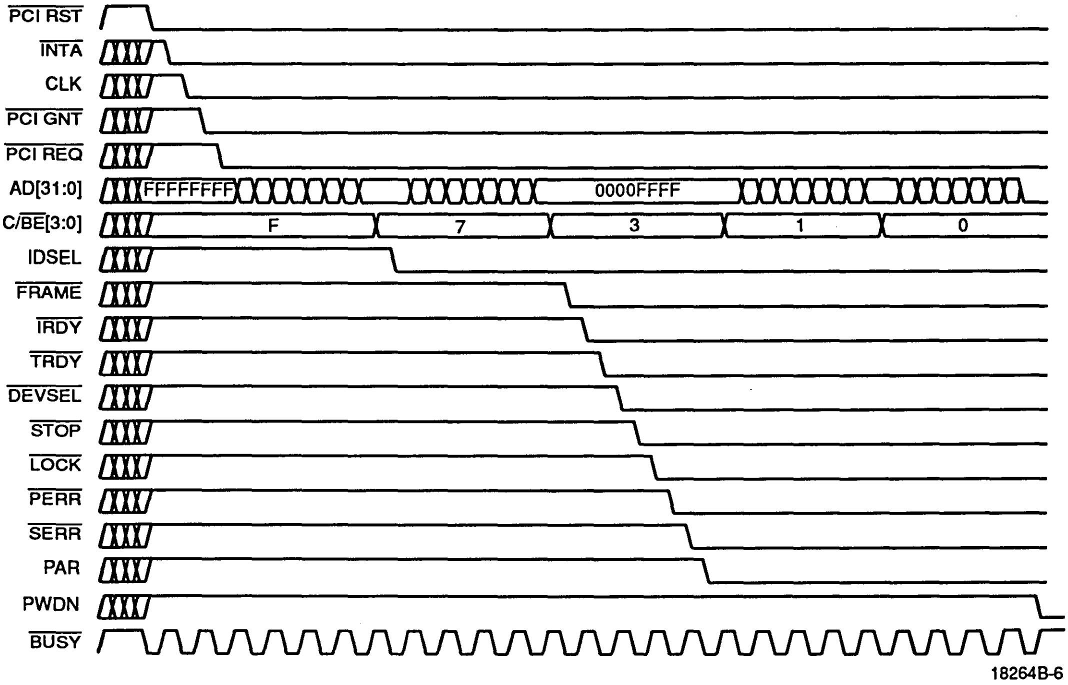
NAND Tree Input #	Pin #	Name	NAND Tree Input #	Pin #	Name	NAND Tree Input #	Pin #	Name
1	120	$\overline{\text{PCI RST}}$	19	16	AD20	37	39	AD13
2	117	$\overline{\text{INTA}}$	20	18	AD19	38	40	AD12
3	121	CLK	21	19	AD18	39	41	AD11
4	124	$\overline{\text{PCI GNT}}$	22	21	AD17	40	42	AD10
5	127	$\overline{\text{PCI REQ}}$	23	22	AD16	41	44	AD9
6	128	AD31	24	23	C/$\overline{\text{BE2}}$	42	45	AD8
7	129	AD30	25	24	$\overline{\text{FRAME}}$	43	47	C/$\overline{\text{BE0}}$
8	131	AD29	26	25	$\overline{\text{IRDY}}$	44	48	AD7
9	132	AD28	27	26	$\overline{\text{TRDY}}$	45	49	AD6
10	2	AD27	28	27	$\overline{\text{DEVSEL}}$	46	51	AD5
11	3	AD26	29	28	$\overline{\text{STOP}}$	47	52	AD4
12	5	AD25	30	29	$\overline{\text{LOCK}}$	48	53	AD3
13	6	AD24	31	31	$\overline{\text{PERR}}$	49	54	AD2
14	7	C/$\overline{\text{BE3}}$	32	32	$\overline{\text{SERR}}$	50	56	AD1
15	9	IDSEL	33	34	PAR	51	57	AD0
16	12	AD23	34	35	C/$\overline{\text{BE1}}$	52	58	PWDN
17	13	AD22	35	36	AD15			
18	15	AD21	36	38	AD14			

$\overline{\text{PCI RST}}$ must be asserted (logic low) to start a NAND tree test sequence. Initially, all NAND tree inputs except $\overline{\text{PCI RST}}$ should be driven high. This will result in a low output at the $\overline{\text{BUSY}}$ pin. If the NAND tree inputs are driven low in the same order as they are connected to build the NAND tree, $\overline{\text{BUSY}}$ will toggle every time an additional input is driven low. $\overline{\text{BUSY}}$ will change to a ONE, when $\overline{\text{INTA}}$ is driven low and all other NAND tree inputs stay high. $\overline{\text{BUSY}}$ will toggle back to low, when CLK is additionally driven low. The square wave will continue until all NAND tree inputs are driven low. $\overline{\text{BUSY}}$ will be low, when all NAND tree inputs are driven low.

When testing is complete, deassert $\overline{\text{PCI RST}}$ to exit this test mode.

Note: Some of the pins connected to the NAND tree are outputs in normal mode of operation. They must not be driven from an external source until the PCscsi controller is configured for NAND tree testing.

Figure 2-3 NAND Tree Waveform



2.7 AM53C974 REGISTER MAP

Configuration Register Map

31	16	15	0	Address Offset
Device ID		Vendor ID		00h
Status		Command		04h
Base Class	Sub Class	Prog. If.	Revision ID	08h
Reserved*	Header Type	Latency Timer	Reserved*	0Ch
Base Address				10h
Reserved*				14h – 38h
Reserved	Reserved	Interrupt Pin	Interrupt Line	3Ch
Reserved for SCSI Software	Reserved for SCSI Software	Reserved for SCSI Software	Reserved for SCSI Software	40h – 4Ch**

** Not Implemented on Am53C974. Writes to these locations will have no effect; reads from these locations will return '00h'.*

*** Reserved for SCSI software.*

SCSI Register Map

Register Acronym	Address (Hex.)	Register Description	Type
CTCREG	(B)+00	Current Transfer Count Register Low	Read
STCREG	(B)+00	Start Transfer Count Register Low	Write
CTCREG	(B)+04	Current Transfer Count Register Middle	Read
STCREG	(B)+04	Start Transfer Count Register Middle	Write
FFREG	(B)+08	SCSI FIFO Register	Read/Write
CMDREG	(B)+0C	SCSI Command Register	Read/Write
STATREG	(B)+10	SCSI Status Register	Read
SDIDREG	(B)+10	SCSI Destination ID Register	Write
INSTREG	(B)+14	Interrupt Status Register	Read
STIMREG	(B)+14	SCSI Timeout Register	Write
ISREG	(B)+18	Internal State Register	Read
STPREG	(B)+18	Synchronous Transfer Period Register	Write
CFIREG	(B)+1C	Current FIFO/Internal State Register	Read
SOFREG1	(B)+1C	Synchronous Offset Register	Write
CNTLREG1	(B)+20	Control Register One	Read/Write
CLKFREG	(B)+24	Clock Factor Register	Write
RES	(B)+28	Reserved	Write
CNTLREG2	(B)+2C	Control Register Two	Read/Write
CNTLREG3	(B)+30	Control Register Three	Read/Write
CNTLREG4	(B)+34	Control Register Four	Read/Write
CTCREG	(B)+38	Current Transfer Count Register High/Part-Unique ID Code	Read
STCREG	(B)+38	Start Current Transfer Count Register High	Write
RES	(B)+3C	Reserved	Write

DMA Register Map

Register Acronym	Address (Hex.)	Register Description	Type
CMD	(B)+40	Command	R/W
STC	(B)+44	Starting Transfer Count	R/W
SPA	(B)+48	Starting Physical Address	R/W
WBC	(B)+4C	Working Byte Counter	R
WAC	(B)+50	Working Address Counter	R
STATUS	(B)+54	Status Register	R
SMDLA	(B)+58	Starting Memory Descriptor List (MDL) Address	R/W
WMAC	(B)+5C	Working MDL Counter	R

CHAPTER

3 POWER MANAGEMENT FEATURES

3.1 INTRODUCTION

As a leader in low-voltage technology, AMD has incorporated power–saving features into the Am53C974. Through hardware and software, the Am53C974 can be powered down to reduce consumption during chip inactivity. This significantly reduces overall power usage, as the system and associated peripherals can benefit from these features.

3.2 SCSI ACTIVITY PIN

The SCSI Bus activity is reflected by the $\overline{\text{BUSY}}$ output line. This signal, when active, indicates that the SCSI Bus is in use, therefore the Am53C974 should not be powered down. This signal is the logical equivalent to the SCSI bus signal $\overline{\text{BSY}}$, however, it is not physically connected to the $\overline{\text{BSY}}$ signal on the bus. To correctly identify the Bus Free State on the SCSI bus, this $\overline{\text{BUSY}}$ output line must be inactive for at least 250 ms (Selection Timeout period).

3.2.1 Reduced Power Mode

When there is no activity on the SCSI Bus, and there are no pending commands, the Am53C974 may be powered down by turning off the input buffers on the SCSI Bus lines (set bit 5 in Control Register Four (B)+34h). For further power reduction, the internal registers may be programmed to a predetermined state, and the input clock disconnected via external logic.

3.3 POWER DOWN PIN (PWDN Pin)

When PWDN is driven active, it sets the PWDN bit in the Status Register (Bit 0, DMA Status Register (B)+54h), signaling the Am53C974 that the host would like to power down the SCSI interface. An interrupt is generated when this bit is first set.

3.3.1 Software Disk Spin-Down

Incorporated into the SCSI ROM BIOS and certain device driver's of AMD's software solution is a module which physically spins down SCSI fixed disks when the host system elects to enact power management on the SCSI system. The software module is activated upon the ROM BIOS and/or other device driver's receipt of an interrupt caused by the PWDN pin being driven active. Upon receipt of this interrupt, the current software process is suspended and software control is given to the power management module.

When the power management module is activated, it checks the status of all SCSI fixed disks on the system under its control. The SCSI fixed disks that are idle are issued a command to spin down their media. All fixed disks that are active at the time will be scheduled for spin down upon completion of their pending commands. Once the BIOS and/or drivers have detected the completion of each fixed disk's final pending commands, they are issued the command to spin down as well.

To spin down the disk drives, the power management module issues a SCSI command (1B– Start/Stop unit). When the command is received by the drive, it spins down and waits in an idle state. For multiple drives on the SCSI bus, the power management driver spins down each drive individually. The drives remain in the idle state until the BIOS and/or driver receives a command for the particular fixed disk. Once this occurs, the drive is issued the command to spin up. When the drive has completely spun up and is ready, the pending command is issued to the drive. Only the particular fixed disk issued the command is instructed to spin up. All other drives will remain in the spun down state until a command is issued to them.

Note: This sequence does not turn off the input buffers as described in the previous section.

CHAPTER 4

THE PCI BUS INTERFACE UNIT

4.1 INTRODUCTION

The Am53C974 handles all PCI bus accesses through its PCI Bus Interface Unit (BIU). The PCI BIU interprets and generates all PCI bus signals in accordance with the PCI specification Rev 2.0. In addition to interfacing the Am53C974 with the PCI bus, the PCI BIU also contains a 256 byte PCI configuration register which is accessible via Configuration Read/Write cycles from the PCI bus. This chapter covers the PCI block of the Am53C974 PCI-SCSI controller. The Am53C974's I/O address map, PCI bus cycles and modes supported are described in detailed as well as the function and contents of its PCI configuration registers. For more information on the PCI bus protocol, refer to the *PCI Local Bus Specification.*

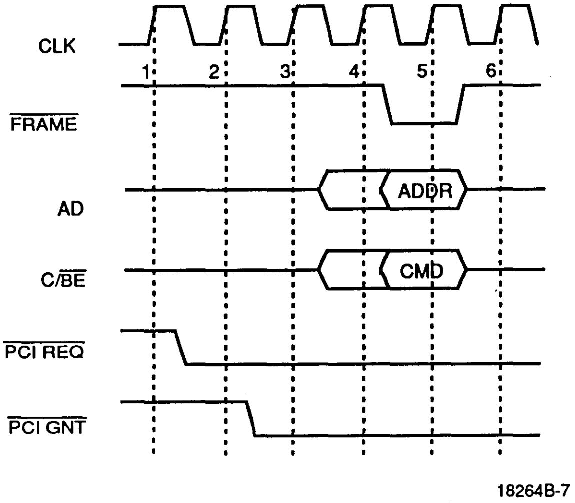
4.2 ADDRESSING

PCI defines three physical address spaces: Memory, I/O, and Configuration. The memory and I/O address space are customary while configuration has been defined to support PCI hardware. Am53C974 accesses to the memory space requires a Memory Read/Write command to the desired memory location while host CPU accesses to the I/O space requires an I/O Read/Write command to the location specified by the Base Address Register of the device's configuration space. That is, the value written to the Base Address Register of the Configuration space defines the I/O location of the Am53C974. Configuration accesses to the Am53C974 are done by issuing a Configuration Read/Write command with the Am53C974 IDSEL line asserted.

4.3 BUS ACQUISITION

The first step in any Am53C974 bus master transfer is to acquire ownership of the bus. This task is handled by synchronous logic within the PCI BIU. Bus ownership is requested with the $\overline{\text{PCI REQ}}$ signal and ownership is granted by the arbiter through the $\overline{\text{PCI GNT}}$ signal. Figure 4-1 shows the Am53C974's bus acquisition timing. In this figure, although bus ownership is granted on clock 3 with the assertion of $\overline{\text{PCI GNT}}$, the Am53C974 will not assert $\overline{\text{FRAME}}$ (indicating the start of bus cycle) until clock 5. Note, however, that although the Am53C974 will begin driving AD[31:0] and C/$\overline{\text{BE}}$[3:0] prior to clock 4, these lines will not be valid until $\overline{\text{FRAME}}$ is asserted. ADSTEP (bit 7) in the PCI command register is set to ONE to indicate that the Am53C974 uses address stepping. However, address stepping is only used for the first address phase of a bus master period.

Figure 4-1 **Bus Acquisition Timing**



4.4 BUS CYCLE DEFINITION

The Am53C974 supports only the relevant PCI bus cycles (seven of the sixteen cycles). Of these seven, three are supported in the Bus Master mode and four in the Slave mode. These cycles are defined by the C/BE [3:0] command lines during the address phase of each PCI bus cycle. Table 4-1 shows these bus cycles and the mode supported by the Am53C974. Note that Bus cycles with an asterisk (*) are ignored by the Am53C974.

Table 4-1 **PCI Bus Cycles Supported by the Am53C974**

C/BE[3:0]	Bus Cycle Type	Mode Supported
0000	Interrupt ACK	*
0001	Special Cycle	*
0010	I/O Read	Slave
0011	I/O Write	Slave
0100	Reserved	*
0101	Reserved	*
0110	Memory Read	Master
0111	Memory Write	Master
1000	Reserved	*
1001	Reserved	*
1010	Config. Read	Slave
1011	Config. Write	Slave
1100	Mem Read Multiple	*
1101	Dual Address Cycle	*
1110	Mem Read Line	Master
1111	Mem Write & Invalidate	*

** These cycles are ignored by the Am53C974.*

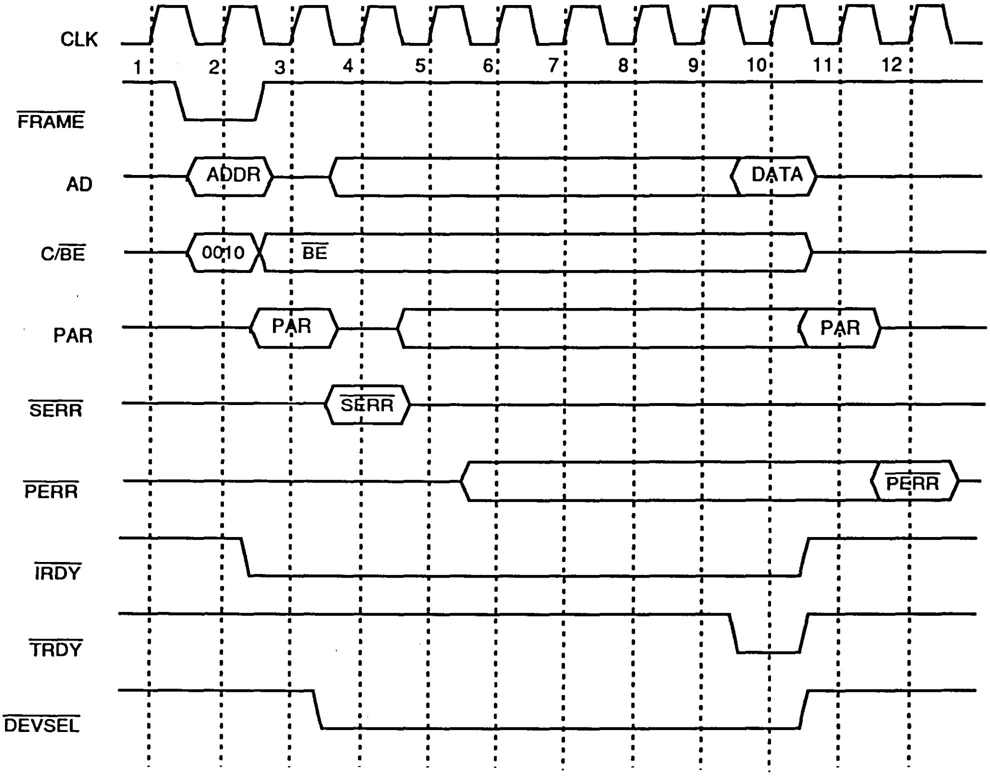
4.5 BUS CYCLE DIAGRAMS

The following are samples of the Am53C974's bus cycles in Table 4-1. Each cycle shows an example timing diagram along with a brief description of the cycle. Note that the cycles shown are only typical PCI bus cycles; each cycle can be distinct because of various factors such as Bus Latency, $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ timing, etc.

4.5.1 Slave I/O Read

The Slave I/O Read command is used by the processor to read internal registers in the Am53C974. It is a single cycle, non-burst 8-bit, 16-bit, or 32-bit transfer which is initiated by the host CPU. The Am53C974 will not produce slave I/O Read commands while a bus master. Slave I/O Read cycles are fixed length cycles, i.e. the Am53C974 will return $\overline{\text{TRDY}}$ on the 10th bus cycle of the transfer. Figure 4-2 shows the timing for a Slave I/O Read bus cycle.

Figure 4-2 Slave I/O Read Timing

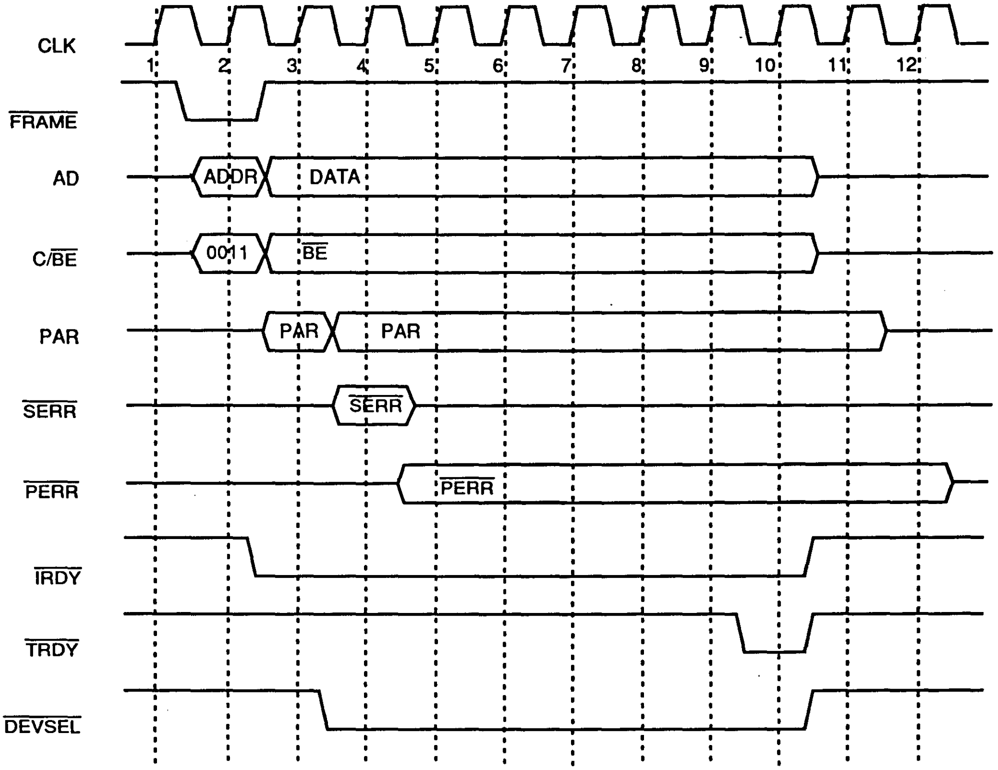


18264B-8

4.5.2 Slave I/O Write

The Slave I/O Write command is used by the processor to write the internal registers in the Am53C974. It is a single cycle, non-burst 8-Bit, 16-bit, or 32-bit transfer which is initiated by the host CPU. The Am53C974 will not produce Slave I/O Write commands while a bus master. Slave I/O Write cycles are fixed length cycles, i.e. the Am53C974 will return $\overline{\text{TRDY}}$ on the 10th bus cycle of the transfer. Figure 4-3 shows the timing for a Slave I/O Write bus cycle.

Figure 4-3 Slave I/O Write Timing



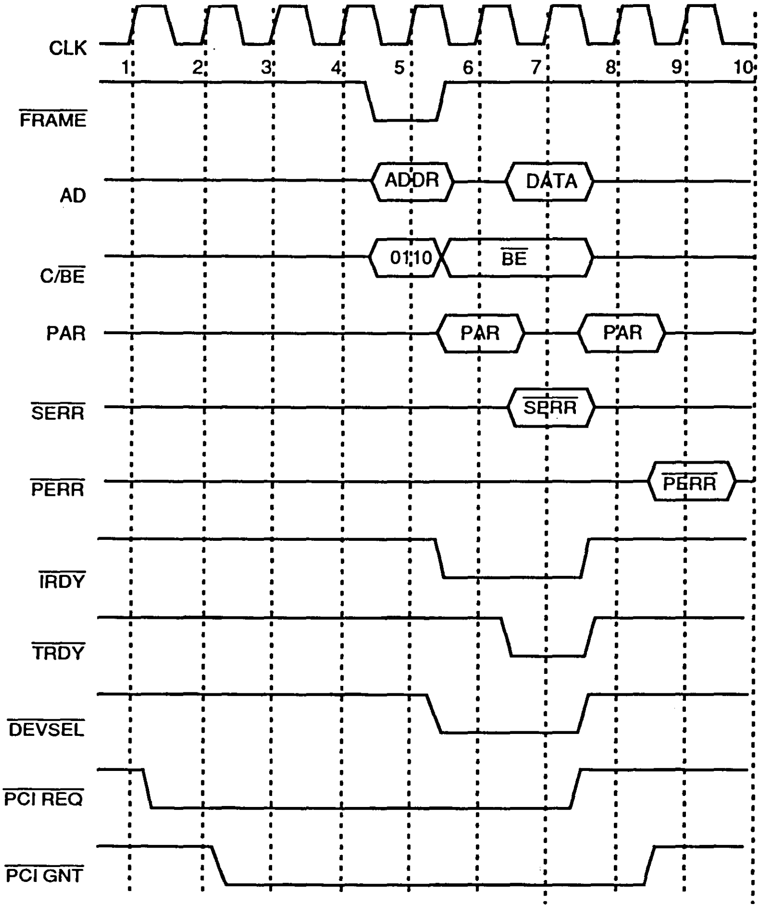
18264B-9

4.5.3 Master Memory Read

The Master Memory Read command is used by the Am53C974 when it will be reading 2 or less 32-bit locations of memory. If the device needs to read more than 2 Doublewords (Dwords) of memory, the Master Memory Read Line command (section 4.5.7) is used. Figure 4-4 shows an example timing diagram for a Master Memory Read command. In this figure, the device issues a request for the bus, is granted access, and then reads a 32-bit Dword from system memory before releasing the bus. The data phase in this diagram takes two clock cycles, this being determined by the timing of $\overline{\text{TRDY}}$.

Note that during a Master Memory Read, the Am53C974 will always activate all byte enables, even though some byte lanes may not contain "valid" data. In such instances, the Am53C974 will internally discard unnecessary bytes.

Figure 4-4 Master Memory Read Timing

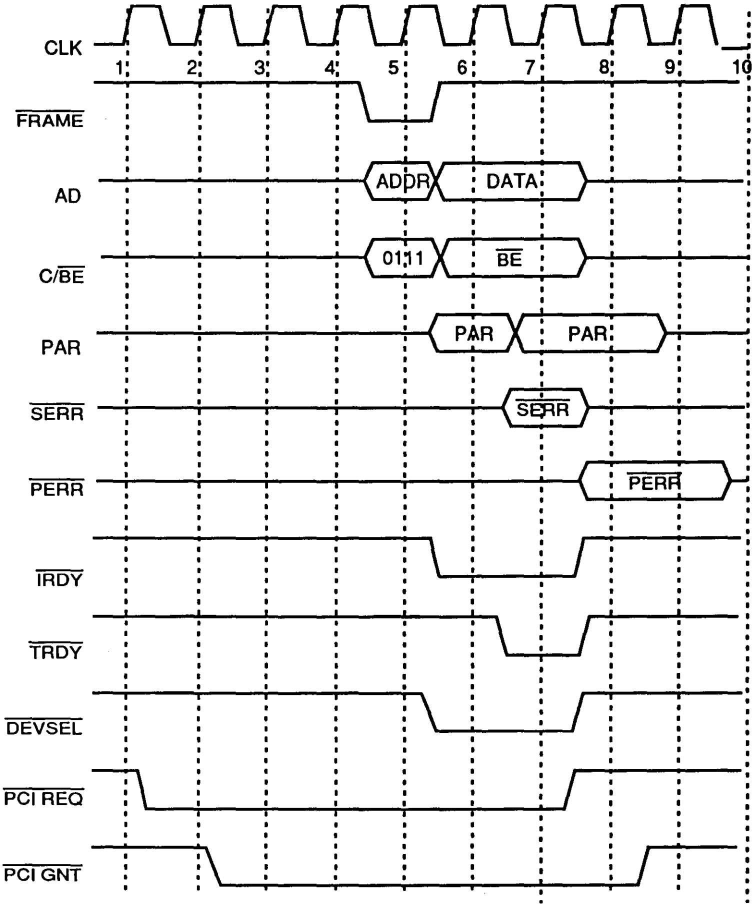


18264B-10

4.5.4 Master Memory Write

The Master Memory Write command is used by the Am53C974 when it will be writing to memory. Figure 4-5 shows an example timing diagram for a Master Memory Write command. In this figure, the device issues a request for the bus, is granted access, and then writes a 32-bit Dword into system memory before releasing the bus. Note that in this example, the data phase transfer takes 2 clock cycles. The timing of this transfer, in this case, was controlled by TRDY.

Figure 4-5 Master Memory Write Timing

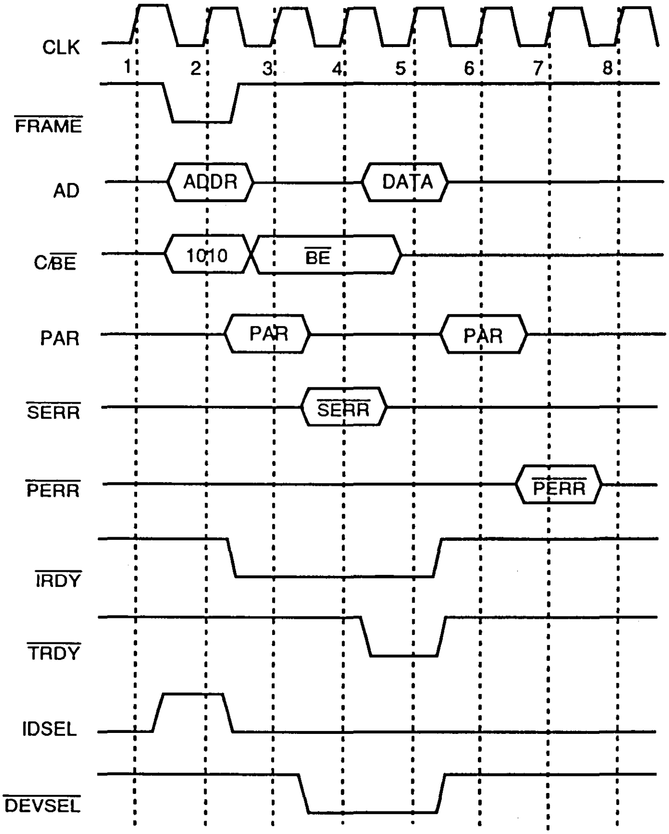


18264B-11

4.5.5 Slave Configuration Read

The Slave Configuration Read command is used by the host CPU to read the PCI configuration space in the Am53C974. This provides the host CPU with information concerning the device and its capabilities. This is a single cycle, non-burst 8-bit, 16-bit, or 32-bit transfer. Slave Configuration Read cycles are fixed length cycles, i.e. the Am53C974 will return TRDY on the 5th bus cycle of the transfer. Figure 4-6 shows the Configuration Read cycle timing.

Figure 4-6 Slave Configuration Read Timing

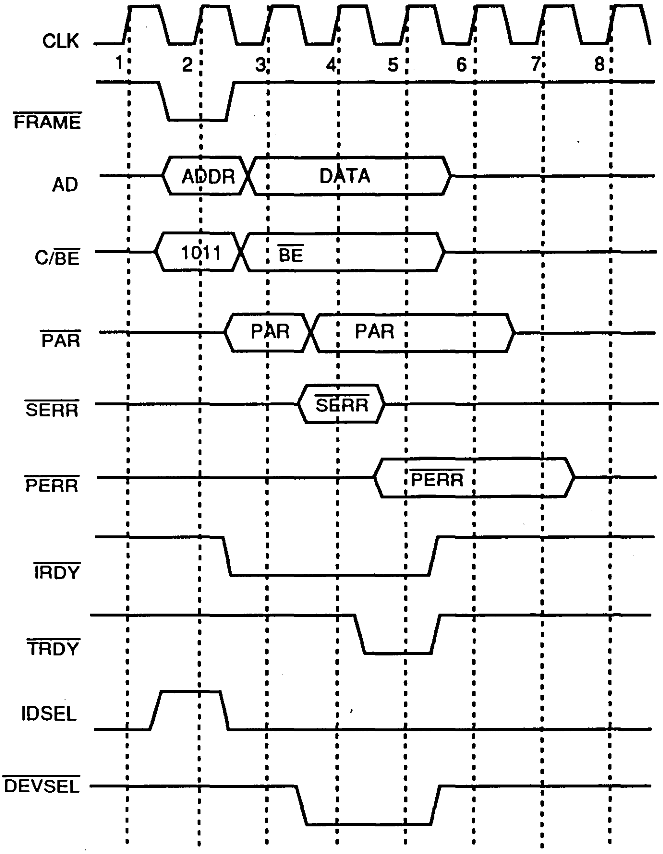


18264B-12

4.5.6 Slave Configuration Write

The Slave Configuration Write command is used by the host CPU to write the configuration space in the Am53C974. This allows the host CPU to control basic activity of the device, such as enable/disable, change I/O location, etc. This is a single cycle, non-burst 8-bit, 16-bit, or 32-bit transfer. Slave Configuration Write cycles are fixed length cycles, i.e. the Am53C974 will return TRDY on the 5th bus cycle of the transfer. Figure 4-7 shows the Configuration Write cycle timing.

Figure 4-7 Slave Configuration Write Timing

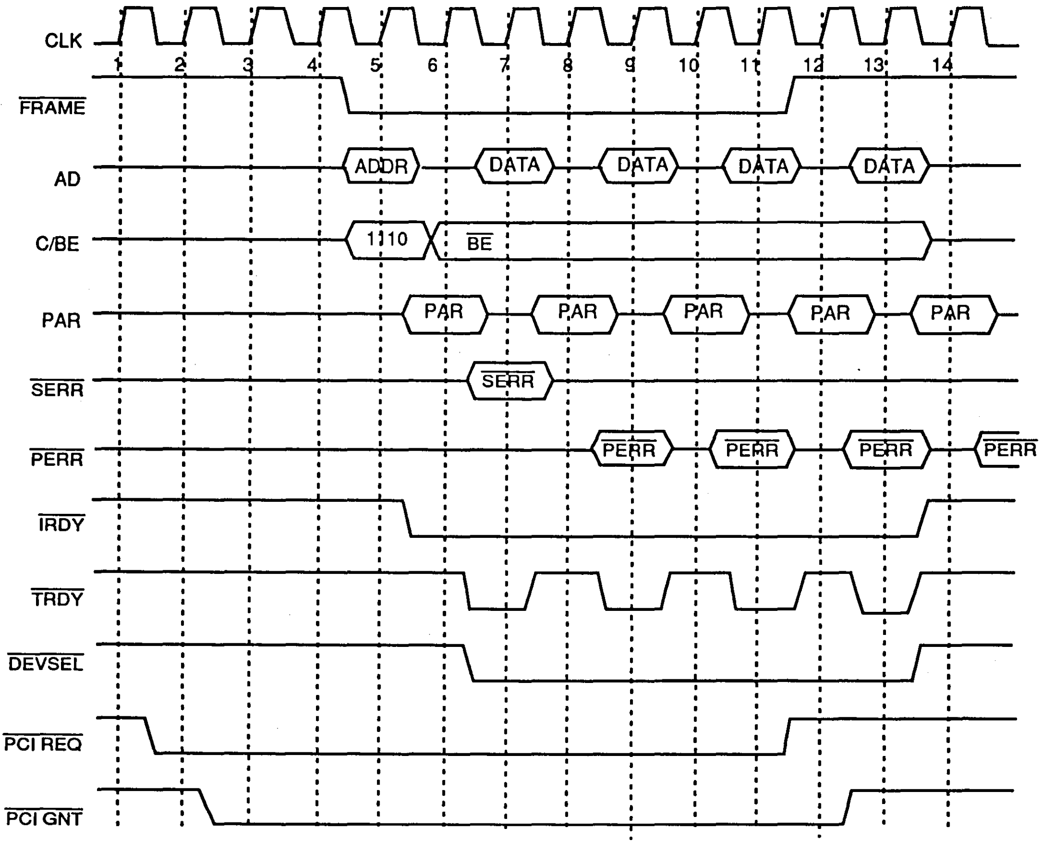


18264B-13

4.5.7 Master Memory Read Line

The Master Memory Read Line command is used by the Am53C974 when it will be reading more than two 32-bit locations of memory. If the device needs to read less than 2 Dwords of memory the Master Memory Read command (Section 4.5.3) is used. Figure 4-8 shows an example timing diagram for a Master Memory Read Line command. In this figure, the device issues a request for the bus, is granted access, and then reads four 32-bit Dwords from system memory before releasing the bus. Note that all data phases in this example take 2 clock cycles, this being determined by the timing of TRDY.

Figure 4-8 Master Memory Read Line Timing



18264B-14

4.6 TRANSACTION TERMINATION

Termination of a PCI transaction may be initiated by either the master or the target. During termination, the master remains in control to bring all PCI transactions to an orderly and systematic conclusion regardless of what caused the termination. All transactions are concluded when $\overline{\text{FRAME}}$ and $\overline{\text{IRDY}}$ are both deasserted, indicating an IDLE cycle.

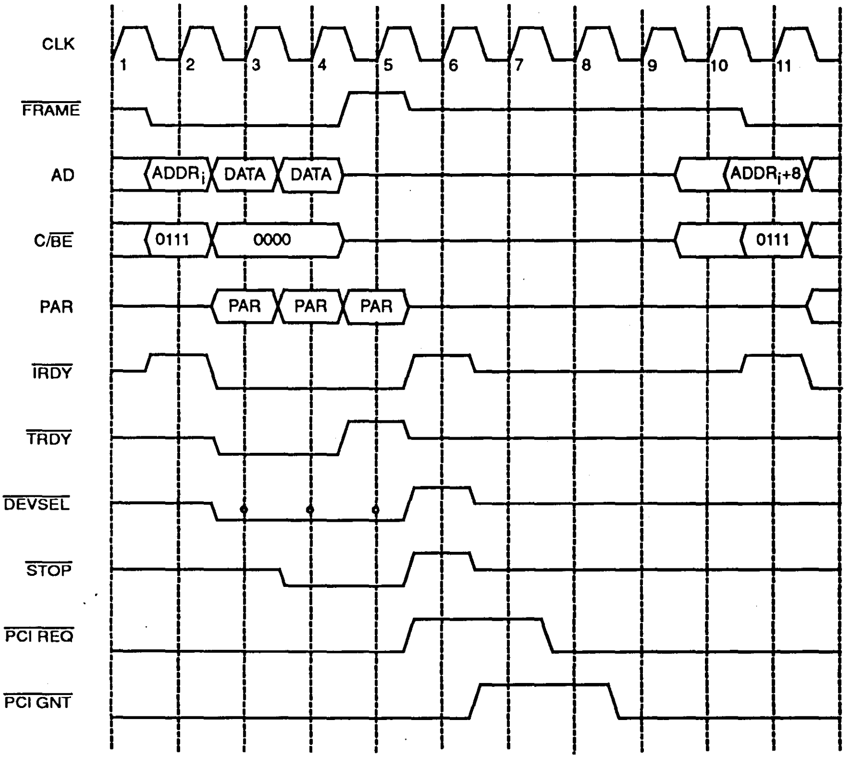
4.6.1 Target Initiated Termination

When the Am53C974 is a bus master, the cycles it produces on the PCI bus may be terminated by the target in one of three different ways: Disconnect with data transfer, disconnect without data transfer, and target abort.

4.6.1.1 Disconnect With Data Transfer

Figure 4-9 shows a disconnection in which one last data transfer occurs after the target asserted $\overline{\text{STOP}}$. $\overline{\text{STOP}}$ is asserted on clock 4 to start the termination sequence. Data is still transferred during this cycle, since both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted. The Am53C974 terminates the current transfer with the deassertion of $\overline{\text{FRAME}}$ on clock 5 and then one clock cycle later with the deassertion $\overline{\text{IRDY}}$. It finally releases the bus on clock 6. The Am53C974 will re-request the bus after 2 clock cycles, if it wants to transfer more data. The starting address of the new transfer will be the address of the next untransferred data.

Figure 4-9 Disconnect With Data Transfer



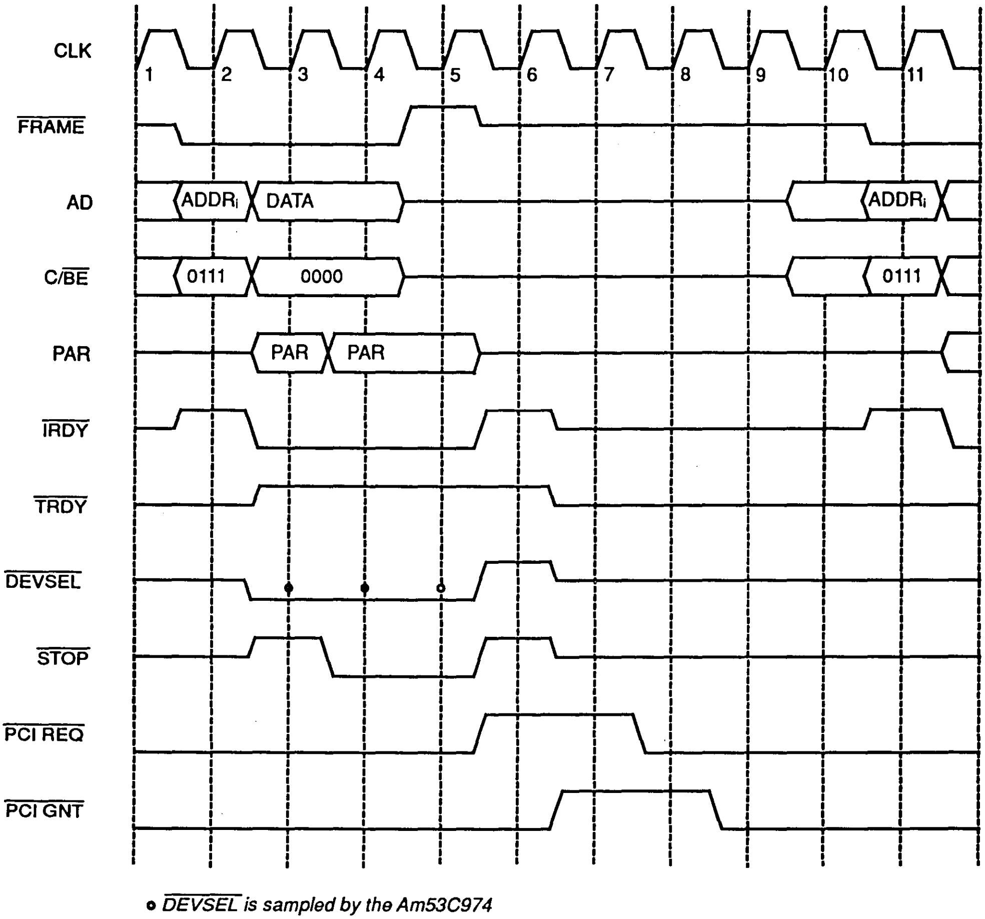
• *$\overline{\text{DEVSEL}}$ is sampled by the Am53C974*

18264B-15

4.6.1.2 Disconnect Without Data Transfer

Figure 4-10 shows a target disconnect sequence during which no data is transferred. $\overline{\text{STOP}}$ is asserted on clock 4 without $\overline{\text{TRDY}}$ being asserted at the same time. The Am53C974 terminates the current transfer with the deassertion of $\overline{\text{FRAME}}$ on clock 5 and one clock cycle later with the deassertion of $\overline{\text{IRDY}}$. It finally releases the bus on clock 6. The Am53C974 will re-request the bus after 2 clock cycles to retry the last transfer. The starting address of the new transfer will be the same address as the last untransferred data.

Figure 4-10 Disconnect Without Data Transfer

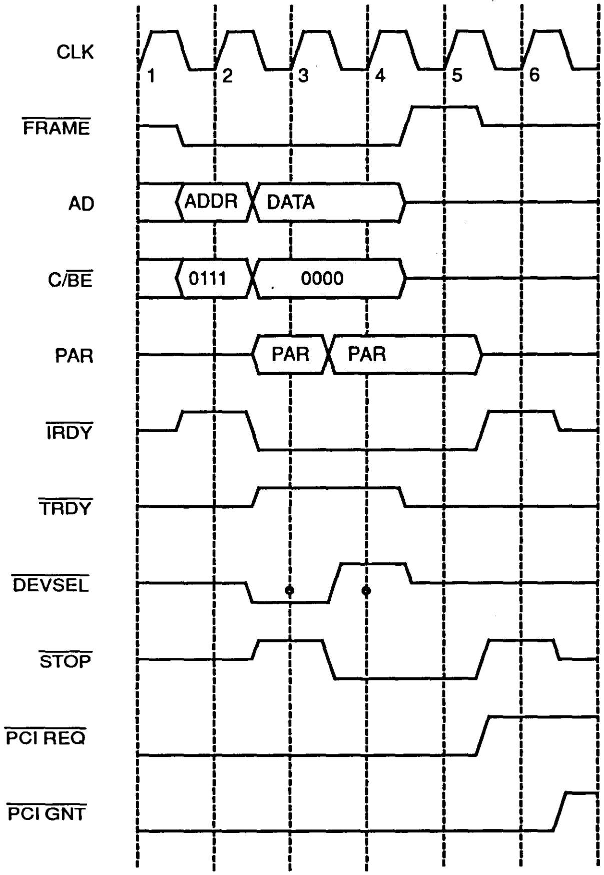


4.6.1.3 Target Abort

Figure 4-11 shows a target abort sequence. The target asserts $\overline{\text{DEVSEL}}$ for one clock. It then deasserts $\overline{\text{DEVSEL}}$ and asserts $\overline{\text{STOP}}$ on clock 4. A target can use the target abort sequence to indicate that it cannot service the data transfer and that it does not want the transaction to be retried. Additionally, the Am53C974 cannot make any assumption about the success of the previous data transfers in the current transaction. The Am53C974 terminates the current transfer with the deassertion of $\overline{\text{FRAME}}$ on clock 5 and one clock cycle later with the deassertion of $\overline{\text{IRDY}}$. It finally releases the bus on clock 6.

Since data integrity is not guaranteed, the Am53C974 cannot recover from a target abort event. Any on-going SCSI activity will be stopped immediately and an interrupt will be generated. The ABORT and ERROR bits will be set in the DMA Status register. The PCI configuration registers will not be cleared. RTABORT (bit 12) in the PCI Configuration Space Status register will be set to indicate that the Am53C974 has received a target abort.

Figure 4-11 Target Abort



o $\overline{\text{DEVSEL}}$ *is sampled by the Am53C974*

18264B-17

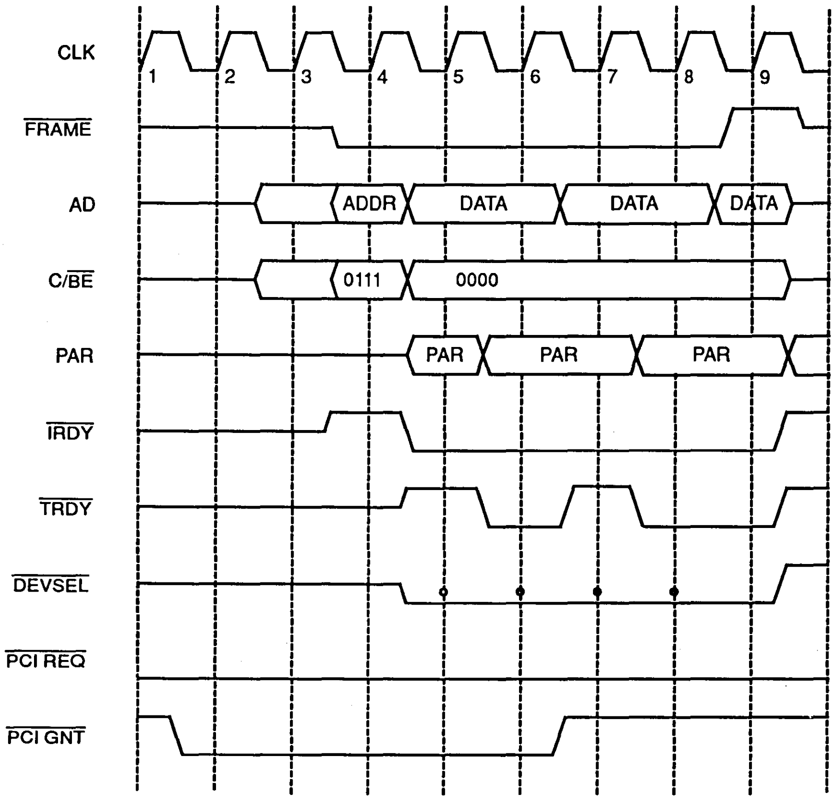
4.6.2 Master Initiated Termination

There are two scenarios besides normal completion of a transaction where the Am53C974 will terminate the cycles it produces on the PCI bus. These are Preemption and Master Abort.

4.6.2.1 Preemption

The central arbiter can take $\overline{\text{PCI GNT}}$ to the Am53C974 away if the current bus operation takes too long. This may happen during DMA bursts. When $\overline{\text{PCI GNT}}$ is taken away, the Am53C974 will finish the current transfer and then immediately release the bus. The Latency Timer in PCI configuration space of the Am53C974 is always set to ZERO. The Am53C974 will keep $\overline{\text{PCI REQ}}$ asserted to regain bus ownership as soon as possible.

Figure 4-12 Preemption



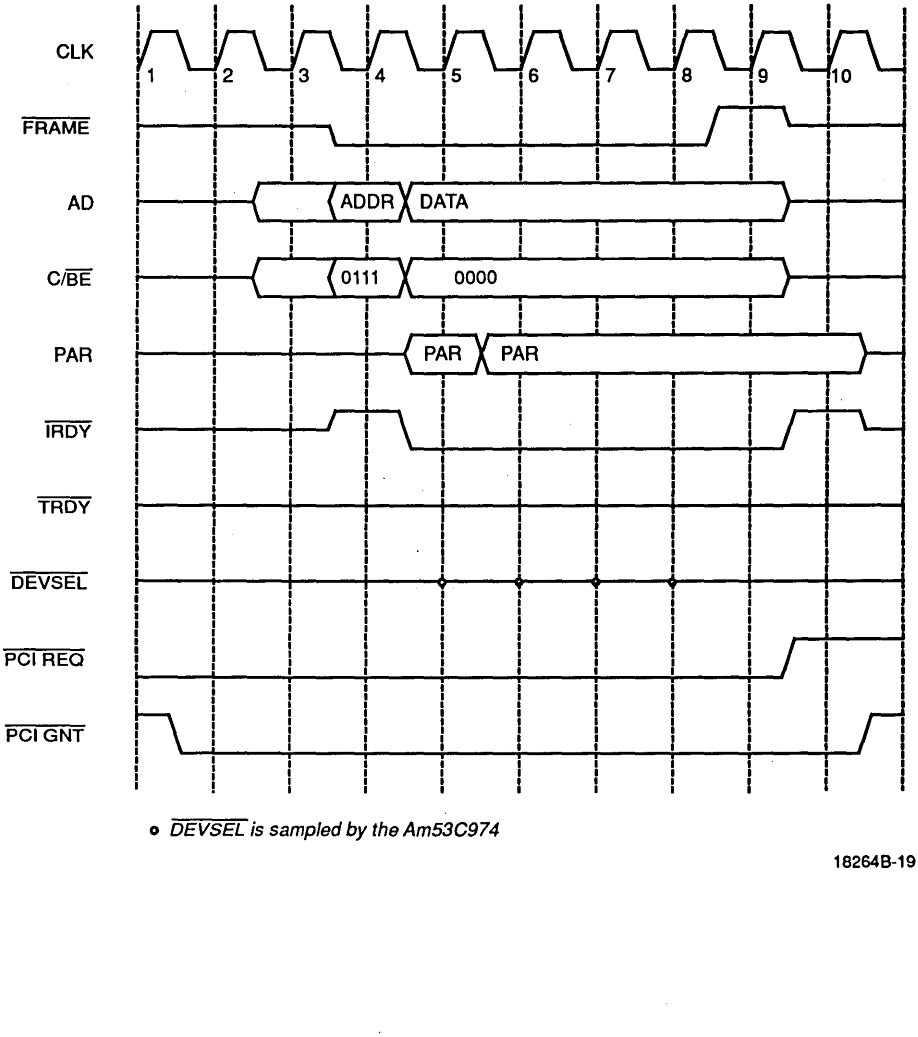
o *$\overline{\text{DEVSEL}}$ is sampled by the Am53C974*

18264B-18

4.6.2.2 Master Abort

The Am53C974 will terminate its cycle with a Master Abort sequence if DEVSEL is not asserted within 4 clocks after FRAME is asserted. Master Abort is treated as a fatal error by the Am53C974. Any on-going SCSI activity will be stopped immediately and an interrupt will be generated. The ABORT and ERROR bits will be set in the DMA Status Register. The PCI configuration registers will not be cleared. RMABORT (bit 13) in the PCI Configuration Space Status register will be set to indicate that the Am53C974 has terminated its transaction with a master abort.

Figure 4-13 Master Abort



○ *DEVSEL is sampled by the Am53C974*

18264B-19

4.7 CONFIGURATION REGISTERS

PCI Configuration Registers are used to determine which devices are in the system as well as to configure those devices. Configuration registers can be accessed any time but only by PCI configuration read/write cycles. This space is divided into two regions: A predefined header region and a device dependent region. The predefined header region contains 64 bytes organized as 4 Dwords while the device dependent region may contain up to 192 bytes also organized as 4 Dwords. The Am53C974 supports the full 64 byte predefined header and only 16 bytes of the device dependent region. Table 4-2 shows the configuration register map for both these regions. In Table 4-2, the first 64 bytes (00h – 3Fh) are the predefined header and the last 16 reserved bytes (40h – 4Fh) belong to the device dependent space.

Table 4-2 Configuration Register Map

31	16	15	0	Address Offset
Device ID		Vendor ID		00h
Status		Command		04h
Base Class	Sub Class	Prog. If.	Revision ID	08h
Reserved*	Header Type	Latency Timer	Reserved*	0Ch
Base Address				10h
Reserved*				14h
Reserved*				18h
Reserved*				1Ch
Reserved*				20h
Reserved*				24h
Reserved*				28h
Reserved*				2Ch
Reserved*				30h
Reserved*				34h
Reserved*				38h
Reserved*	Reserved*	Interrupt Pin	Interrupt Line	3Ch
Reserved for SCSI Software	Reserved for SCSI Software	Reserved for SCSI Software	Reserved for SCSI Software	40h**
Reserved for SCSI Software	Reserved for SCSI Software	Reserved for SCSI Software	Reserved for SCSI Software	44h**
Reserved for SCSI Software	Reserved for SCSI Software	Reserved for SCSI Software	Reserved for SCSI Software	48h**
Reserved for SCSI Software	Reserved for SCSI Software	Reserved for SCSI Software	Reserved for SCSI Software	4Ch**

** Not Implemented on Am53C974. Writes to these locations will have no effect; reads from these locations will return '00h'.*

*** Reserved for SCSI software.*

All PCI compliant devices, including the Am53C974, must support the *Vendor ID, Device ID, Command and Status register* in the header portion. Implementation of the other registers in this header is optional depending on device functionality. In Table 4-2, an asterisk (*) means the location is NOT implemented on the Am53C974 while a double asterisk (**) specifies that the location is reserved for use by the SCSI software. Write operations to unimplemented registers in the configuration space are treated as no-ops. That is, the access will be completed normally on the bus and the data discarded. Read accesses to unimplemented registers are completed normally and a data value of '00h' is returned.

4.7.1 Predefined Header Register Description

The following only describes the functions of the registers that are supported by the Am53C974. Refer to the *PCI Local Bus Specification* for more detailed information on PCI registers.

4.7.1.1 Vendor ID Register

Address 00h **READ ONLY**

This register identifies the manufacturer of this device as Advanced Micro Devices, Inc. (AMD) The Vendor ID is '1022h.'

4.7.1.2 Device ID Register

Address 02h **READ ONLY**

This register uniquely identifies this device within AMD's product line. The Am53C974 Device ID is '2020h.'

4.7.1.3 Command Register

Address 04h **READ/WRITE**

The Command Register is used to control the gross functionality of the device. It controls a device's ability to generate and respond to PCI bus cycles. To logically disconnect the Am53C974 from all PCI bus cycles except Configuration cycles, a value of zero should be written to this register. The Command Register is cleared by a PCI reset.

15	14	13	12	11	10	9	8
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	FBTBEN	SERREN
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
ADSTEP	PERREN	VGASNOOP	MWIEN	SCYCEN	BMEN	MEMEN	IOEN
1	0	0	0	0	0	0	0

Bit 15:10 – Reserved

These 6 bits are reserved by the PCI Specification. Write operations to these locations have no affect on the device. Read operations from these locations will return 0's.

Bit 9 – FBTBEN – Fast Back-to-Back Enable

This bit is hardwired to a value of '0' since Am53C974 back-to-back transactions are only allowed to the same agent.

Bit 8 – SERREN – SERR Enable

This read/write bit is an enable bit for the SERR driver. When this bit is set to '1', the SERR driver is enabled. When this bit is reset to '0', the SERR driver is disabled. This bit and bit 6 (Parity Error Response) must be set to '1' to report address parity errors. This bit's state is zero after a device reset.

Bit 7 – ADSTEP – Wait Cycle Control

This bit is hardwired to a value of '1' since the Am53C974 always does address stepping. The Am53C974 uses address stepping for the first address phase of each bus master period. FRAME will be asserted on the second clock following the assertion of PCI GNT, indicating a valid address on the AD bus.

Bit 6 – PERREN – Parity Error Response Enable

This read/write bit controls the Am53C974's response to parity errors. When PERREN is '0' and the Am53C974 detects a parity error, it only sets the Detected Parity Error bit in the PCI Configuration Space Status Register. When PERREN is '1', the Am53C974 asserts PERR on the detection of a data parity error. It also sets the DATAPERR bit (bit 8 in the PCI Configuration Space Status Register) when the data parity error occurred during a master cycle. PERREN also enables reporting address parity errors through the SERR pin and the SERR bit in the PCI Configuration Space Status Register. This bit must be reset to '0' after PCI RST. Parity is still generated by the device even if this bit is disabled (0).

Bit 5 – VGASNOOP – VGA Palette Snoop

This bit is hardwired to a value of '0' since the Am53C974 is not a VGA compatible device.

Bit 4 – MWIEN – Memory Write & Invalidate Enable

This bit is hardwired to a value of '0' since the Am53C974 does not generate memory write & invalidate commands. Instead, the memory write command must be used.

Bit 3 – SCYCEN – Special Cycles Enable

The Am53C974 will ignore all Special Cycle operations since this bit is hardwired to a value of '0'.

Bit 2 – BMEN – Bus Master Enable

This read/write bit controls the Am53C974's ability to act as a master on the PCI bus. When this bit is '0', the device is disabled from generating PCI accesses. When this bit is 1, the device is allowed to behave as a bus master.

Bit 1 – MEMEN – Memory Space Enable

This bit is hardwired to a value of '0' since the Am53C974 does not respond to memory space accesses.

Bit 0 – IOEN – I/O Space Enable

This read/write bit controls the Am53C974's response to I/O space accesses. When this bit is '0', the device will not respond to I/O space accesses. When this bit is '1', the device is allowed to respond to I/O space accesses. The host must set IOEN before the first I/O access to the device. The Base Address register at address (10H) must be programmed with a valid I/O address before setting IOEN.

4.7.1.4 Status Register

Address 06h **READ/WRITE**

The Status register is used to record status information for PCI bus related activities. Reads from this register function normally, however writes function differently. On a write of '1', bits will be reset (from 1 to 0), not set. For example, to reset bit 15 and not affect any other bits, a value of 1000_0000_0000_0000 should be written to the Status register.

15	14	13	12	11	10	9	8
PERR	SERR	RMABORT	RTABORT	STABORT	DEVSEL1	DEVSEL0	DATAPERR
X	X	X	X	X	0	1	X

7	6	5	4	3	2	1	0
FBBOK	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0	0	0	0	0	0	0	0

Bit 15 – PERR – Detected Parity Error

This bit is used by the Am53C974 to report parity errors. This bit is set to '1' whenever the device detects a parity error. This bit is cleared by writing a '1' to this location. The value of this bit is undefined after a device reset. The Am53C974 samples the AD(31:00), C/$\overline{\text{BE}}$(3:0) and the PAR lines for a parity error at the following times:

- In slave mode, during the address phase of any PCI bus command.
- In slave mode, during the data phase of all I/O and Configuration Write commands that select the Am53C974.
- In master mode, during the data phase of all Memory Read and Memory Read line commands.

During the data phase of the memory write command, the Am53C974 sets the PERR bit if the target reports a data parity error by asserting the $\overline{\text{PERR}}$ signal. PERR is not effected by the state of the Parity Error Response Enable bit (bit 6 in the PCI Configuration Space Command register).

Bit 14 – SERR – Signaled System Error

This bit is set to '1' when the $\overline{\text{SERR}}$ pin is asserted. This bit is cleared by writing a '1' to this location.

Bit 13 – RMABORT – Received Master Abort

As a bus master, the Am53C974 will set this bit to '1' whenever its transaction is terminated with a Master-abort. This bit is cleared by writing a '1' to this location.

Bit 12 – RTABORT – Received Target Abort

As a bus master, the Am53C974 will set this bit to '1' whenever its transaction is terminated with a target-abort. This bit is cleared by writing a '1' to this location.

Bit 11 – STABORT – Signaled Target Abort

As a target device, this bit is set to '1' by the Am53C974 whenever it terminates a transaction with a target-abort. This bit is cleared by writing a '1' to this location.

Bit 10:9 – DEVSEL1:0 – DEVSEL Timing

These bits encode the timing of the DEVSEL signal. These bits are hardwired to a value of '0' and '1' (for bits 10 and 9 respectively) since the Am53C974 uses medium assertion timing for the DEVSEL signal. That is, DEVSEL is asserted two clocks after FRAME is asserted. These bits are read-only and indicate the time that the Am53C974 asserts DEVSEL for any bus command.

Bit 8 – DATAPERR – Data Parity Detected

DATAPERR is set when the Am53C974 detects a data parity error during master mode and the Parity Error Response enable bit (bit 6 in the PCI Configuration Space Command register) is set.

During the data phase of all Memory Read and Memory Read Line commands, the Am53C974 checks for parity errors by sampling the AD(31:00), C/BE(3:0) and the PAR lines. During the data phase of all Memory Write commands, the Am53C974 checks the PERR input to detect whether the target has reported a parity error.

DATAPERR is set by the Am53C974 and is cleared by writing a '1' to this location. Writing a '0' has no effect.

Bit 7 – FBBOK – Fast Back-to-Back Capable

This bit is hardwired to a value of '0' since the Am53C974 is not capable of accepting fast back-to-back transactions when the transactions are not to the same agent.

Bit 6–0 – Reserved

These 7 bits are reserved by the PCI Specification. Write operations to these locations have no affect on the device. Read operations from these locations will return 0's.

4.7.1.5 Revision ID Register

Address 08h **READ ONLY**

This register specifies the device specific revision number. The current value of this register is '00h'.

4.7.1.6 Programming Interface Register

Address 09h **READ ONLY**

This register identifies the programming interface of this device. The value in this register is '00h'.

4.7.1.7 Sub-Class Register

Address 0Ah **READ ONLY**

This register identifies this device as a SCSI Controller as defined by the PCI specification. The value in this register is '00h'.

4.7.1.8 Base Class Register

Address 0Bh **READ ONLY**

This register identifies this device as a Mass Storage controller as defined by the PCI specification. The value in this register is '01h'.

4.7.1.9 Latency Timer Register

Address 0Dh **READ ONLY**

This register specifies the maximum time the Am53C974 can continue with bus master transfers after the system arbiter has removed $\overline{\text{PCI GNT}}$. The time is measured in CLK cycles. The working copy of the timer will start counting down when the Am53C974 asserts $\overline{\text{FRAME}}$ for the first time during a bus mastership period. The counter will freeze at ZERO. When the counter is ZERO and $\overline{\text{PCI GNT}}$ is deasserted by the system arbiter, the Am53C974 will finish the current data phase and then immediately release the bus.

The value for the Am53C974 Latency Timer Register is '00h', which indicates that when the Am53C974 is preempted, it will always release the bus promptly after finishing the current data phase.

4.7.1.10 Header TYPE Register

Address 0Eh **READ ONLY**

7	6	5	4	3	2	1	0
FUNCT	LAYOUT6	LAYOUT5	LAYOUT4	LAYOUT3	LAYOUT2	LAYOUT1	LAYOUT0
0	0	0	0	0	0	0	0

This is an 8-bit register that describes the format of the PCI Configuration Space locations 10h to 3Ch and that identifies a device to be single or multi-function. This register is located at address 0Eh in the PCI Configuration Space and is Read only. The value contained in this register is 00h.

4.7.1.11 Base Address Register

Address 10h **READ/WRITE**

This read/write register defines the Base Address for the Am53C974. Bit 0 is hardwired to a value of '1' to indicate that the base address is mapped into the I/O space while bit 1 is 'reserved' and will always return a '0' when read. That is, a value of '01' for bits 1 and 0 respectively will be returned on reads.

31	30	29	28	27	26	25	24
IOBASE26	IOBASE25	IOBASE24	IOBASE23	IOBASE22	IOBASE21	IOBASE20	IOBASE19
X	X	X	X	X	X	X	X

23	22	21	20	19	18	17	16
IOBASE18	IOBASE17	IOBASE16	IOBASE15	IOBASE14	IOBASE13	IOBASE12	IOBASE11
X	X	X	X	X	X	X	X

15	14	13	12	11	10	9	8
IOBASE10	IOBASE9	IOBASE8	IOBASE7	IOBASE6	IOBASE5	IOBASE4	IOBASE3
X	X	X	X	X	X	X	X

7	6	5	4	3	2	1	0
IOBASE2	IOBASE1	IOBASE0	IOSIZE2	IOSIZE1	IOSIZE0	Reserved	IOSPACE
X	0	0	0	0	0	0	1

Bit 31:5 – IOBASE 26:0 – I/O Base Address

These bits are written by the host to specify the location of the Am53C974 in all of I/O space. IOBASE 26:0 must be written with a valid address before the Am53C974 slave I/O mode is turned on with the setting of the IOEN bit (bit 0 in the PCI Configuration Space Command register).

When the Am53C974 is enabled for I/O mode (IOEN is set), it monitors the PCI bus for a valid I/O command. If the value on AD(31:05) during the address phase of the cycles matches the value of IOBASE, the Am53C974 will drive $\overline{\text{DEVSEL}}$ indicating it will respond to the access.

IOBASE 26:0 read and written by the host.

Bit 4:2 – IOSIZE 2:0 – I/O Size Requirements

IOSIZE 2:0 indicates the size of the I/O space the Am53C974 requires. When the host writes a value of FFFF_FFFF to the Base Address register, it will read back a value of '0' in bits 4–2. This indicates an Am53C974 I/O space requirement of 32 bytes.

Bit 1 – Reserved

This bit is reserved. Writes to this location have no effect. Reads from this location will return a '0'.

Bit 0 – IOSPACE – I/O Space Indicator

This bit indicates that the Base Address Register describes an I/O Base address. Writes to this location have no effect. Read from this location will return a '1'.

4.7.1.12 Interrupt Line Register

Address 3Ch **READ/WRITE**

The interrupt line register is used to communicate the routing of the interrupt. This read/write register is written by the POST (Power-On Self Test) software as it initializes the PCI devices in the system. The value in this register tells which input of the system interrupt controller(s) the Am53C974's interrupt pin is connected to. Device drivers and operating systems can use this information to determine priority and vector information. Values in the register are system architecture specific. For example, in x86 based PCs, the values in this register correspond to IRQ numbers (0–15) of the standard dual 8259 configuration. Values between 15 and 255 are reserved. Value 255 is defined as "unknown" or "no connection" to the interrupt controller.

4.7.1.13 Interrupt Pin Register

Address 3Dh **READ ONLY**

This register indicates which interrupt pin the device is using. This register is hard wired with a value of '1' because the Am53C974 only uses $\overline{\text{INTA}}$.

4.7.2 Device Dependent Register Description

The Am53C974 implements 16 bytes (4 Dwords located at locations 40h, 44h, 48h, and 4Ch) of the 192 byte device dependent registers. These 16 bytes are scratch data registers provided for use by SCSI device drivers. Developers of SCSI device drivers for the Am53C974 can use these register for their own needs provided that AMD's SCSI drivers are not used. If AMD SCSI drivers are used, these registers must not be modified. Note that since these are scratch registers, they may be used to contain any data. For example, AMD's SCSI drivers for the Am53C974 uses these registers to hold specific information concerning the state of the SCSI bus and each Target device connected. Examples of information contained in these registers as used by AMD's drivers are current SCSI bus status for each device, synchronous parameters, protected/real mode driver initialization flags, etc. The next section describes how AMD's SCSI device drivers take advantage of these registers.

4.7.3 AMD's Scratch Register Usage

The registers located at locations 40h, 44h, 48h, and 4Ch (Table 4–1) are four 32–bit registers (16 bytes total) which are used by AMD's SCSI device drivers to hold information about the state of the SCSI bus and the target devices connected. Table 4–3 illustrates how AMD's SCSI software defines and uses these registers. In Table 4–3, seven of the eight registers are used for target devices and one is used for the host. That is, there is one Target Configuration register for each SCSI Target and one Host Configuration register for the host.

Table 4-3 Scratch Register Definition for AMD's PCscsi Software

SCSI Configuration Register	PCI Configuration Byte	Bits 15:0
0	41h, 40h	SCSI Configuration Register 0
1	43h, 42h	SCSI Configuration Register 1
2	45h, 44h	SCSI Configuration Register 2
3	47h, 46h	SCSI Configuration Register 3
4	49h, 48h	SCSI Configuration Register 4
5	4Bh, 4Ah	SCSI Configuration Register 5
6	4Dh, 4Ch	SCSI Configuration Register 6
7	4Fh, 4Eh	SCSI Configuration Register 7

The following define the SCSI configuration registers for target devices and Host adapters for the Am53C974 SCSI software drivers.

4.7.3.1 Target Device Configuration Register Definition READ/WRITE

The Target Device Configuration Register layout is shown below. Bit placements follow "Little Endian" ordering.

15	14	13	12	11	10	9	8
Reserved	Reserved	Reserved	SPD4	SPD3	SPD2	SPD1	SPD0
0	0	0	X	X	X	X	X

7	6	5	4	3	2	1	0
SOFF3	SOFF2	SOFF1	SOFF0	STATUS2	STATUS1	STATUS0	PRES
X	X	X	X	X	X	X	X

Bits 15:13 – Reserved

These 3 bits are reserved by AMD's SCSI software driver.

Bits 12:8 – SPD4:0 – Synchronous Period

These bits define the synchronous period negotiated for the SCSI target device.

Bits 7–4 – SOFF3:0 – Synchronous Offset

These bits define the synchronous offset negotiated for the SCSI target device in number of bytes. A value of '0' indicates Asynchronous transfers. Valid values are from 1 to 15 (bytes).

Bits 3:1 – STATUS2:0 – SCSI Bus Status

These bits define the current state of the SCSI Target Device relating to the SCSI Bus. Valid values for the SCSI Bus Status are as follows:

Bits 3:1	SCSI Bus Status
000	Data Out Phase
001	Data In Phase
010	Command Phase
011	Status Phase
100	Idle
101	Active and Disconnected
110	Message Out Phase
111	Message In Phase

Bit 0 – PRES – Present

This bit is used to indicate that the target device is present and active. If this bit is set to '1', then the target device is present on the SCSI bus and all other bits are considered valid. If this bit is reset to '0', then the target device is assumed to be not present on the SCSI bus and all other bits must be reset to '0'. The exception to this case is if the Target Present bit is reset to '0' and the SCSI Bus Status bits are set to '1xx' (where 'x' is a don't care). In this case, the configuration register definition indicates the host adapter target ID.

4.7.3.2 Host Configuration Register Definition READ/WRITE

The Host Adapter Device Configuration Register layout is shown below. Bit placements follow "Little Endian" ordering.

15	14	13	12	11	10	9	8
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	RESET
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
SBNV	SBN2	SBN1	SBN0	HOST	PROTECT	RM	TP
0	0	0	0	1	X	X	0

Bits 15:9 – Reserved

These 7 bits are reserved by AMD's SCSI software driver.

Bit 8 – RESET – SCSI Bus Reset Has Taken Place

If this bit is set to '1', it indicates that a SCSI Bus Reset has occurred. This is useful for Protected Mode/Real Mode driver SCSI controller sharing. If device configuration parameters, such as Mode Select information or Synchronous Negotiation, have been issued to the SCSI devices, these parameters may no longer be valid upon the SCSI bus reset. When this bit is set, it indicates to the non–controlling driver that a Bus Reset has occurred and that appropriate action should be taken.

Bit 7 – SBNV – Starting BIOS Number Valid

When set to '1', this bit indicates that the Starting BIOS Number (SBN2:0, Bits 6–4) is valid. When reset to '0', the Starting BIOS Number is invalid.

Bit 6:4 – SBN2:0 – Starting BIOS Number

These bits are valid if the Starting BIOS Number Valid bit (bit 7) is set to '1'. This value ranges from 0 to 7 and indicates the starting BIOS unit number of the SCSI BIOS. For example, if the value is '1', this indicates that the SCSI BIOS starts controlling fixed disks at BIOS unit '81h'. If the value is '3', SCSI BIOS fixed disks start at BIOS unit '83h' and so on.

Bit 3 – HOST – Host

This bit is set to '1' to indicate that the device associated with this register is a host device.

Bit 2 – PROTECT – Protected Mode Driver Initialized

This bit is set to '1' when a Protected Mode device driver initializes the SCSI controller. A Real Mode driver that regains control due to a mode change (i.e. Windows to DOS or Netware 3.X to DOS, etc.) will reset this bit to '0' to indicate that once the Protected Mode driver regains control of the SCSI controller, it must re-initialize itself in order to continue proper operation. Upon re-initialization of the SCSI controller by the Protected Mode driver, this bit will once again be set to '1'.

Bit 1 – RM – Real Mode Driver Initialized

This bit is set to '1' when a Real Mode device driver initializes the SCSI controller. A Protected Mode driver that loads and initializes will reset this bit to '0' to indicate that if and when the Real Mode driver regains control of the SCSI controller, it must re-initialize itself in order to operate. Upon re-initialization of the SCSI controller by the Real Mode driver, this bit will once again be set to '1'.

Bit 0 – TP – Target Present Bit

This bit is set to '0' and is used in conjunction with Bit 3, which is set to '1', to indicate that this configuration register defines the Host configuration.

CHAPTER

5 THE FAST SCSI BLOCK

5.1 FUNCTIONAL OVERVIEW

The functionality of the SCSI block is described in the following section. Topics to be covered are:

- Part-unique ID
- SCSI FIFO Threshold
- Data Transmission
- $\overline{\text{REQ}}/\overline{\text{ACK}}$ Control
- Parity
- Reset Levels

5.1.1 Part-Unique ID

The Am53C974 contains a part-unique ID code which is stored in the MSB of the Current Transfer Count Register. The code reflects the chip's revision level and family code. This 8-bit code may be read when all the conditions are true.

- After power up or a chip reset has occurred
- Before the Current Transfer Counter ((B)+38h) is loaded

The part-unique ID code in Register ((B)+38h) will read as follows:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	1	0	0	1	0

5.1.2 SCSI FIFO Threshold

The threshold value for the SCSI FIFO is two bytes (one word). When this threshold is reached, the SCSI block will indicate to the DMA engine that it is capable of receiving or sending data bytes.

5.1.3 Data Transmission

Data transmission rates will vary from system to system, depending on the number of devices configured on the SCSI bus, as well as the transfer rates that each individual device is capable of sustaining.

Transfer rates for the Am53C974 are controlled by the FASTSCSI and FASTCLK bits in Control Register Three, as well by the Extended Timing Feature in Control Register One. The chart below show the effects of different bit configurations on minimum asynchronous and synchronous cycle times.

FASTSCSI Cntl Reg 3 Bit3	FASTSCSI Cntl Reg 3 Bit4	Ext. Timing Mode Cntl Reg 1 Bit 7	Min Cycles per Data Setup Asynch Xfer	Min Cycles per Data Setup Synch Xfer	Min Cycles per Period Synch Xfer
0	X	0	2	2	5
0	X	1	3	3	6
1	0	0	3	3	8
1	0	1	5	3	8
1	1	0	3	2	4
1	1	1	5	2	4

To achieve 10 MB/sec transmission rates, the following requirements must be true:

- A 40 MHz clock (50% duty cycle) must be supplied to SCSICLK1.
- The target must be able to sustain Fast SCSI timings
- Bits 3 and 4 in Control Register Three must be set to '1'
- The lower three bits of Register ((B)+24h), the Clock Conversion Factor Register must be programmed to '000'
- The lower 5 bits of the Synchronous Transfer Period Register ((B)+18h) must be set to a value of '04h.'

Control Register Three contains two bits which modify the SCSI state machine to produce both FAST and Normal SCSI timings. Synchronous data transmission rates are dependent on the input clock frequency selected, as well as the transfer period. The registers listed above should be programmed consistently.

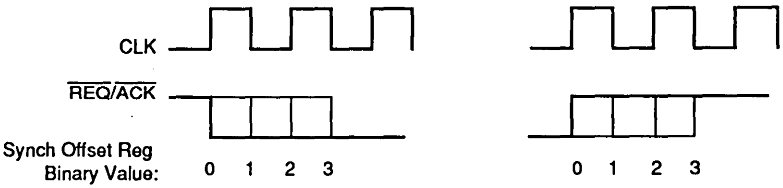
Bits 4:0 in the Synchronous Transfer Period Register ((B)+18h) specify the timing between the leading edges of consecutive $\overline{\text{REQ}}$ and $\overline{\text{ACK}}$ pulses during synchronous transfers. For programming information, refer to the register level descriptions.

5.1.4 $\overline{\text{REQ/ACK}}$ Control

The assertion and deassertion time for the $\overline{\text{REQ}}$ and $\overline{\text{ACK}}$ signals may be controlled via the Synchronous Offset Register ((B)+1Ch). Bits 7:6 control $\overline{\text{REQ/ACK}}$ deassertion delay, while bits 5:4 control $\overline{\text{REQ/ACK}}$ assertion delay. The deassertion for $\overline{\text{REQ/ACK}}$ may be moved ahead .5 clock cycles, or it may be delayed for up to 1.5 clock cycles. Deassertion delay options depend on the status of the FASTCLK bit in Control Register Three. Assertion delay for $\overline{\text{REQ/ACK}}$ can vary from 0 to 1.5 clock cycles.

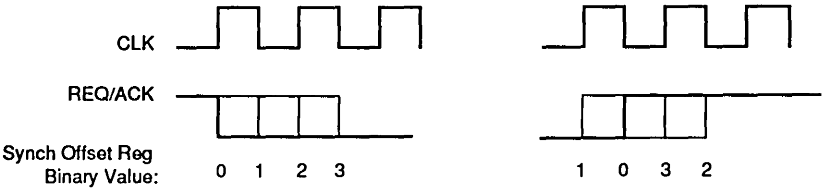
For programming information, refer to the register level descriptions. The following drawings illustrate the $\overline{\text{REQ/ACK}}$ assertion/deassertion feature:

FASTCLK Enabled



18264B-20

FASTCLK Disabled



18264B-21

Note: Care must be taken in programming this feature, as it is possible to violate SCSI-2 timing specifications.

5.1.5 Parity

Parity checking features are implemented via two bits in the Status Register and Control Register One Parity checking can be implemented on data flowing in from the SCSI bus. Parity is always generated internally by the Am53C974 for data moving onto the SCSI bus.

Feature	Bit Name	Bit #	Register
Parity from SCSI	Parity Error Reporting	4	Control Reg One ((B)+20h)
Parity Status	Parity Error	5	Status Register ((B)+10h)

5.1.5.1 Parity From the SCSI Bus

The Parity Error Reporting Bit (Bit 4, Control Register One) applies parity checking on all incoming bytes from the SCSI bus. This feature is cleared to '0' by a hardware reset.

When this feature is enabled, the Am53C974 will check parity on all data received from the SCSI bus. Any detected error will be flagged by setting bit 5 in the SCSI Status Register, and $\overline{\text{ATN}}$ will be asserted on the SCSI bus. However, no interrupt will be generated.

When this feature is disabled (bit 4 set to '0'), no parity check is done on incoming bytes; rather, the Am53C974 generates parity internally for each byte.

5.1.6 Reset Levels

The Am53C974 handles three different levels of resets, which are described below:

5.1.6.1 Hard Reset: (H)

This reset is initiated either at power-up, through the Reset Device command, or when the $\overline{\text{PCI RST}}$ pin is asserted externally through hardware. When this condition is true, all chip operations halt and functions are reset. The Am53C974 SCSI block is left in a Disconnected State.

When the Reset Device Command is executed, it will remain at the top of the SCSI Command Register, thereby holding the Am53C974 and its register set in a reset state until a NOP command is issued. The NOP command will release the Command Register as soon as it is loaded.

5.1.6.2 Soft Reset: (S)

This reset is initiated by the Reset SCSI bus command which asserts $\overline{\text{SCSIRST}}$, or by the SCSI bus reset conditions which is characterized by $\overline{\text{SCSIRST}}$ asserted.

The Reset SCSI bus command will drive $\overline{\text{SCSIRST}}$ active for 25 ms to 40 ms (depending on the input clock frequency and conversion factor), returning the Am53C974 to the Disconnected State. An interrupt is not generated upon completion, however, a SCSI reset interrupt will be generated if Bit 6 of Control Register One is enabled.

The SCSI bus Reset condition is initiated when the $\overline{\text{SCSIRST}}$ pin on the Am53C974 is asserted by another SCSI device on the bus. The chip returns to a Disconnected State, and a SCSI reset interrupt is generated to the microprocessor if bit 6 in Control Register One is enabled.

5.1.6.3 Disconnected Reset: (D)

This reset is caused by various commands or situations which cause the Am53C974 to disconnect from the SCSI bus.

- The Am53C974 is the Initiator and the SCSI bus moves to a Bus Free state
- The Selection command terminates due to selection time-out
- The functions reset during the Disconnected Reset are:
- All SCSI signals except $\overline{\text{SCSIRST}}$ are deasserted
- The SCSI Command Register is initialized to empty
- The Internal State bits are cleared (IS1:0 = '00')
- Disconnect and Initiator command modules are reset

The table below describes chip operations and features which are affected by the various reset levels.

Chip Operation/Feature	Reset Level
Deassert all SCSI signals except $\overline{\text{SCSIRST}}$*	
*$\overline{\text{SCSIRST}}$ cleared by Hard Reset only	HSD
Reset disconnect and initiator command modules	HSD
Command register FIFO initialized to empty	HSD
Reset Internal State Bits in Registers (B)+18h and (B)+1Ch	HS
Clear Internal State Register bits:	
Enable select (IS2 = '0')	HS
Initiator (IS 1:0 = '00')	HSD
Reset command sequence module	HS
Reset DMA Interface	HS
Reset bus-initiated selection module	HS
Clear SCSI Status Register ((B)+10h)	H
Clear Interrupt Status Register ((B)+14h)	H
Release $\overline{\text{INT}}$ pin	H
Deassert $\overline{\text{SCSIRST}}$ signal	H
Synchronous Offset = '0'	H
Synchronous Transfer Period Register = '5'	H
Initialize SCSI FIFO to empty condition	H
Clear all Control Registers	H
Set Clock Conversion Factor = '2'	H

H = Hard Reset

S = Soft Reset

D = Disconnected Reset

5.2 REGISTER DESCRIPTION

The Am53C974 SCSI registers are mapped to a double word address space as shown in the table below. However, the actual register data occupies only the least significant byte of the address. The register addresses are represented by the PCI Configuration Base Address (B) and its corresponding offset value. The Base Address for the Am53C974 is stored at register address (10h).

Register Acronym	Address (Hex.)	Register Description	Type
CTCREG	(B)+00	Current Transfer Count Register Low	Read
STCREG	(B)+00	Start Transfer Count Register Low	Write
CTCREG	(B)+04	Current Transfer Count Register Middle	Read
STCREG	(B)+04	Start Transfer Count Register Middle	Write
FFREG	(B)+08	SCSI FIFO Register	Read/Write
CMDREG	(B)+0C	SCSI Command Register	Read/Write
STATREG	(B)+10	SCSI Status Register	Read
SDIDREG	(B)+10	SCSI Destination ID Register	Write
INSTREG	(B)+14	Interrupt Status Register	Read
STIMREG	(B)+14	SCSI Timeout Register	Write
ISREG	(B)+18	Internal State Register	Read
STPREG	(B)+18	Synchronous Transfer Period Register	Write
CFIREG	(B)+1C	Current FIFO/Internal State Register	Read
SOFREG1	(B)+1C	Synchronous Offset Register	Write
CNTLREG1	(B)+20	Control Register One	Read/Write
CLKFREG	(B)+24	Clock Factor Register	Write
RES	(B)+28	Reserved	Write
CNTLREG2	(B)+2C	Control Register Two	Read/Write
CNTLREG3	(B)+30	Control Register Three	Read/Write
CNTLREG4	(B)+34	Control Register Four	Read/Write
CTCREG	(B)+38	Current Transfer Count Register High/Part-Unique ID Code	Read
STCREG	(B)+38	Start Current Transfer Count Register High	Write
RES	(B)+3C	Reserved	Write

5.2.1 Register Bit Map: Read

Register Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Xfer Cntr (B)+00h	XFER CNT (07)	XFER CNT (06)	XFER CNT (05)	XFER CNT (04)	XFER CNT (03)	XFER CNT (02)	XFER CNT (01)	XFER CNT (00)
Xfer Cntr (B)+04h	XFER CNT (15)	XFER CNT (14)	XFER CNT (13)	XFER CNT (12)	XFER CNT (11)	XFER CNT (10)	XFER CNT (09)	XFER CNT (08)
Xfer Cntr (B)+38h	XFER CNT (23)	XFER CNT (22)	XFER CNT (21)	XFER CNT (20)	XFER CNT (19)	XFER CNT (18)	XFER CNT (17)	XFER CNT (16)
FIFO (B)+08h	FIFO (07)	FIFO (06)	FIFO (05)	FIFO (04)	FIFO (03)	FIFO (02)	FIFO (01)	FIFO (00)
Command (B)+0Ch	DMA	CMD (06)	CMD (05)	CMD (04)	CMD (03)	CMD (02)	CMD (01)	CMD (00)
Status (B)+10h	INT	ILLEGAL OP	PARITY ERROR	CTZ	RES	PHASE MSG	PHASE C/D	PHASE I/O
Interrupt (B)+14h	SCSI RST	INVALID CMD	DISC	SVC REQ	SUCCESS OP	RESEL	RES	RES
Internal State (B)+18h	RES	RES	RES	RES	SYNCH OFFSET FLAG	INTERNAL STATE	INTERNAL STATE	INTERNAL STATE
Current FIFO/ Int State (B)+1Ch	INTERNAL STATE	INTERNAL STATE	INTERNAL STATE	CURRENT FIFO	CURRENT FIFO	CURRENT FIFO	CURRENT FIFO	CURRENT FIFO
Control 1 (B)+20h	XTEND TIMING	DISABLE INT	RES	PAR ERROR REPORT	RES	ID (02)	ID (01)	ID (00)
Clk Factor (B)+24h	RES	RES	RES	RES	RES	CLK FACTOR	CLK FACTOR	CLK FACTOR
Reserved (B)+28h	RES	RES	RES	RES	RES	RES	RES	RES
Control 2 (B)+2Ch	RES	ENABLE FEATURES	RES	RES	RES	RES	RES	RES
Control 3 (B)+30h	ADD'L ID CHK	RES	RES	FAST SCSI	FAST CLOCK	RES	RES	RES
Control 4 (B)+34h	GLITCH EATER	GLITCH EATER	POWER DOWN	RES	RES	ACTIVE NEG	RES	RES
Reserved (B)+3Ch	RES	RES	RES	RES	RES	RES	RES	RES

5.2.2 Register Bit Map: Write

Register Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Xfer Cntr (B)+00h	XFER CNT (07)	XFER CNT (06)	XFER CNT (05)	XFER CNT (04)	XFER CNT (03)	XFER CNT (02)	XFER CNT (01)	XFER CNT (00)
Xfer Cntr (B)+04h	XFER CNT (15)	XFER CNT (14)	XFER CNT (13)	XFER CNT (12)	XFER CNT (11)	XFER CNT (10)	XFER CNT (09)	XFER CNT (08)
Xfer Cntr (B)+38h	XFER CNT (23)	XFER CNT (22)	XFER CNT (21)	XFER CNT (20)	XFER CNT (19)	XFER CNT (18)	XFER CNT (17)	XFER CNT (16)
FIFO (B)+08h	FIFO (07)	FIFO (06)	FIFO (05)	FIFO (04)	FIFO (03)	FIFO (02)	FIFO (01)	FIFO (00)
Command (B)+0Ch	DMA	CMD (06)	CMD (05)	CMD (04)	CMD (03)	CMD (02)	CMD (01)	CMD (00)
Destination ID (B)+10h	RES	RES	RES	RES	RES	DEST ID	DEST ID	DEST ID
Time Out (B)+14h	TIME (07)	TIME (06)	TIME (05)	TIME (04)	TIME (03)	TIME (02)	TIME (01)	TIME (00)
Synch Xfer Pd (B)+18h	RES	RES	RES	SYNCH PERIOD	SYNCH PERIOD	SYNCH PERIOD	SYNCH PERIOD	SYNCH PERIOD
Synch Offset (B)+1Ch	$\overline{\text{REQ/ACK}}$ DEASSERT	$\overline{\text{REQ/ACK}}$ DEASSERT	$\overline{\text{REQ/ACK}}$ ASSERT	$\overline{\text{REQ/ACK}}$ ASSERT	SYNCH OFFSET	SYNCH OFFSET	SYNCH OFFSET	SYNCH OFFSET
Control 1 (B)+20h	XTEND TIMING	DISABLE INT	RES	PAR ERROR REPORT	RES	ID (02)	ID (01)	ID (00)
Clk Factor (B)+24h	RES	RES	RES	RES	RES	CLK FACTOR	CLK FACTOR	CLK FACTOR
Reserved (B)+28h	RES	RES	RES	RES	RES	RES	RES	RES
Control 2 (B)+2Ch	RES	ENABLE FEATURES	RES	RES	RES	RES	RES	RES
Control 3 (B)+30h	ADD'L ID CHK	RES	RES	FAST SCSI	FAST CLOCK	RES	RES	RES
Control 4 (B)+34h	GLITCH EATER	GLITCH EATER	POWER DOWN	RES	ACTIVE NEG	ACTIVE NEG	RES	RES
Reserved (B)+3Ch	RES	RES	RES	RES	RES	RES	RES	RES

5.2.3 Register Descriptions

The values shown below each bit reflect register reset values. The register shall default to these values following a power-up or chip reset. Bit level descriptions are valid for the LSB at each address location. Remaining bytes at each address location are 'reserved.'

5.2.3.1 Current Transfer Count Register (CTCREG)
Address [(B)+00h, (B)+04h, (B)+38h] READ ONLY

Address (B)+38h

7	6	5	4	3	2	1	0
CRVL23	CRVL22	CRVL21	CRVL20	CRLV19	CRVL18	CRVL17	CRVL16
X	X	X	X	X	X	X	X

Address (B)+04h

7	6	5	4	3	2	1	0
CRVL15	CRVL14	CRVL13	CRVL12	CRLV11	CRVL10	CRVL9	CRVL8
X	X	X	X	X	X	X	X

Address (B)+00h

7	6	5	4	3	2	1	0
CRVL7	CRVL6	CRVL5	CRVL4	CRLV3	CRVL2	CRVL1	CRVL0
X	X	X	X	X	X	X	X

Bit 23:0 – CRVL 23:0 – Current Value

This is a three-byte register which decrements to keep track of the number of bytes transferred during a DMA transfer. Reading these registers returns the current value of the counter. The counter will decrement by one for every byte and by two for every word transferred. The transaction is complete when the count reaches zero, and bit 4 of the SCSI Status Register ((B)+10h) is set. Should the sequence terminate early, the sum of the values in the Current FIFO ((B)+1Ch) and the Current Transfer Count Register reflect the number of bytes remaining.

The least significant byte is located at address ((B)+00h), the middle byte is located at address ((B)+04h), and the most significant byte is located at address ((B)+38h). Register ((B)+38h) extends the total width of the register from 16 to 24 bits, and is only enabled when the Enable Features bit (bit 6) of Control Register Two is set to a value of '1'.

These registers are automatically loaded with the values in the Start Transfer Count Register every time a DMA command is issued. However, following a chip or power on reset, up until the time register ((B)+38h) is loaded, the Am53C974s part-unique ID can be obtained by reading register ((B)+38h).

The value in the Current Transfer Count Register will be decremented as follows:

Asynch Data In:	active edge of $\overline{\text{ACK}}$
Synch Data In:	active edge of $\overline{\text{DACK}}$
Data Out:	active edge of $\overline{\text{DACK}}$

5.2.3.2 Start Transfer Count Register (STCREG)

Address [(B)+00h, (B)+04h, (B)+38h] **WRITE**

Address (B)+38h

7	6	5	4	3	2	1	0
STVL23	STVL22	STVL21	STVL20	STVL19	STVL18	STVL17	STVL16
X	X	X	X	X	X	X	X

Address (B)+04h

7	6	5	4	3	2	1	0
STVL15	STVL14	STVL13	STVL12	STVL11	STVL10	STVL9	STVL8
X	X	X	X	X	X	X	X

Address (B)+00h

7	6	5	4	3	2	1	0
STVL7	STVL6	STVL5	STVL4	STVL3	STVL2	STVL1	STVL0
X	X	X	X	X	X	X	X

Bit 23:0 – STVL 23:0 – Start Value

This is a three-byte register which contains the number of bytes to be transferred during a DMA operation. The value in the Start Transfer Count Register must be programmed prior to command execution. The value programmed in this register should be the same as the value programmed in the DMA Starting Transfer Counter ((B)+44h).

The least significant byte is located at address ((B)+00h), the middle byte is located at address ((B)+04h), and the most significant byte is located at address ((B)+38h). Register ((B)+38h) extends the total width of the register from 16 to 24 bits, and is only enabled when the Enable Features bit (bit 6) of Control Register Two is set to a value of '1'. This sets the maximum transfer count to 16 MBytes. When a value of '0' is written to these registers, the transfer count will be set to the maximum.

These registers retain their value until overwritten, and are therefore unaffected by a hardware or software reset. This reduces programming redundancy since it is no longer necessary to reprogram the count for subsequent DMA transfers of the same size.

5.2.3.3 SCSI FIFO Register (FFREG)

Address (B)+08h **READ/WRITE**

7	6	5	4	3	2	1	0
FF7	FF6	FF5	FF4	FF3	FF2	FF1	FF0
0	0	0	0	0	0	0	0

Bit 7:0 – FF 7:0 – FIFO

The FIFO on the Am53C974 is 16 bytes deep and is used to transfer SCSI data to and from the Am53C974. The bottom of the FIFO may be accessed via a read or write to this register. This is the only register that can be accessed with $\overline{\text{REQ}}$ or $\overline{\text{ACK}}$. This register is reset to zero by hardware or software reset or if the Clear FIFO command is issued.

5.2.3.4 SCSI Command Register (CMDREG)

Address (B)+0Ch **READ/WRITE**

7	6	5	4	3	2	1	0
DMA	CMD6	CMD5	CMD4	CMD3	CMD2	CMD1	CMD0
X	X	X	X	X	X	X	X

Commands to the Am53C974 are issued by writing to this register which is two bytes deep. Commands may be queued, and will be read from the bottom of the queue. At the completion of the bottom command, the top command, if present, will drop to the bottom of the register to begin execution. All commands are executed within six clock cycles of dropping to the bottom of the SCSI Command Register, with the exception of the Reset SCSI Bus, Reset Device, and DMA Stop commands. These commands are not queued and are executed within four clock cycles of being loaded into the top this register.

Interrupts are generated upon completion of some commands. Should back-to-back commands generate interrupts, and the first interrupt has not been serviced, the interrupt from the second (top) command will be stacked behind the first. The SCSI Status Register ((B)+10h), Interrupt Register ((B)+14h), and Internal State Register ((B)+18h) will be updated to reflect the second interrupt after the microprocessor services the first interrupt.

Reading this register will return the command currently being executed (or the last command executed if there are no pending commands). When this register is cleared, existing commands will be terminated and any queued commands will be ignored. However, clearing this register does not reset the bits to '00h'.

Under the following conditions, the SCSI Command Register will be cleared and maintained in a reset state (00h), until the host services the Interrupt Status Register ((B)+18h).

- Illegal Command
- SCSI Bus reset or disconnect
- Completion of

  Bus-initiated Selection or Reselection
  Select command
- Selection or reselection timeout
- Not in Message In phase for the second byte of the Initiator Command Complete Steps
- Unexpected phase change during an Information Transfer or Transfer Pad Bytes command

Bit 7 – DMA – Direct Memory Access

When set, this bit notifies the device that the command is a DMA instruction, when reset it is a non-DMA instruction.

For DMA instructions the Current Transfer Count Register (CTCREG) will be loaded with the contents of the Start Transfer Count Register (STCREG). The data is then transferred and the CTCREG is decremented for each byte until it reaches zero. Data is transferred between system memory and the SCSI bus via the bus-mastering DMA engine.

Non-DMA instructions do not modify the Transfer Count Registers ((B)+00h, (B)+04h, and (B)+38h), since the number of bytes transferred is a function of the operation rather than the transfer count. These type of instructions move data between the SCSI FIFO and the SCSI bus, and requires host processor intervention to handle data transmission between the SCSI FIFO and memory.

CMDREG – Bit 6:0 – CMD 6:0 – Command

These command bits decode the commands that the device needs to perform. There are a total of 16 commands grouped into three categories: Initiator Commands, Idle State Commands and General Commands.

5.2.3.5 SCSI Status Register (STATREG)

Address (B)+10h **READ ONLY**

7	6	5	4	3	2	1	0
INT	IOE	PE	CTZ	Reserved	MSG	C/D	I/O
0	0	0	0	X	X	X	X

This read-only register contains flags which indicate the status of the chip and the current phase of the SCSI bus. These bits are read in conjunction with the Interrupt Status Register ((B)+14h) to qualify the reason for the interrupt. This register should always be read prior to servicing the Interrupt Status Register (INSTREG) since bits 7:3 will be reset to '0' once the Interrupt Status Register is read. If command stacking is used, the phase bits may be latched by setting the ENF bit (Control Register Two, bit 6). With this feature enabled, the SCSI bus phase of the last complete command (preceding the interrupt) will be latched by bits 2:0.

Bits 7:3 are reset to '0' during a hardware reset.

Bit 7 – INT – Interrupt

The INT bit is set when the SCSI block detects an interrupt condition. This bit will be cleared by a hardware or software reset. Reading the Interrupt Status Register ((B)+14h) will deassert the interrupt output and also clear this bit.

Note: SCSI interrupt conditions will also be flagged in the DMA Status Register ((B)+54h, Bit 4).

Bit 6 – IOE – Illegal Operation Error

The IOE bit is set when an illegal operation is attempted. This condition will not cause an interrupt, and will therefore be detected by reading the Status Register ((B)+10h) while servicing another interrupt. The following conditions will cause the IOE bit to be set:

- DMA and SCSI transfer directions are opposite.
- FIFO overflows or data is overwritten.
- In Initiator mode and unexpected phase change detected during synchronous data transfer.
- Command Register overwritten.

This bit is cleared by reading the Interrupt Status Register ((B)+14h) or by a hard or soft reset.

Bit 5 – PE – Parity Error

The PE bit is set if any of the parity checking options are enabled and the device detects a parity error on bytes sent or received on the SCSI Bus. Parity options are controlled by bit 4 in Control Register One ((B)+20h), and by bit 2 in Control Register Two ((B)+2Ch). Detection of a parity error condition will not cause an interrupt but will be reported with other interrupt causing conditions.

This bit will be cleared by reading the Interrupt Status Register ((B)+14h) or by a hard or soft reset.

Bit 4 – CTZ – Count To Zero

The CTZ bit is set when the Current Transfer Count Register ((B)+00h, (B)+04h, (B)+38h) has decremented to zero. This bit is reset when the Current Transfer Count Register is re-loaded. Reading the Interrupt Status Register ((B)+14h) will not affect this bit. This bit will however be cleared by a hard or soft reset.

Note: A non-DMA NOP will not reset the CTZ bit since it does not load the Current Transfer Count Register. However, a DMA NOP will reset this bit since it loads the Current Transfer Count Register.

Bit 3 – Reserved

This bit is reserved.

Bit 2 – MSG – Message
Bit 1 – C/D – Command/Data
Bit 0 – I/O – Input/Output

The MSG, C/D and I/O bits together are referred to as the SCSI Phase bits. They indicate the phase of the SCSI bus. These bits may be latched or unlatched depending on whether or not the ENF bit in Control Register Two is set. In the latched mode the SCSI phase bits are latched at the end of a command and the latch is opened when the Interrupt Status Register ((B)+14h) is read. In the unlatched mode, they indicate the real-time phase of the SCSI bus.

Bit 2 MSG	Bit 1 C/D	Bit 0 I/O	SCSI Phase
1	1	1	Message In
1	1	0	Message Out
1	0	1	Reserved
1	0	0	Reserved
0	1	1	Status
0	1	0	Command
0	0	1	Data In
0	0	0	Data Out

5.2.3.6 SCSI Destination ID Register (SDIDREG)

Address (B)+10h **WRITE**

7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	Reserved	Reserved	DID2	DID1	DID0
0	0	0	0	0	X	X	X

Bit 7:3 – Reserved

Bit 2:0 – DID 2:0 – Destination ID

The DID 2:0 bits are the encoded SCSI ID of the device on the SCSI bus which needs to be selected or reselected. At power-up the state of these bits is undefined. The DID 2:0 bits are not affected by reset.

DID2	DID1	DID0	SCSI ID
1	1	1	7
1	1	0	6
1	0	1	5
1	0	0	4
0	1	1	3
0	1	0	2
0	0	1	1
0	0	0	0

5.2.3.7 Interrupt Status Register (INSTREG)

Address (B)+14h **READ ONLY**

7	6	5	4	3	2	1	0
SRST	ICMD	DIS	SR	SO	RESEL	Reserved	Reserved
0	0	0	0	0	0	X	X

The Interrupt Status Register (INSTREG) indicates the reason for the interrupt. This register is used with the SCSI Status Register ((B)+10h) and Internal State Register ((B)+18h) to determine the reason for the interrupt. Reading the Interrupt Status Register will clear all three registers. Therefore the SCSI Status Register ((B)+10h) and Internal State Register ((B)+18h) should be examined prior to reading this register.

This register should only be read when an interrupt is pending. All bits will be cleared to '0' by a hardware reset.

Bit 7 – SRST – SCSI Reset

The SRST bit will be set if a SCSI Reset is detected and SCSI reset reporting is enabled via the DISR (bit 6) of Control Register One ((B)+20h).

Bit 6 – ICMD – Invalid Command

The ICMD bit will be set if the device detects an illegal command code. This bit is also set if a command code is detected from a mode that is different from the mode the device is currently in. Once set, an invalid command interrupt will be generated.

Bit 5 – DIS – Disconnected

The DIS bit can be set when the device disconnects from the SCSI bus. In the Initiator mode this bit will be set if the Target disconnects; while in Idle mode, this bit will be set if a Selection or Reselection timeout occurs.

Bit 4 – SR – Service Request

The SR bit can be set when another device on the SCSI bus has a service request. In the Initiator mode, this bit is set when the Target requests an information transfer phase.

Bit 3 – SO – Successful Operation

The SO bit can be set when an operation has successfully completed. In the Initiator mode this bit is set after a Target has been successfully selected, after a command has successfully completed and after an Information Transfer Command when the Target requests a Message In phase.

Bit 2 – RESEL – Reselected

The RESEL bit is set at the end of the Reselection phase indicating that the device has been reselected as an Initiator.

Bit 1 – Reserved

This bit is reserved.

Bit 0 – Reserved

This bit is reserved.

5.2.3.8 SCSI Timeout Register (STIMREG)

Address (B)+14h **WRITE**

7	6	5	4	3	2	1	0
STIM7	STIM6	STIM5	STIM4	STIM3	STIM2	STIM1	STIM0
X	X	X	X	X	X	X	X

This register determines how long the Initiator will wait for a response to a Selection before timing out. It should be set to yield 250 ms to comply with ANSI standards for SCSI. The maximum time out period may be calculated using the following formulas. A hardware reset will clear this register.

Bit 7:0 – STIM 7:0 – SCSI Timer

The value loaded in STIM 7:0 can be calculated as shown below:

STIM 7:0 = [(SCSI Time Out) (Clock Frequency) / (8192 (Clock Factor))]

Example:

SCSI Time Out (in seconds): 250 ms.

(Recommended by the ANSI Standard) = 250 x 10–3 s.

Clock Frequency: 40 MHz. (assume) = 40 x 106 Hz.

Clock Factor: CLKF 2:0 from Clock Conversion Register ((B)+24h) = '000'

STIM 7:0 = (250 x 10–3) X (40 x 106) / (8192 (5)) = 244.14 decimal

The decimal value of **244.14 must be rounded up to 245** (the next integer value), and its hexadecimal value of 'F5h' should be written to this register.

5.2.3.9 Internal State Register (ISREG)

Address (B)+18h **READ ONLY**

7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	Reserved	$\overline{\text{SOF}}$	IS2	IS1	IS0
X	X	X	X	0	0	0	0

The lower four bits of this register track the progress of a sequence-type command. It is updated after each successful completion of an intermediate operation. If an error occurs, the host can read this register to determine the point where the command failed and take the necessary procedure for recovery. Reading the Interrupt Status Register ((B)+14h) while an interrupt is pending will clear this register. A hard or soft reset will also clear this register.

Bit 7:4 - Reserved

Bit 3 - SOF - Synchronous Offset Flag

The SOF is reset when the Synchronous Offset Register ((B)+1Ch) has reached its maximum value of 15.

Note: The SOF bit is active LOW.

Bit 2:0 - IS 2:0 - Internal State

The IS 2:0 bits along with the Interrupt Status Register ((B)+14h) indicates the status of the successfully completed intermediate operation. Refer to the following Status Decode section for more details.

Status Decode:

Initiator Select without $\overline{\text{ATN}}$ Steps

Internal State Register ((B)+18h) Bits 2:0 (Hex)	Interrupt Status Register ((B)+14h) Bits 7:0 (Hex)	Explanation
0	20	Arbitration steps completed. Selection time-out occurred, then disconnected
4	18	Selection without $\overline{\text{ATN}}$ steps fully executed
3	18	Sequence halted during command transfer due to premature phase change (Target)
2	18	Arbitration and selection completed; sequence halted because Target failed to assert command phase

Initiator Select with $\overline{\text{ATN}}$ Steps

Internal State Register ((B)+18h) Bits 2:0 (Hex)	Interrupt Status Register ((B)+14h) Bits 7:0 (Hex)	Explanation
0	20	Arbitration steps completed; Selection time–out occurred then disconnected
4	18	Selection with $\overline{\text{ATN}}$ steps fully executed
3	18	Sequence halted during command transfer due to premature phase change; some CDB bytes may not have been sent; check FIFO flags
2	18	Message out completed; sent one message byte with $\overline{\text{ATN}}$ true, then released $\overline{\text{ATN}}$; sequence halted because Target failed to assert command phase after message byte was sent
0	18	Arbitration and selection completed; sequence halted because Target did not assert message out phase; $\overline{\text{ATN}}$ still driven by the Am53C974

Status Decode (continued):

Initiator Select with $\overline{\text{ATN3}}$ Steps		
Internal State Register ((B)+18h) Bits 2:0 (Hex)	Interrupt Status Register ((B)+14h) Bits 7:0 (Hex)	Explanation
0	20	Arbitration steps completed; Selection time-out occurred then disconnected
4	18	Selection with $\overline{\text{ATN3}}$ steps fully executed
3	18	Sequence halted during command transfer due to premature phase change; some CDB bytes may not have been sent; check FIFO flags
2	18	One , two, or three message bytes sent; sequence halted because Target failed to assert command phase after third message byte or prematurely released message out phase; $\overline{\text{ATN}}$ released only if third message byte was sent
0	18	Arbitration and selection completed; sequence halted because Target did not assert message out phase; $\overline{\text{ATN}}$ still driven by the Am53C974

Initiator Select with $\overline{\text{ATN}}$ and Stop Steps		
Internal State Register ((B)+18h) Bits 2:0 (Hex)	Interrupt Status Register ((B)+14h) Bits 7:0 (Hex)	Explanation
0	20	Arbitration steps completed; Selection time–out occurred then disconnected
0	18	Arbitration and selection completed; sequence halted because Target did not assert message out phase; $\overline{\text{ATN}}$ still driven by the Am53C974
1	18	Message out completed; one message byte sent; $\overline{\text{ATN}}$ on

5.2.3.10 Synchronous Transfer Period Register (STPREG)

Address (B)+18h **WRITE**

7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	STP4	STP3	STP2	STP1	STP0
X	X	X	0	0	1	0	1

The Synchronous Transfer Period Register (STPREG) contains a 5-bit value indicating the number of clock cycles each byte will take to be transferred over the SCSI bus in synchronous mode. The STPREG defaults to 5 clocks/byte after a hard or soft reset.

Bits 7:5 – Reserved

Bits 4:0 – STP 4:0 – Synchronous Transfer Period

The STP 4:0 bits are programmed to specify the synchronous transfer period or the number of clock cycles for each byte transferred in the synchronous mode. The minimum value for STP 4:0 is 4 clocks/byte.

The following tables list Synchronous Transfer Period options for both Fast and Normal SCSI modes. Table entries follow the binary code, and may be extrapolated if necessary. The synchronous transfer requirements as defined by the ANSI specification are listed for each instance.

	Setup	Hold	Assert/Negate
Normal synchronous	55 ns	100 ns	90 ns
Fast Synchronous	25 ns	35 ns	30 ns

FASTSCSI enabled
FASTCLK enabled, 40 MHz clock frequency:

Data hold: 2 cycles	**Assert: 2 cycles**			
STP4-0 (hex)	Clocks per cycle	Data Setup (cycles)	Negate (cycles)	Transfer Rate (MBytes/sec)
4	4	2	2	10.0
5	5	3	3	8.0
6	6	4	4	6.6
7	7	5	5	5.7
8	8	6	6	5.0
9	9	7	7	4.4
A	10	8	8	4.0
B	11	9	9	3.6
C	12	10	10	3.3
D	13	11	11	3.0

FASTSCSI disabled
FASTCLK enabled, 40 MHz clock frequency:

Data hold: 5 cycles	**Assert: 4 cycles**			
STP4-0 (hex)	Clocks per cycle	Data Setup (cycles)	Negate (cycles)	Transfer Rate (MBytes/sec)
7	8	3	4	5.0
8	9	4	5	4.4
9	10	5	6	4.0
A	11	6	7	3.6
B	12	7	8	3.3
C	13	8	9	3.0
D	14	9	10	2.8
E	15	10	11	2.6
F	16	11	12	2.5
10	17	12	13	2.3
11	18	13	14	2.2
12	19	14	15	2.1
13	20	15	16	2.0

FASTSCSI disabled
FASTCLK disabled, 25 MHz clock frequency:

Data hold: 3 cycles	Assert: 2.5 cycles			
STP4-0 (hex)	Clocks per cycle	Data Setup (cycles)	Negate (cycles)	Transfer Rate (MBytes/sec)
5	5	2	2.5	5.0
6	6	3	3.5	4.2
7	7	4	4.5	3.6
8	8	5	5.5	3.1
9	9	6	6.5	2.8
A	10	7	7.5	2.5
B	11	8	8.5	2.3
C	12	9	9.5	2.1
D	13	10	10.5	1.9

5.2.3.11 Current FIFO/Internal State Register (CFISREG)

Address (B)+1Ch **READ ONLY**

7	6	5	4	3	2	1	0
IS2	IS1	IS0	CF4	CF3	CF2	CF1	CF0
0	0	0	0	0	0	0	0

This register has two fields, the Current FIFO field and the Internal State field.

Bits 7:5 - IS 2:0 - Internal State

The Internal State Register (ISREG) tracks the progress of a sequence-type command. These bits IS 2:0 are duplicated from the IS 2:0 field in the Internal State Register ((B)+18h).

Bits 4:0 - CF 4:0 - Current FIFO

The CF 4:0 bits are the binary coded value of the number of bytes in the SCSI FIFO. These bits should not be read when the device is transferring data since this count may not be stable. The maximum value read from this register is 10h or 16 decimal due to the size of the SCSI FIFO.

When the Am53C974 is the Initiator and the phase changes to Synchronous Data In from either Message Out or Command Phase, CF4:0 will latch the number of message or command bytes that were not transmitted to the SCSI Bus. This value will be held until the next command begins. All bytes in the SCSI FIFO will be flushed, and only incoming data bytes will be retained.

5.2.3.12 Synchronous Offset Register (SOFREG)

Address (B)+1Ch **WRITE**

7	6	5	4	3	2	1	0
RAD1	RAD0	RAA1	RAA0	SO3	SO2	SO1	SO0
0	0	0	0	0	0	0	0

The Synchronous Offset Register (SOFREG) controls $\overline{\text{REQ/ACK}}$ deassertion/assertion delay and stores a 4-bit count of the number of bytes that can be sent to (or received from) the SCSI bus during synchronous transfers without a $\overline{\text{REQ}}$ (or $\overline{\text{ACK}}$). Bytes exceeding the threshold will be sent one byte at a time (asynchronously). That is, each byte will require an $\overline{\text{REQ/ACK}}$ handshake. To set up an asynchronous transfer, the SOFREG is set to zero. The SOFREG is set to zero after a hard or soft reset.

Bits 7:6 – RAD 1:0 – $\overline{\text{REQ/ACK}}$ Deassertion

These bits may be programmed to control the deassertion delay of the $\overline{\text{REQ}}$ and $\overline{\text{ACK}}$ signals during synchronous transfers. Deassertion delay is expressed in input clock cycles, and depends on the implementation of FASTCLK.
(See Control Register Three, bit 3)

SOFREG Bits 7:6	FASTCLK Ctrl 3, bit 3	Deassertion Delay $\overline{\text{REQ/ACK}}$ Input Clock Cycles
00	0	Default – 0 cycles
01	0	1/2 cycle early
10	0	1 cycle delay
11	0	1/2 cycle delay
00	1	Default – 0 cycles
01	1	1/2 cycle delay
10	1	1 cycle delay
11	1	1 1/2 cycles delay

Bits 5:4 – RAA 1:0 – $\overline{\text{REQ/ACK}}$ Assertion

These bits may be programmed to control the assertion delay of the $\overline{\text{REQ}}$ and $\overline{\text{ACK}}$ signals during synchronous transfers. Unlike deassertion delay, assertion delay is independent of the FASTCLK setting.

SOFREG Bits 5:4	Assertion Delay $\overline{\text{REQ/ACK}}$ Input Clock Cycles
00	Default – 0 cycles
01	1/2 cycle delay
10	1 cycle delay
11	1 1/2 cycles delay

Note: Exercise caution when programming bits 7:4 in the Synchronous Offset Register as it is possible to violate the SCSI-2 timing specifications.

Bits 3:0 – SO 3:0 – Synchronous Offset 3:0

The SO 3:0 bits are the binary coded value of the number of bytes that can be sent to (or received from) the SCSI bus without an $\overline{\text{ACK}}$ (or $\overline{\text{REQ}}$) signal. A zero value designates Asynchronous transfers, while a non-zero value designates the byte offset for synchronous transfers. The Am53C974 supports a maximum synchronous offset of 15 bytes.

5.2.3.13 Control Register One (CNTLREG1)

Address (B)+20h **READ/WRITE**

7	6	5	4	3	2	1	0
ETM	DISR	Reserved	PERE	Reserved	SID2	SID1	SID0
0	0	0	0	0	X	X	X

The Control Register One (CNTLREG1) programs the operating parameters for the Am53C974.

Bit 7 – ETM – Extended Timing Mode

Enabling this feature will increase the minimum setup time for data being transmitted on the SCSI bus. This bit should only be set if the external cabling conditions produce SCSI timing violations. FASTCLK operation is unaffected by this feature.

Bit 6 – DISR – Disable Interrupt on SCSI Reset

The DISR bit masks the reporting of the SCSI reset. When the DISR bit is set and a SCSI reset is asserted, the device will disconnect from the SCSI bus and remain idle without interrupting the host processor. When the DISR bit is reset and a SCSI reset is asserted the device will respond by interrupting the host processor. The DISR bit is reset to zero by a hard or soft reset.

Bit 5 – Reserved

This bit is reserved and must always be programmed to '0'.

Bit 4 – PERE – Parity Error Reporting Enable

The PERE bit enables the checking and reporting of parity errors on incoming SCSI bytes during the information transfer phase. When the PERE bit set and bad parity is detected, the PE bit in the SCSI Status Register will be set but an interrupt will not be generated. In the Initiator mode the $\overline{\text{ATN}}$ signal will also be asserted on the SCSI bus. When the PERE bit is reset and bad parity occurs, the error is not detected and no action is taken.

Bit 3 – Reserved

This bit is reserved and must always be programmed to '0'.

Bit 2:0 – SID 2:0 – SCSI ID 2:0

The Chip ID 2:0 bits specify the binary coded value of the device ID on the SCSI bus. The device will arbitrate with this ID and will respond to Reselection with this ID. At power-up the state of these bits are undefined. These bits are not affected by hard or soft reset.

5.2.3.14 Clock Factor Register (CLKFREG)

Address (B)+24h **WRITE**

7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	Reserved	Reserved	CLKF2	CLKF1	CLKF0
0	0	0	0	0	0	1	0

The Clock Factor Register (CLKFREG) must be set to indicate the input frequency range of the device. This value is crucial for controlling various timings to meet the SCSI specification. The value of bits CLKF 2:0 can be calculated by rounding off the quotient of (Input Clock Frequency in MHz)/(5 MHz). The device has a frequency range of 10 to 40 MHz.

Bits 7:3 – Reserved

These bits are reserved and must always be programmed to '0'.

Bits 2:0 – CLKF 2:0 – Clock Factor 2:0

The CLKF 2:0 bits specify the binary coded value of the clock factor. The CLKF 2:0 bits will default to a value of 2 by a hard or soft reset.

CLKF2	CLKF1	CLKF0	Input Clk Freq (MHz)
0	1	0	10
0	1	1	10.01 to 15
1	0	0	15.01 to 20
1	0	1	20.01 to 25
1	1	0	25.01 to 30
1	1	1	30.01 to 35
0	0	0	35.01 to 40

Note: CLKF2:0 must be set to '000' (binary) and a 40 MHz clock must be used to generate the CLK signal in order to achieve 10 MB/sec synchronous transfer rates.

5.2.3.15 RESERVED

Address (B)+28h **WRITE**

7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0	0	0	0	0	0	0	0

Bits 7:0 – Reserved

These bits are reserved and must always be programmed to '00h'.

5.2.3.16 Control Register Two (CNTLREG2)

Address (B)+2Ch **READ/WRITE**

7	6	5	4	3	2	1	0
Reserved	ENF	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0	0	0	0	0	0	0	0

Control Register Two (CNTLREG2) programs various operating parameters for the Am53C974.

Bit 7 – Reserved

This bit is reserved and must always be programmed to '0'.

Bit 6 – ENF – Enable Features

When set to a value of '1', this bit activates the following product enhancements:

1) The Current Transfer Count Register High ((B)+38h) will be enabled, extending the transfer counter from 16 to 24 bits to allow for larger transfers.
2) Following a chip or power on reset, up until the point where the Current Transfer Count Register High ((B)+38h) is loaded with a value, reading this register will return the Am53C974's part-unique ID.
3) The SCSI phase will be latched at the completion of each command by bits 2:0 in the SCSI Status Register ((B)+10h). When this bit is '0', the SCSI Status Register will reflect real-time SCSI phases.

A software or hardware reset will clear this bit to its default value of '0'; a SCSI reset will leave this bit unaffected.

Bit 5:4 – Reserved (Read Only)

This bit is reserved and will always return a value of '0' when read.

Bit 3:0 – Reserved

These bits are reserved and must always be programmed to '0'.

5.2.3.17 Control Register Three (CNTLREG3)

Address (B)+30h **READ/WRITE**

7	6	5	4	3	2	1	0
ADIDCHK	Reserved	Reserved	FASTSCSI	FASTCLK	Reserved	Reserved	Reserved
0	0	0	0	0	0	0	0

Bit 7 – ADIDCHK – Additional ID Check

This bit enables additional check on ID message during bus-initiated Select with $\overline{\text{ATN}}$. The Am53C974 will check bits 7, and bits 5:3 in the first byte of the ID message during Selection. An interrupt will be generated if bit 7 is '0', or if bits 5, 4, or 3 are '1'.

Bit 6:5 – Reserved

These bits are reserved and must always be programmed to '0'.

Bit 4 – FASTSCSI – Fast SCSI

Bit 3 – FASTCLK – Fast SCSI Clocking

These bits configure the Am53C974's state machine to support both Fast SCSI timings and SCSI-1 timings. These bits affect the SCSI transfer rate, and must be considered in conjunction with the Am53C974's clock frequency and mode of operation.

CNTLREG3			
FASTSCSI Bit 4	**FASTCLK Bit 3**	**Clock Frequency**	**Mode of Operation**
1	1	25 MHz – 40 MHz	10 MBytes/sec, Fast SCSI
0	1	25 MHz – 40 MHz	5 MBytes/sec, SCSI-1
—	0	<=25 MHz	5 MBytes/sec, SCSI-1

— = don't care

Bit 2 – Reserved

This bit is reserved and must always be programmed to '0'.

Bit 1 – Reserved (Read Only)

This bit is reserved and will always return a value of '0' when read.

Bit 0 – Reserved

This bit is reserved and must always be programmed to '0'.

5.2.3.18 Control Register Four (CNTLREG4)

Address (B)+34h **READ/WRITE**

7	6	5	4	3	2	1	0
GE1	GE0	PWD	Reserved	RES (R) RAE (W)	Reserved	Reserved	Reserved
0	0	0	0	0	0	0	0

This register is used to control several features implemented in the SCSI block.
At power up, this register will contain a '0' value on all bits except bit 4.

Bit 7:6 – GE1:0 – GLITCH EATER

The GLITCH EATER Circuitry has been implemented on $\overline{REQ}$, $\overline{ACK}$, and data lines and are controlled by bits 7and 6. The valid signal window may be adjusted by setting the bits according to the combinations listed below.

CNTLREG4		
Bit 7 GE1	**Bit 6 GE0**	**Valid Signal Window**
0	0	12 ns
1	0	25 ns
0	1	35 ns
1	1	0 ns

Note: Changes in the valid signal window will affect data setup and hold times for Fast SCSI timings.

Bit 5 – PWD – Reduced Power Feature

Setting this bit to '1' enables AMD's reduced power feature. This feature turns off the input buffers on all the SCSI bus signal lines to reduce power consumption. For further power savings, the clock input may be removed to the Am53C974 via external logic.

Bit 4 – Reserved

This bit is reserved for internal use.

Bit 3 (Read Only) – RES – Reserved

This bit is reserved for internal use.

Bit 3 (Write Only) – RAE – Active Negation Control

Bit 2 – RADE – Active Negation Control

Bits 2 and 3 control the Active Negation Drivers which may be enabled on $\overline{\text{REQ}}$, $\overline{\text{ACK}}$, or DATA lines. The following table shows the programming options for this feature:

CNTLREG4		
Bit 3 RAE	Bit 2 RADE	Function Selected
0	0	Active Negation Disabled
1	0	Active Negation on $\overline{\text{REQ}}$ and $\overline{\text{ACK}}$ only
—	1	Active Negation on $\overline{\text{REQ}}$, $\overline{\text{ACK}}$ and DATA

— = don't care

Bit 1:0 – Reserved

This bit is reserved for internal use.

5.2.3.19 RESERVED

Address (B)+3Ch **WRITE**

7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0	0	0	0	0	0	0	0

Bit 7:0 – Reserved

These bits are reserved and must always be programmed to '00h'.

5.3.3.20 Part-Unique ID Register (CTCREG)

Address (B)+38h **READ ONLY**

This register extends the transfer counter from 16 to 24 bits and is only enabled when the ENF bit is set (bit 6, Control Register Two). The descriptions accompanying the Start Transfer Count Registers and the Current Transfer Count Registers should be referenced for more information regarding the transfer counter.

This register is also used to store the part-unique ID code for the Am53C974. This information may be accessed when all of the following are true:

- A power up or chip reset has taken place
- A value has not been loaded into this register

5.3 DEVICE COMMANDS

The device commands can be broadly divided into two categories, DMA commands and non-DMA commands. DMA commands are those which cause data movement between the host memory and the SCSI bus while non-DMA commands are those that cause data movement between the device FIFO and the SCSI bus. The Most Significant Bit of the command byte differentiate the DMA from the non-DMA commands.

When a DMA command is issued, the contents of the Start Transfer Count Register will be loaded into the Current Transfer Count Register. Data transmission will continue until the Current Transfer Count Register decrements to zero.

Non-DMA commands do not modify the Current Transfer Count Register and are unaffected by the value in the Current Transfer Count Register. For non-DMA commands, the number of bytes transmitted depends solely on the operation in progress.

Command	Code (Hex.)	
	Non-DMA Mode	DMA Mode
Initiator Commands		
Information Transfer	10	90
Initiator Command Complete Steps	11	91
Message Accepted	12	–
Transfer Pad Bytes	–	98
Set $\overline{\text{ATN}}$*	1A	–
Reset $\overline{\text{ATN}}$*	1B	–
Idle State Commands		
Select without $\overline{\text{ATN}}$ Steps	41	C1
Select with $\overline{\text{ATN}}$ Steps	42	C2
Select with $\overline{\text{ATN}}$ and Stop Steps	43	C3
Enable Selection/Reselection*	44	C4
Disable Selection/Reselection	45	–
Select with $\overline{\text{ATN}}$3 Steps	46	C6
General Commands		
No Operation*	00	80
Clear FIFO*	01	–
Reset Device*	02	–
Reset SCSI Bus**	03	–

Notes:

** These commands do not generate interrupt.*

*** An interrupt is generated when SCSI bus reset interrupt reporting is not disabled (see Control Register1/DISR bit6).*

5.3.1 Command Stacking

The microprocessor may stack commands in the Command Register ((B)+0Ch) since it functions as a two-byte deep FIFO. Non-DMA commands may not be stacked, and commands which transfer data in opposing directions should not be stacked together.

If DMA commands are queued together, the Start Transfer Count must be written before the associated command is loaded into the Command Register. Since multiple interrupts can occur when commands are stacked, it is recommended that the ENF bit in Control Register Two (Bit 6) be set in order to the SCSI phase bits in the SCSI Status Register ((B)+10h) at the completion of a command. This allows the host to determine the phase of the interrupting command without having to consider phase changes that occurred after the stacked command began execution.

Note: Command stacking and queuing should only be used during SCSI Data In or Data Out transfers.

5.3.2 Invalid Commands

When an illegal command is written to the Am53C974, the Invalid Command Bit (Bit 6, Register (B)+14h) will be set to '1', and an interrupt will be generated to the host. When this happens, the interrupt must be serviced before another command may be written to the Command Register.

An Invalid command is defined as a command written to the Am53C974 that is either not supported, not allowed in the specified mode, or a command that has an unsupported command mode.

The following conditions will also cause an Invalid Command interrupt to occur:

- An Initiator Information Transfer, Transfer Pad, or Command Complete is issued when $\overline{\text{ACK}}$ is still asserted.
- A Selection command is issued with the DMA bit enabled, if the Selection command was previously issued with the DMA enabled.

5.3.3 Command Window

The window at the point where the Disable Selection/Reselection command (45h/C5h) has been loaded into the Command Register ((B)+0Ch), and before bus-initiated Selection begins, has been eliminated. This prevents a false Successful Operation Interrupt from being generated when the Selection sequence continues to completion after the Disable command has been loaded.

5.3.4 Initiator Commands

Initiator commands are executed by the device when it is in the Initiator mode. If the device is not in the Initiator mode and an Initiator command is received the device will ignore the command, generate an Invalid Command interrupt and clear the Command Register.

Should the Target disconnect from the SCSI bus by deasserting the $\overline{\text{BSY}}$ signal line while the Am53C974 (Initiator) is waiting for the Target to assert $\overline{\text{REQ}}$, a Disconnected Interrupt will be issued 1.5 to 3.5 clock cycles following $\overline{\text{BSY}}$ going false.

Upon receipt of the last byte during Message In phase, $\overline{\text{ACK}}$ will remain asserted to prevent the Target from issuing any additional bytes, while the Initiator decides to accept/reject the message. If non-DMA commands are used, the last byte signals the SCSI FIFO is empty. If DMA commands are used, the Current Transfer Count signals the last byte.

A Reset SCSI Bus command (03h/83h) will force the Am53C974 to abort the current operation and disconnect from the bus. If the DISR bit is reset (Bit 6, Control Register One (B)+20h)), the host processor will be interrupted with a SCSI Reset Interrupt before the Am53C974 proceeds to Disconnect.

If parity checking is enabled in the Initiator mode during the Data-in phase and an error is detected, ATN will be asserted for the erroneous byte before deasserting ACK.

5.3.4.1 Information Transfer Command (Command Code 10h/90h)

The Information Transfer command is used to transfer information bytes over the SCSI bus. This command may be issued during any SCSI Information Transfer phase. Synchronous data transmission requires use of the DMA mode.

The device will continue to transfer information until it is terminated by any one of the following conditions:

- The Target changes the SCSI bus phase before the expected number of bytes are transferred. The Am53C974 clears the Command Register (CMDREG), and generates a Service Request interrupt when the Target asserts REQ.
- Transfer has successfully completed. If the phase is Message Out, the Am53C974 deasserts ATN before asserting ACK for the last byte of the message. When the Target asserts REQ, a Service Request interrupt is generated.
- In the Message In phase when the device receives the last byte. The Am53C974 keeps the ACK signal asserted and generates a Successful Operation interrupt.

During Synchronous Data transfers the Target may send up to the maximum synchronous threshold number of REQ pulses to the Initiator. If it is the Synchronous Data–In phase then the Target sends the data and the REQ pulses. These bytes are stored by the Initiator in the FIFO as they are received.

The Information Transfer Command, when issued during the following SCSI phases and terminated in Synchronous Data phases, is handled as described below:

- Message In/Status Phase – When a phase change to Synchronous Data-In or Synchronous Data-Out is detected by the device, the Command Register is cleared and the DMA interface is disabled to prevent any transfer of data (phase) bytes.
- If the phase change is to Synchronous Data-In and bad parity is detected on the data bytes coming in, it is not reported since the Status Register will report the status of the command just completed. The parity error flag and the ATN signal will be asserted when the next Information Transfer command begins execution.
- Message Out/Command Phase – When a phase change to Synchronous Data–In or Synchronous Data-Out is detected by the device, the Command Register is cleared and the DMA interface is disabled to prevent any transfer of data (phase) bytes.
- If the phase change is to Synchronous Data-In and bad parity is detected on the data bytes coming in, it is not reported since the Status Register will report the status of the command just completed. The parity error flag and the ATN signal will be asserted when the next Information Transfer command begins execution.
- The SCSI FIFO Register will be latched and will remain in that condition until the next command begins execution. The value in the SCSI FIFO Register indicates the number of non-data bytes in the SCSI FIFO when the phase changed to Synchronous Data-In. These bytes are cleared from the FIFO, and only incoming data bytes are retained.

- In the Synchronous Data-Out phase, the threshold counter is incremented as $\overline{\text{REQ}}$ pulses are received. The transfer is completed when the FIFO is empty and the Current Transfer Count Register is '0'. The threshold counter will not be '0'.
- In the Synchronous Data-In phase, the Current Transfer Count Register is decremented as bytes are read from the SCSI FIFO rather than when the bytes are being written to the SCSI FIFO. The transfer is completed when Current Transfer Count Register is '0'. However, the SCSI FIFO may not be empty.

5.3.4.2 Initiator Command Complete Steps (Command Code 11h/91h)

The Initiator Command Complete Steps command is normally issued when the SCSI bus is in the Status In phase. One Status byte followed by one Message byte is transferred if this command completes normally. After receiving the message byte the device will keep the $\overline{\text{ACK}}$ signal asserted to allow the Initiator to examine the message and assert the $\overline{\text{ATN}}$ signal if it is unacceptable. The command terminates early if the Target does not switch to the Message In phase or if the Target disconnects from the SCSI bus. This command does not utilize the Internal State Register ((B)+18h).

5.3.4.3 Message Accepted Command (Command Code 12h)

The Message Accepted Command is used to release the $\overline{\text{ACK}}$ signal. This command is normally used to complete a Message In handshake. Upon execution of this command the device generates a Service Request interrupt after $\overline{\text{REQ}}$ is asserted by the Target.

After the device has received the last byte of message, it keeps the $\overline{\text{ACK}}$ signal asserted. This allows the device to either accept or reject the message. To accept the message, Message Accepted Command is issued. To reject the message the $\overline{\text{ATN}}$ signal must be asserted (with the help of the Set $\overline{\text{ATN}}$ Command) before issuing the Message Accepted Command. In either case, the Message Accepted Command has to be issued to release the $\overline{\text{ACK}}$ signal.

5.3.4.4 Transfer Pad Bytes Command (Command Code 18h/98h)

The Transfer Pad Bytes Command is used to recover from an error condition. This command is similar to the Information Transfer Command, only the information bytes consists of null data. It is used when the Target expects more data bytes than the Initiator has to send. It is also used when the Initiator receives more information than expected from the Target.

When sending data to the SCSI bus, the SCSI FIFO is loaded with null bytes which are sent out to the SCSI bus. Although an actual DMA request is not made, DMA interface must be enabled when pad bytes are transmitted since the Am53C974 uses the Current Transfer Count Register to terminate transmission.

This command terminates under the same conditions as the Information Transfer Command, but the device does not keep the $\overline{\text{ACK}}$ signal asserted during the last byte of the Message In phase. Should this command terminate prematurely due to a Disconnect or a phase change before the Current Transfer Count Register decrements to zero, the SCSI FIFO may contain residual Pad bytes.

5.3.4.5 Set $\overline{\text{ATN}}$ Command (Command Code 1Ah)

The Set $\overline{\text{ATN}}$ Command is used to drive the $\overline{\text{ATN}}$ signal active on the SCSI bus. An interrupt is not generated at the end of this command. The $\overline{\text{ATN}}$ signal is deasserted before asserting the $\overline{\text{ACK}}$ signal during the last byte of the Message Out phase.

Note: The $\overline{\text{ATN}}$ signal is asserted by the device without this command in the following cases:

- *If any select with $\overline{\text{ATN}}$ command is issued and the arbitration is won.*
- *An Initiator needs the Target's attention to send a message. The $\overline{\text{ATN}}$ signal is asserted before deasserting the $\overline{\text{ACK}}$ signal.*

5.3.4.6 Reset $\overline{\text{ATN}}$ Command (Command Code 1Bh)

The Reset $\overline{\text{ATN}}$ Command is used to deassert the $\overline{\text{ATN}}$ signal on the SCSI bus. An interrupt is not generated at the end of this command. This command is used only when interfacing with devices that do not support the Common Command Set (CCS). These older devices do not deassert their $\overline{\text{ATN}}$ signal automatically on the last byte of the Message Out phase. This device does deassert its $\overline{\text{ATN}}$ signal automatically on the last byte of the Message Out phase.

5.3.5 Idle State Commands

The Idle State Commands can be issued to the device only when the device is disconnected from the SCSI bus. If these commands are issued to the device when it is logically connected to the SCSI bus, the commands are ignored, an Invalid Command interrupt is generated, and the Command Register (CMDREG) is cleared.

5.3.5.1 Select without $\overline{\text{ATN}}$ Steps Command (Command Code 41h/C1h)

The Select without $\overline{\text{ATN}}$ Steps Command is used by the Initiator to select a Target. When this command is issued, the device arbitrates for the control of the SCSI bus. When the device wins arbitration, it selects the Target device and transfers the Command Descriptor Block (CDB). Before issuing this command the SCSI Timeout Register (STIMREG), Control Register One (CNTLREG1), and the SCSI Destination ID Register (SDIDREG) must be set to the proper values. If DMA is enabled, the Start Transfer Count Register (STCREG) must be set to the total length of the command. If DMA is not enabled, the data must be loaded into the FIFO before issuing this command. This command will be terminated early if the SCSI Timeout Register times out, if the Target does not go to the Command Phase following the Selection Phase, or if the Target exits the Command Phase prematurely. A Successful Operation interrupt will be generated following normal command execution.

5.3.5.2 Select with $\overline{\text{ATN}}$ Steps Command (Command Code 42h/C2h)

The Select with $\overline{\text{ATN}}$ Steps Command is used by the Initiator to select a Target. When this command is issued, the device arbitrates for the control of the SCSI bus. When the device wins arbitration, it selects the Target device with the $\overline{\text{ATN}}$ signal asserted and transfers the Command Descriptor Block (CDB) and a one byte message. Before issuing this command the SCSI Timeout Register (STIMREG), Control Register One (CNTLREG1) and the SCSI Destination ID Register (SDIDREG) must be set to the proper values. If DMA is enabled, the Start Transfer Count Register (STCREG) must be set to the total length of the command and message. If DMA is not enabled, the data must be loaded into the FIFO before issuing this command. This command will be terminated early in the following situations:

- The SCSI Timeout Register times out
- The Target does not go to the Message Out Phase following the Selection Phase
- The Target exits the Message Phase early
- The Target does not go to the Command Phase following the Message Out Phase
- The Target exits the Command Phase early

A Successful Operation/Service Request interrupt is generated when this command is completed successfully.

5.3.5.3 Select with $\overline{\text{ATN}}$ and Stop Steps Command (Command Code 43h/C3h)

The Select with $\overline{\text{ATN}}$ and Stop Steps Command is used by the Initiator to send messages with lengths other than 1 or 3 bytes. When this command is issued, the device executes the Selection process, transfers the first message byte, then STOPS the sequence. $\overline{\text{ATN}}$ is not deasserted at this time, allowing the Initiator to send additional message bytes after the ID message. To send these additional bytes, the Initiator must write the transfer counter with the number of bytes which will follow, then issue a Transfer Information Command. (Note: the Target is still in the Message Out phase when this command is issued). $\overline{\text{ATN}}$ will remain asserted until the Current Transfer Count Register decrements to zero.

The SCSI Timeout Register (STIMREG), Control Register One (CNTLREG1), and the SCSI Destination ID Register (SDIDREG) must be set to the proper values before the Initiator issues this command. This command will be terminated early if the STIMREG times out or if the Target does not go to the Message Out Phase following the Selection Phase.

5.4.5.4 Enable Selection/Reselection Command (Command Code 44H/C4H)

The Enable Selection/Reselection Command is used to respond to a bus-initiated Selection or Reselection. Upon disconnecting from the bus the Selection/Reselection circuit is automatically disabled by the device. This circuit must be enabled for the Am53C974 to respond to subsequent reselection attempts and the Enable Selection/Reselection Command is issued to do that. This command is normally issued within 250 ms (select/reselect timeout) after the device disconnects from the bus. If DMA is enabled, the device loads the received data to the buffer memory. If the DMA is disabled, the received data stays in the FIFO.

5.3.5.5 Disable Selection/Reselection Command (Command Code 45H)

The Disable Selection/Reselection Command is used by the Target to disable response to a bus-initiated Reselection. When this command is issued before a bus-initiated Selection or Reselection is in progress, it resets the internal state bits previously set by the Enable Selection/Reselection Command. The device also generates a Successful Operation interrupt to the processor. If however, this command is issued after a bus-initiated Selection/Reselection has begun, this command and all incoming commands are ignored since the Command Register (CMDREG) is held reset. The Am53C974 also generates a Selected or Reselected interrupt when the sequence is complete.

5.3.5.6 Select with $\overline{\text{ATN}}$3 Steps Command (Command Code 46h/C6h)

The Select with $\overline{\text{ATN}}$3 Steps Command is used by the Initiator to select a Target. This command is similar to the Select with $\overline{\text{ATN}}$ Steps Command, except that it sends exactly three message bytes. When this command is issued the Am53C974 arbitrates for control of the SCSI bus. When the device wins arbitration, it selects the Target device with the $\overline{\text{ATN}}$ signal asserted and transfers the Command Descriptor Block (CDB) and three message bytes. Before issuing this command the SCSI Timeout Register (STIMREG), Control Register One (CNTLREG1), and the SCSI Destination ID Register (SDIDREG) must be set to the proper values. If DMA is enabled, the Start Transfer Count Register (STCREG) must be set to the total length of the command. If DMA is not enabled, the data must be loaded into the FIFO before issuing this command. This command will be terminated early in the following situations:

- The SCSI Timeout Register times out
- The Target does not go to the Message Out Phase following the Selection Phase
- The Target removes Command Phase early
- The Target does not go to the Command Phase following the Message Out Phase
- The Target exits the Command Out Phase early

A Successful Operation/Service Request interrupt is generated when this command is executed successfully.

5.3.6 General Commands

5.3.6.1 No Operation Command (Command Code 00h/80h)

The No Operation Command administers no operation, therefore an interrupt is not generated upon completion. This command is issued following the Reset Device Command to clear the Command Register (CMDREG). A No Operation Command in the DMA mode may be used to verify the contents of the Start Transfer Count Register (STCREG). After the STCREG is loaded with the transfer count and a DMA No Operation Command is issued, reading the Current Transfer Count Register (CTCREG) returns the transfer count value.

5.3.6.2 Clear FIFO Command (Command Code 01h)

The Clear FIFO Command is used to initialize the SCSI FIFO to the empty condition. The Current FIFO Register (CFISREG) reflects the empty FIFO status and the bottom of the FIFO is set to zero. No interrupt is generated at the end of this command.

5.3.6.3 Reset Device Command (Command Code 02h)

The Reset Device Command immediately stops any device operation and resets all the functions of the device. Additionally, it returns the device to the disconnected state and it generates a hard reset. The Reset Device Command remains on the top of the Command Register FIFO holding the device in the reset state until the No Operation Command is loaded. Once loaded, the No Operation command serves to re-enable the Command Register.

5.3.6.4 Reset SCSI Bus Command (Command Code 03h)

The Reset SCSI Bus Command forces the $\overline{\text{SCSIRST}}$ signal active for a period of 25 ms, and drives the chip to the Disconnected state. An interrupt is not generated upon command completion, however, if bit 6 is set to '0' in Control Register One (CNTLREG1), a SCSI Reset interrupt will be issued.

CHAPTER

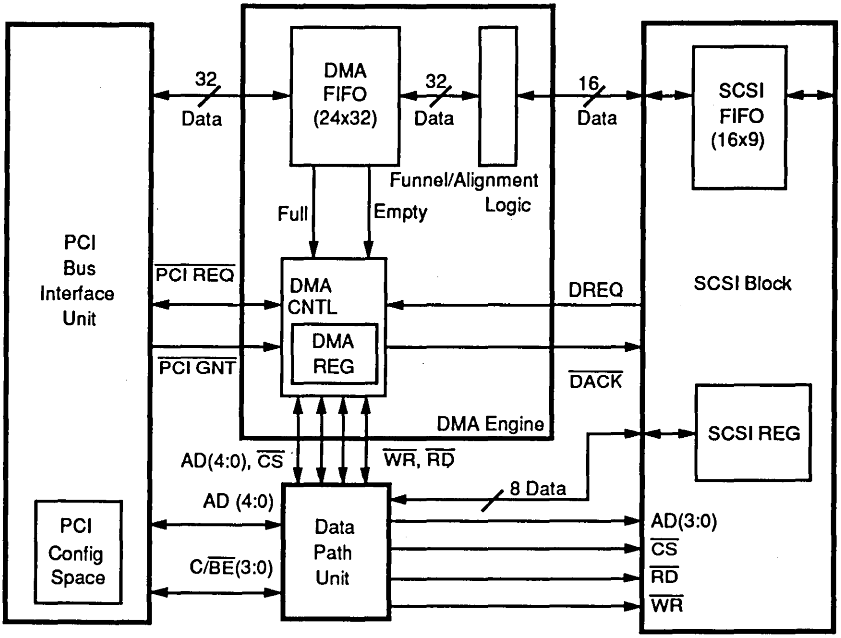
6 DMA ENGINE

6.1 INTRODUCTION

The DMA Engine in the Am53C974 provides bus-mastering capabilities to allow flexibility and performance advantages over slave PCI-SCSI devices. Built into the engine is a 96-byte (24 DW) FIFO and additional logic to handle the transition between the 32 bit PCI bus and the 8-bit SCSI bus.

Figure 6-1 illustrates the DMA Engine in relation to the PCI interface and the SCSI block. As its most basic function, the DMA engine acts as the DMA controller in a bus master capacity on the PCI bus, transferring data between memory and the SCSI block. All Command, Data, Status, and Message bytes pass through the DMA FIFO on their way to or from the SCSI bus. However, for PIO accesses to the SCSI registers, the DMA FIFO is bypassed as data moves directly from the SCSI block to the PCI interface. Since PIO operations do not pass through the funneling logic and DMA FIFO, data is transferred one byte at a time from the SCSI block to the PCI interface via the least significant byte lane. (The three most significant byte lanes will contain null data.)

Figure 6-1 PCI BIU – DMA Engine – SCSI Block



18264B-22

Since the PCI bus is 4 bytes wide and the SCSI bus is only 1 byte wide, funneling logic is included in this engine to handle byte alignment and to ensure that data is properly transferred between the SCSI bus and the wider PCI bus. All boundary conditions are handled through hardware by the DMA Engine.

The DMA engine is also designed for block type (4 KByte page) transfers to support scatter-gather operations. Implementation of this feature is described further in Section 6.8.

6.2 DATA PATH UNIT

The Data Path Unit receives address AD(4:0) and C/$\overline{\text{BE}}$ (3:0) inputs from the PCI bus through the PCI Bus Interface Unit, and routes appropriate addresses, $\overline{\text{RD}}$, $\overline{\text{WR}}$, and $\overline{\text{CS}}$ control lines to either the DMA engine or the SCSI block, depending on the state of the AD4 address line. If AD4 is '0', then register accesses are to the SCSI registers. However, if AD4 is '1', then register accesses are routed to the DMA registers.

6.3 DMA FIFO

Data transfers from the SCSI FIFO to the DMA FIFO take place each time the threshold of two bytes is reached on the SCSI side. The transfer is initiated by the SCSI block when DREQ is asserted, and continues with the $\overline{\text{DACK}}$ handshaking which typically takes place in DMA accesses. Data is accumulated in the DMA FIFO until a threshold of 16 DW (64 bytes) is reached. Data is then burst across the PCI bus to memory. Residue data which is less than the threshold in each FIFO is sent in non-contiguous bursts. For memory read operations, data is sent in burst mode to the DMA FIFO and continues through to the SCSI FIFO and onto the SCSI bus.

6.3.1 DMA BLAST Command

This command is used to retrieve the contents of the DMA FIFO when the Target disconnects during a DMA Write operation. Users are cautioned against using this command for recovery during a DMA read operation.

If the current DMA write operation is interrupted by a Target disconnect ($\overline{\text{INTA}}$ asserted, and SCSI Transfer Counter ≠ 0), the DMA BLAST command may be used to retrieve any data bytes within the DMA FIFO. The following procedure outlines its use:

1. Read the DMA Status Register. It should indicate that a SCSI interrupt is pending.
2. Read the SCSI FIFO Flags register. If the value ≠ 0, wait for the SCSI FIFO to empty its contents into the DMA FIFO. When the value = 0, execute the DMA BLAST command.

Note: In some odd byte conditions, one residual byte will be left in the SCSI FIFO, and the FIFO Flags will never count to '0'. When this happens, the residual byte should be retrieved via PIO following completion of the BLAST operation

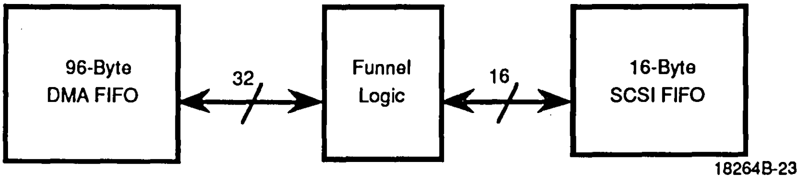
3. The DMA BLAST command is executed by writing '01' to bits 1:0 in the DMA Command Register. When issued, this command will move the DMA FIFO contents into the system memory.
4. Completion of the BLAST operation is signaled when bit 5 in the DMA Status Register is set to '1'.
5. Set the DMA engine to IDLE when all data bytes have been moved to memory.

Note: An interrupt is not generated upon completion of this command.

6.4 FUNNELING LOGIC

Figure 6-4 shows the internal DMA logic interface with the SCSI block. The DMA FIFO interfaces to the Funnel Logic block via a 32-bit data bus, and the funnel logic properly reduces this stream of data to a 16-bit stream to properly interface with the SCSI FIFO.

Figure 6-4 DMA FIFO to SCSI FIFO Interface



6.5 SCSI DMA PROGRAMMING SEQUENCE

The following section outlines the procedure for executing SCSI DMA operations:

1. Program the DMA Engine to the IDLE state
2. Program the SCSI block registers (e.g. synchronous operation, offset values, etc.)
3. Program the DMA registers
4. START the DMA engine
5. At the end of the DMA transaction, set the DMA engine back to the IDLE state

6.6 MDL BASED DMA PROGRAMMING

The following section outlines the procedure for executing MDL based DMA operation:

1. Set up the MDL list
2. Use the programming sequence defined earlier for initiating a SCSI DMA transfer

6.7 DMA Registers

The following is a description of the DMA register set or the DMA Channel Context Block (DMA CCB). These registers control the specifics for DMA operations such as transfer length and scatter-gather options. The three read-only working counter registers allow the system software and driver to monitor the DMA transaction. The register addresses are represented by the PCI Configuration Base Address (B) and its corresponding offset value. The Base address for the Am53C974 is stored at register address (10h).

Register Acronym	Address (Hex.)	Register Description	Type
CMD	(B)+40	Command	R/W
STC	(B)+44	Starting Transfer Count	R/W
SPA	(B)+48	Starting Physical Address	R/W
WBC	(B)+4C	Working Byte Counter	R
WAC	(B)+50	Working Address Counter	R
STATUS	(B)+54	Status Register	R
SMDLA	(B)+58	Starting Memory Descriptor List (MDL) Address	R/W
WMAC	(B)+5C	Working MDL Counter	R

6.7.1 Command Register (CMD)
Address (B)+40h READ /WRITE

The upper 3 bytes of the Command Register are reserved while the remaining (LSB) byte is used to control different features of the DMA engine. This register must be written twice to ensure proper operation.

The first PCI I/O write to this register must issue an 'IDLE' command (CMD1:0 = '01' or '00') and set bits 7:4 for the DMA operation. The following PCI I/O write should then issue a 'START' command (CMD1:0 = '11') to begin the DMA operation. During this second register write cycle, bits 7 and 4 must be programmed as they were during the first PCI I/O write cycle, and must be preserved throughout the operation. The settings for the interrupt control bits (bits 6:5) do not need to be preserved. However, the features that bits 6:5 control will be implemented according to their state at the time a 'START' command is issued. At the completion of the DMA operation, the DMA engine should be restored to the *IDLE* state.

The power up/reset state is shown in the register map that follows.

31	30	29	28	27	26	25	24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
X	X	X	X	X	X	X	X

23	22	21	20	19	18	17	16
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
X	X	X	X	X	X	X	X

15	14	13	12	11	10	9	8
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
X	X	X	X	X	X	X	X

7	6	5	4	3	2	1	0
DIR	INTE_D	INTE_P	MDL	Reserved	DIAG	CMD1	CMD0
0	0	0	0	X	0	0	0

Bit 31:8 – Reserved

These bits are reserved for future implementation and should always be programmed to '0'.

Bit 7 – DIR – Direction of Transfer

This bit, active logic '1' true, indicates a read data transfer (i.e., read from device – DMA write to memory). A logic '0' false value implies a write data transfer (DMA read from memory – write to device). This bit is cleared by a hard reset.

Bit 6 – INTE_D – DMA Transfer Interrupt Enable

This bit active logic '1' true, enables the channel to cause an active interrupt upon completion or receipt of an error condition during a DMA transfer. This bit is cleared by a hard reset.

Bit 5 – INTE_P – Page Transfer Interrupt Enable

This bit active logic '1' true, enables the channel to cause an active interrupt upon the completion of each page transfer during DMA Scatter-Gather operations. This bit is cleared by a hard reset.

Bit 4 – MDL – Map to Memory Descriptor List

This bit, active logic '1' true, enables the mapping of physical memory addresses into the Memory Descriptor List (MDL). In this mode, the SPA register will contain the translated MDL entries. When this bit is '0', the Am53C974 will operate in non-MDL mode, in which case the SPA register will contain actual 32-bit physical memory addresses.

Bit 3 – Reserved

This bit is reserved for future implementation and should be programmed to '0'.

Bit 2 – DIAG – Diagnostic

This bit is reserved for diagnostics only. It will be reset to '0' and shall remain zero. When this bit is set with the BLAST command, the Am53C974 will fill the memory buffer with random data and set the 'DONE' bit in the DMA Status register, regardless of the SCSI bus activities.

Bit 1:0 – CMD1:0 – Command Code Bits

These bits are encoded to represent commands for the DMA engine. They will be cleared by a hard reset condition. Command codes are described as follows:

CMD1	CMD0	Command	Description
0	0	IDLE	The DMA channel is inactive. Writes of these values are NOPs.
0	1	BLAST	Empties all data bytes in DMA FIFO to memory during a DMA write operation. Upon completion, the 'BCMPLT' bit will be set in the DMA Status register. This command should not be used during a DMA read operation.
1	0	ABORT	Terminates the current DMA transfer. The DMA engine should be restored to the 'IDLE' state following execution of this command. *Note: This is only valid after a 'START' command is issued.*
1	1	START	Initiates a new DMA transfer. These bits must remain set throughout the DMA operation until the 'DONE' bit in the DMA Status Register is set. *Note: This command should be issued only after all other control bits have been initialized.*

X= don't care

6.7.2 Starting Transfer Count (STC) Address (B)+44h READ/WRITE

The STC register contains a 24-bit read/write value which represents the number of bytes to be transferred. The value programmed in this register should be identical to the value programmed in the SCSI Start Transfer Count Register ((B+00h, (B)+04h, (B)+38h). This register is not modified by the DMA transfer logic, and can be read by the software at any time. The system software may modify this register after the DMA transfer has been started. For each transfer, this register must be reloaded.

31	30	29	28	27	26	25	24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
X	X	X	X	X	X	X	X

23	22	21	20	19	18	17	16
STC23	STC22	STC21	STC20	STC19	STC18	STC17	STC16
X	X	X	X	X	X	X	X

15	14	13	12	11	10	9	8
STC15	STC14	STC13	STC12	STC11	STC10	STC9	STC8
X	X	X	X	X	X	X	X

7	6	5	4	3	2	1	0
STC7	STC6	STC5	STC4	STC3	STC2	STC1	STC0
X	X	X	X	X	X	X	X

Bit 31:24 – Reserved

These bits are reserved for future implementation and should always be programmed to '0'.

Bit 23:0 – STC23:0 – Starting Transfer Count

This 24-bit count represents the number of bytes to be transferred during the current DMA operation. These bits shall be cleared to an indeterminate state following a hard reset.

6.7.3 Starting Physical Address (SPA)

Address (B)+48h READ/WRITE

The SPA register is a 32-bit read/write address that is used as the starting address value for the DMA transfer. This register is not modified by the DMA transfer logic, and can be read by the software at any time. This register may be modified after the DMA transfer has been started.

31	30	29	28	27	26	25	24
SPA31	SPA30	SPA29	SPA28	SPA27	SPA26	SPA25	SPA24
X	X	X	X	X	X	X	X

23	22	21	20	19	18	17	16
SPA23	SPA22	SPA21	SPA20	SPA19	SPA18	SPA17	SPA16
X	X	X	X	X	X	X	X

15	14	13	12	11	10	9	8
SPA15	SPA14	SPA13	SPA12	SPA11	SPA10	SPA9	SPA8
X	X	X	X	X	X	X	X

7	6	5	4	3	2	1	0
SPA7	SPA6	SPA5	SPA4	SPA3	SPA2	SPA1	SPA0
X	X	X	X	X	X	X	X

Bit 31:0 – SPA31:0 – Starting Physical Address

This value represents the starting memory address for the DMA transfer. During a scatter-gather operation, bits 31:12 will be programmed through hardware with the MDL entries. Therefore, bits 11:0 should be programmed with the starting page offset.

6.7.4 Working Byte Counter (WBC)

Address (B)+4Ch READ ONLY

The WBC register contains a 24-bit read-only counter that is initialized to the value in the STC register when the transfer begins. The counter occupies the 3 lower bytes of this address, and reflects the number of bytes transferred during a DMA operation. When the DMA transfer stops, a non zero value in this register indicates that the operation aborted earlier than expected (i.e. an error occurred). This registers' intermediate value may be read by software between DMA burst transactions.

31	30	29	28	27	26	25	24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
X	X	X	X	X	X	X	X

23	22	21	20	19	18	17	16
WBC23	WBC22	WBC21	WBC20	WBC19	WBC18	WBC17	WBC16
X	X	X	X	X	X	X	X

15	14	13	12	11	10	9	8
WBC15	WBC14	WBC13	WBC12	WBC11	WBC10	WBC9	WBC8
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
WBC7	WBC6	WBC5	WBC4	WBC3	WBC2	WBC1	WBC0
0	0	0	0	0	0	0	0

Bit 31-24 - Reserved

These bits are reserved for future implementation and should always be programmed to '0'.

Bit 23:0 - WBC23:0 - Working Byte Counter

These bits are loaded with the value in the Starting Transfer Counter (STC, Register (B)+44h) when the DMA transfer begins. It is decremented by 1, 2, or 4, as data is sent to the PCI Bus. This count will be decremented to '0' at the completion of a DMA transfer (reflected by the 'DONE' bit in the DMA Status Register). This register will be set to '0' with a hard reset.

6.7.5 Working Address Counter (WAC)

Address (B)+50h READ ONLY

The WAC register is a 32-bit read-only address that increments the memory address during a DMA transfer. This register's intermediate values can be read by software in between DMA burst transactions.

31	30	29	28	27	26	25	24
WAC31	WAC30	WAC29	WAC28	WAC27	WAC26	WAC25	WAC24
1	1	1	1	1	1	1	1

23	22	21	20	19	18	17	16
WAC23	WAC22	WAC21	WAC20	WAC19	WAC18	WAC17	WAC16
1	1	1	1	1	1	1	1

15	14	13	12	11	10	9	8
WAC15	WAC14	WAC13	WAC12	WAC11	WAC10	WAC9	WAC8
1	1	1	1	1	1	1	1

7	6	5	4	3	2	1	0
WAC7	WAC6	WAC5	WAC4	WAC3	WAC2	WAC1	WAC0
1	1	1	1	1	1	1	1

Bit 31:0 – WAC31:0 – Working Address Counter

This register is initialized to the value in the Starting Physical Address (SPA, Register (B)+48) Register and is incremented to the next double word (DWord) boundary when the DMA transfer begins. As data is processed by the DMA channel, the contents of this register will be incremented by DWord addresses. When the current transfer terminates, this register will reflect the address for the next access. Upon hard reset, this register is set to '1'.

6.7.6 Status Register (STATUS)
Address (B)+54h READ ONLY

The upper 3 bytes of the Status register are reserved. The state of the lower six bits report the state of the DMA channel and any termination conditions. These flags are automatically set to logic "0" when a new DMA transfer is started. Reading this register will clear it and its associated interrupt.

31	30	29	28	27	26	25	24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
X	X	X	X	X	X	X	X

23	22	21	20	19	18	17	16
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
X	X	X	X	X	X	X	X

15	14	13	12	11	10	9	8
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
X	X	X	X	X	X	X	X

7	6	5	4	3	2	1	0
Reserved	Reserved	BCMBLT	SCSINT	DONE	ABORT	ERROR	PWDN
X	X	0	0	0	0	0	X

Bit 31:8 – Reserved

These bits are reserved for future implementation.

Bit 7:6 – Reserved

These bits are reserved and read as zeros.

Bit 5 – BCMPLT – BLAST Complete

This bit, when set to '1' (true) indicates that the 'BLAST' command has completed, and the DMA F IFO is empty. This bit is only valid when the BLAST command is issued for a SCSI Disconnect and Reselect operation.

Bit 4 – SCSIINT – SCSI Interrupt

This bit when set to '1' (true) indicates that an interrupt condition has occurred in the SCSI block. This read-only bit is cleared only when the appropriate SCSI registers are serviced. The SCSI registers must be cleared by servicing the SCSI Status Register, Internal State Register and the Interrupt Status Register in the order mentioned above

Bit 3 – DONE – DMA Transfer Terminated

This bit, active logic '1' true, indicates that the DMA transfer request has successfully completed. When this bit is set, the Working Byte Count (WBC23:0) is zero, and an interrupt is generated. Reading this bit will clear it and its associated interrupt. Refer to Section 6.9 on Interrupts.

Bit 2 – ABORT – DMA Transfer Aborted

This bit, active logic '1' true, indicates that the DMA transfer request was aborted for one of the following reasons:

- the ABORT command was issued
- PCI Master Abort
- PCI Target Abort

When this bit is set, an interrupt will be generated. Reading this bit will clear it and its associated interrupt.

Bit 1 – ERROR – DMA Transfer Error

This bit, active logic '1' true, indicates that the DMA transfer request terminated with one of the error conditions present on the PCI bus. If the INTE_D bit is set in the CMD register, an interrupt will be generated. Reading this bit will clear it and its associated interrupt.

Bit 0 – PWDN – Power Down Indicator

This bit, active logic '1' true, reflects the state of the PWDN input pin. When first set, an interrupt is generated. This bit will be cleared when the PWDN pin is deasserted, however, the interrupt caused when this bit is set will be cleared when this register is read.

6.7.7 Starting Memory Descriptor List Address (SMDLA) Address (B)+58h READ/WRITE

The SMDLA register is a 32-bit read/write address that is used as the starting address of the scatter-gather Memory Descriptor List. This register is not modified by the DMA transfer logic, and may be read by the software at any time. The system software can modify this register after the DMA transfer has been started.

31	30	29	28	27	26	25	24
SMDLA31	SMDLA30	SMDLA29	SMDLA28	SMDLA27	SMDLA26	SMDLA25	SMDLA24
X	X	X	X	X	X	X	X

23	22	21	20	19	18	17	16
SMDLA23	SMDLA22	SMDLA21	SMDLA20	SMDLA19	SMDLA18	SMDLA17	SMDLA16
X	X	X	X	X	X	X	X

15	14	13	12	11	10	9	8
SMDLA15	SMDLA14	SMDLA13	SMDLA12	SMDLA11	SMDLA10	SMDLA9	SMDLA8
X	X	X	X	X	X	X	X

7	6	5	4	3	2	1	0
SMDLA7	SMDLA6	SMDLA5	SMDLA4	SMDLA3	SMDLA2	SMDLA1	SMDLA0
X	X	X	X	X	X	X	X

Bit 31:0 – SMDLA31:0 – Starting Memory Descriptor List Address

These bits reflect the starting address in the Memory Descriptor List used for scatter-gather operations. The MDL start address must be double word aligned since the hardware ignores non-zero values written to the two low address bits. Upon hard reset, this register is set to an indeterminate value.

6.7.8 Working MDL Address Counter (WMAC)
Address (B)+5Ch READ ONLY

The WMAC register is a 32-bit read-only address that is initialized to the value in the SMDLA register when the transfer begins. Its value is incremented by 4 as successive page entries in the MDL are fetched. This register will contain the address of the last MDL entry read when the transfer terminates. This register's intermediate values may be read by software between DMA burst transactions.

31	30	29	28	27	26	25	24
WMAC31	WMAC30	WMAC29	WMAC28	WMAC27	WMAC26	WMAC25	WMAC24
1	1	1	1	1	1	1	1

23	22	21	20	19	18	17	16
WMAC23	WMAC22	WMAC21	WMAC20	WMAC19	WMAC18	WMAC17	WMAC16
1	1	1	1	1	1	1	1

15	14	13	12	11	10	9	8
WMAC15	WMAC14	WMAC13	WMAC12	WMAC11	WMAC10	WMAC9	WMAC8
1	1	1	1	1	1	1	1

7	6	5	4	3	2	1	0
WMAC7	WMAC6	WMAC5	WMAC4	WMAC3	WMAC2	WMAC1	WMAC0
1	1	1	1	1	1	0	0

Bit 31:0 – WMAC31:0 – Working MDL Address Counter

Bits 31:2 are set to '1' following a hard reset, while bits 1:0 are set to '0'.

6.8 DMA SCATTER-GATHER MECHANISM

The Am53C974 contains a scatter-gather translation mechanism which facilitates faster data transfers. This feature uses a Memory Descriptor List (a list of contiguous physical memory addresses) which is stored in system memory. Use of the Memory Descriptor List allows a single SCSI transfer to be read from (or written to) non-contiguous physical memory locations. This mechanism avoids copying the transfer data and MDL list, which was previously requirėd for conventional DMA operations.

6.8.1 Memory Descriptor List (MDL)

The MDL is a non-terminated (no End Of File marker) list of 32-bit page frame addresses, which is always aligned on a Double Word boundary. The format is shown below:

31 12	11 0
Page Frame Address	Ignored

6.8.2 DMA Scatter – Gather Operation (4k aligned elements)

The scatter-gather mechanism described below assumes 4k page alignment for all MDL entries except the first and last entry. This feature is enabled by setting the MDL bit in the DMA Command register (Bit 4, Address (B)+40h).

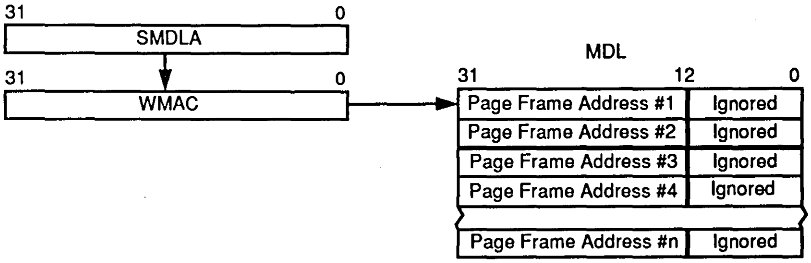
1. a) Prepare the Memory Descriptor List (MDL) through software and store it in system memory.

   b) Load the address of the into the Start Memory Descriptor List Address (SMDLA) register.

   c) Program the Starting Transfer Count (STC) register with the total transfer length (i.e., # of bytes). Also program the Starting Physical Address (SPA) register (bits 11:0) with the starting offset of the first page (Bits 31:12 in the Starting Physical Address Register will be programmed through hardware with the MDL entries).

   d) When the START DMA command is written to the DMA Command register, the value in the SMDLA register is loaded into the Working MDL Address Counter (WMAC) register which points to the appropriate entry in the MDL list as shown below.

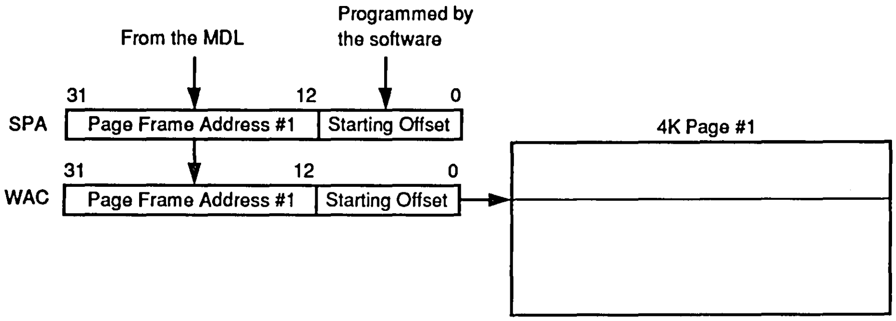
Note: The value in the SMDLA register is double word aligned. Therefore, read/write transactions will always begin on a double word boundary.



18264B-24

In this example, WMAC register is pointing to page frame address #1. When the DMA crosses the page boundary, WMAC register is incremented and points to the next page entry (page frame address #2).

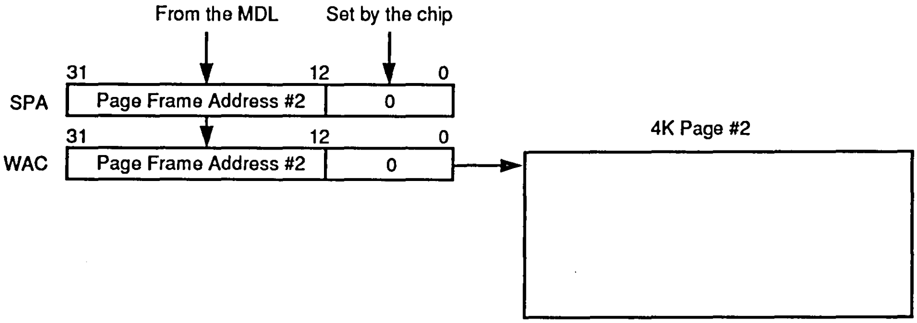
2. The Am53C974 reads only the page frame address (bits 31:12) from the MDL entry and OR's it with the first page offset value in the Starting Physical Address (SPA) register (bits 11:0). This 32-bit value is loaded into the Working Address Counter (WAC) register and becomes the physical address for page#1, as shown below.



18264B-25a

When the WAC register (bits 11:0) reaches the maximum page length, i.e., FFFh, the Am53C974 increments the WMAC register to point to the next page entry (i.e., page frame address #2 in this example) and resets the WAC register (bits 11:0) to '000h'.

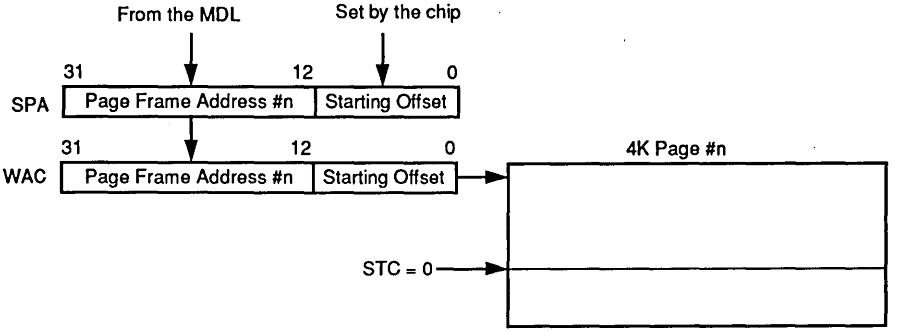
3. The chip reads the page frame address (bits 31:12) from the second MDL and OR's it with the WAC register (bits 11:0). This becomes the physical address for page#2. Since the WAC register (bits 11:0) was reset to '000h' previously (see step 2), the WAC register points at the beginning of the Page Frame Address #2 as shown below.



18264B-25

Again when WAC (bits 11:0) reaches the maximum page length count, the Am53C974 resets it to '000h' and increments the WMAC register to the next MDL entry. The operation continues in this way until WMAC register reaches the last MDL entry (Page Frame Address #n in this example).

4. The WAC register points to the beginning of the last page#n and the DMA operation continues until the byte count is exhausted in the Working Byte Counter (WBC) register. When WBC=0, the chip stops incrementing the WAC register. This is shown below.



18264B-26

Also, at WBC=0 the chip stops incrementing the WMAC register.

When a new DMA operation is initialized, the new first page offset value in the SPA register (bits 11:0) is loaded into the WAC register and DMA operation is performed following the above steps.

6.8.3 DMA Scatter – Gather Operation (Non-4k aligned elements MDL not set)

There is another way to implement a scatter-gather operation which does not force the data elements to be aligned on 4k boundaries. It assumes a "traditional" scatter-gather list of the following format:

Element 0	Physical Address Byte Count
Element 1	Physical Address Byte Count
	...
Element n	Physical Address Byte Count

This second implementation is described as follows:

1. Set the SCSI Start Transfer Count Register ((B)+00h, (B)+04h, (B)+38h) to the Byte count of the first Scatter-Gather element.
2. Program the DMA Starting Transfer Count Register ((B)+44h) to the Byte Count of the first Scatter-Gather element.
3. Program the DMA Starting Physical Address Register ((B)+48h) to the Physical Address of the first Scatter-Gather element.
4. Start the SCSI operation by issuing a SCSI Information Transfer command.
5. Start the DMA Engine with DMA Transfer Interrupt Enable (Bit 6, (B)+40h).
6. When the Scatter-Gather element's Byte Count is exhausted, the DMA engine will generate an interrupt.
7. Reprogram the next Scatter-Gather element's Byte Count into the SCSI Start Transfer Count Register and the DMA Starting Transfer Count Register.
8. Reprogram the DMA Starting Physical Address Register ((B)+48h) to the Physical Address of the next Scatter-Gather element.
9. Repeat steps 4–8 until the Scatter-Gather list is completed.

6.9 INTERRUPTS

Interrupts may come from two sources: the DMA engine or the SCSI block. Upon receipt of an interrupt ($\overline{\text{INTA}}$ asserted), the DMA Status register should be serviced first to identify the interrupt source(s). DMA engine related interrupts are cleared when the related flags are read in the DMA Status register. However, SCSI block interrupts will be cleared only when the SCSI Status, Internal State, and Interrupt Status Registers are read.

The source of the DMA interrupt will be flagged by the appropriate bit in the DMA Status Register. The reasons for interrupt are:

- Successful completion of a DMA transfer request. (Bit 6 in the DMA Command Register ((B)+40h) must be set to enable this interrupt)
- An address error occurred on the PCI bus during a DMA transfer (Bit 6 in the DMA Command Register ((B)+40h) must be set to enable this interrupt)
- The DMA transfer request was aborted (ABORT command, or Master or Target Abort)
- The PWDN pin is first asserted
- After completion of each page transfer during MDL operations. (Bit 5 in the DMA Command Register ((B)+40h) must be set to enable this interrupt)

An interrupt from the SCSI block will automatically set bit 4 (SCSIINT) in the DMA Status register (Address (B)+54h). The SCSI block will generate an interrupt under the following conditions:

- SCSI Reset occurred
- Illegal command code issued
- The target disconnects from the SCSI bus
- SCSI bus service request
- Successful completion of a command
- The Am53C974 has been reselected

CHAPTER

7 SYSTEM CONSIDERATIONS

7.1 INTRODUCTION

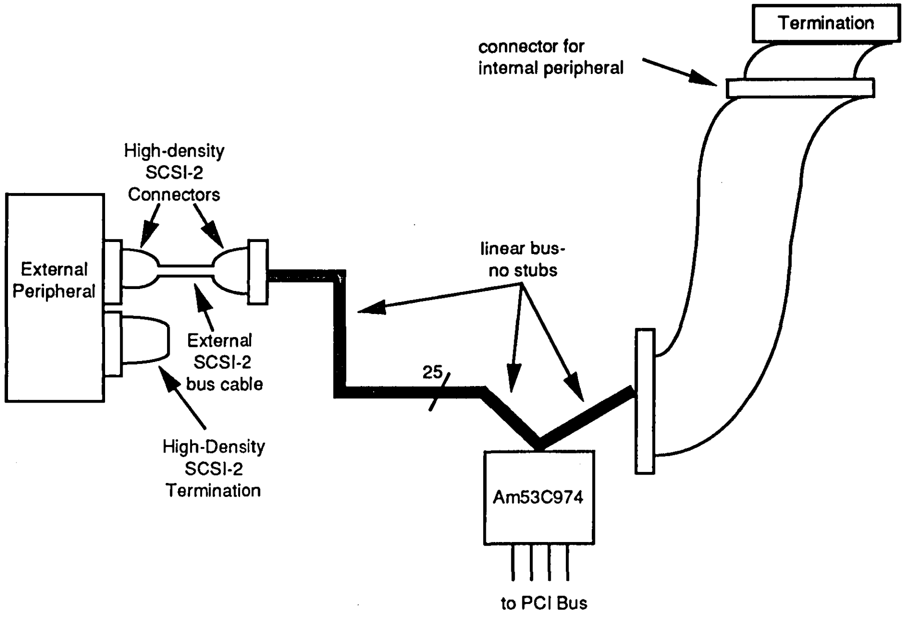
This chapter covers motherboard design considerations which use the Am53C974. It discusses SCSI component placement, signal routing, PCI interface recommendations, noise considerations and termination schemes.

7.2 SIGNAL ROUTING AND SCSI PLACEMENT

The main components for board design include the Am53C974 (whose maximum trace length from the PCI speedway is 1.5″), SCSI connectors (either one or two depending internal/external support), a 40 MHz crystal oscillator and regulated terminators (on-board) which can be up to .1 m (3.937 in) away from the Am53C974.

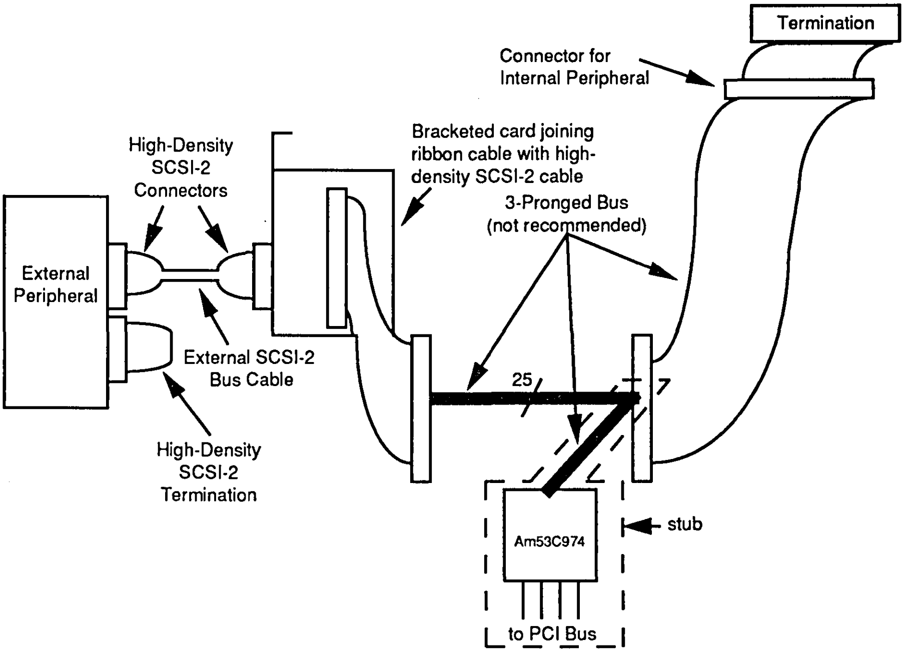
There are many ways to route SCSI bus traces on a host adapter board or motherboard. Ideally, traces from the internal SCSI bus connector, from the Am53C974 and from the external SCSI bus connector should all connect in series. Care should be taken not to have any stubs in the SCSI bus. (Stubs are any extensions off of the main bus. The maximum SCSI bus stub length allowed is .1 m). This routing scheme helps maintain signal integrity by reducing the possibility of signal reflections and other undesirable effects. Auto-routing programs used for board layout may not follow this scheme, and may create "non-ideal" environments by routing internal and external on-board connectors first instead of routing both sets of traces to the Am53C974. When peripherals are added either internally or externally, a "three-pronged" SCSI bus will be created instead of a linear one. If this or any other stub problem occurs, changes should be made manually to follow the ideal scheme. Refer to Figures 7-1 and 7-2.

Figure 7-1 Ideal Routing Scheme



18264B-27

Figure 7-2 A Poor Routing Scheme



18264B-28

7.2.1 The Motherboard

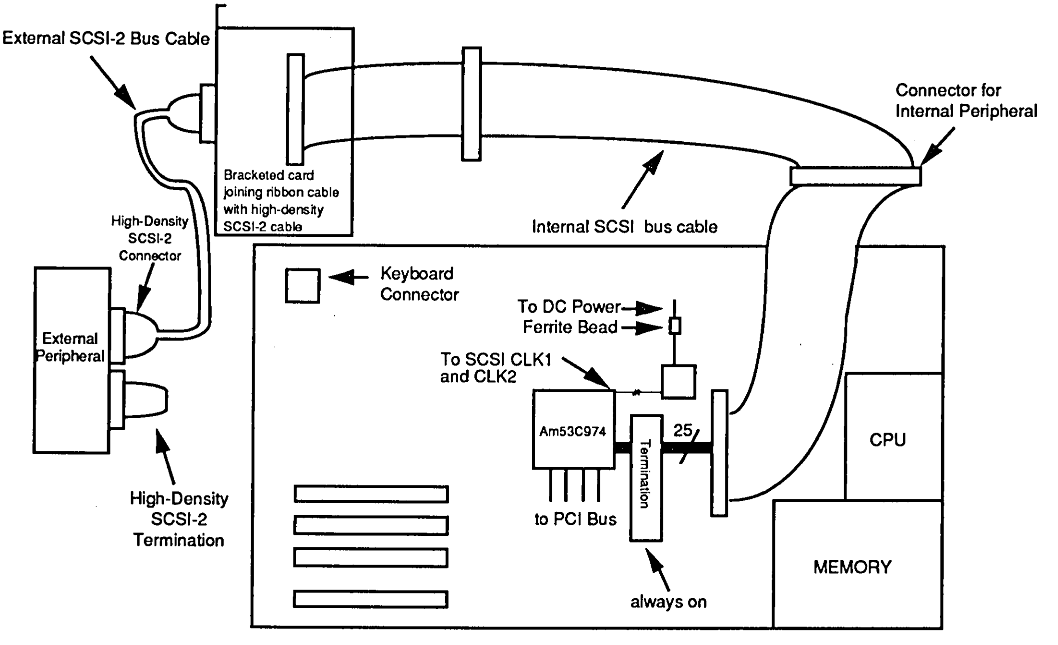
The following two layouts may be used as a guideline for the design of motherboards which incorporate the Am53C974. For both layouts, the SCSI connectors should be placed as far away from the PCI Speedway as possible. Each layout refers to termination considerations which are described in further detail in section 7.4.

7.2.1.1 Layout #1

This approach avoids the cost of placing an external connector on the motherboard. In this configuration, the Am53C974 is always at one end of the SCSI bus, therefore, the regulated terminators remain active. This eliminates the problem of switching the regulated terminators on or off to accommodate peripheral configurations. This approach also preserves the ideal linear routing scheme. The following lists the requirements for implementation, while Figure 7-3 illustrates this approach.

- One connector on the motherboard connected to one end of the internal bus ribbon cable and its components.
- The other end of the internal bus ribbon cable connected to one end of the external bus high density cable and its peripherals via a bracketed add-on card. The internal bus may also be crimped to a SCSI connector mounted on a bracket.
- On board regulated terminators a maximum of .1 m (3.937 in) from the Am53C974.
- External terminators connected to the end of the external bus.

Figure 7-3 Motherboard Layout — Approach #1


External SCSI-2 Bus Cable
Bracketed card joining ribbon cable with high-density SCSI-2 cable
Connector for Internal Peripheral
Internal SCSI bus cable
High-Density SCSI-2 Connector
External Peripheral
High-Density SCSI-2 Termination
Keyboard Connector
To DC Power
Ferrite Bead
To SCSI CLK1 and CLK2
Am53C974
Termination
25
to PCI Bus
always on
CPU
MEMORY

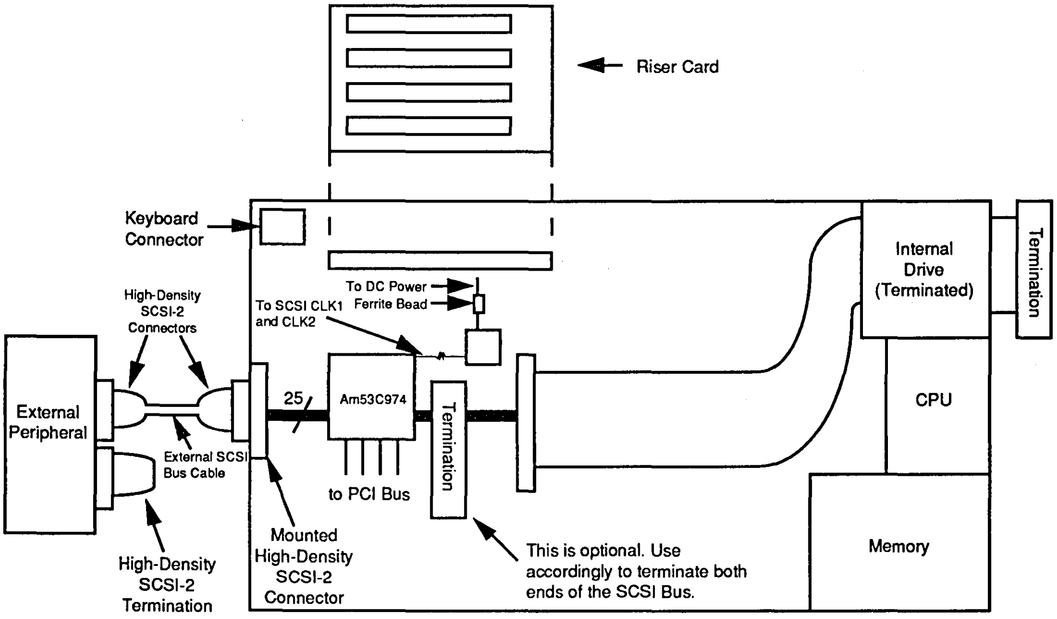
18264B-29

7.2.1.2 Layout #2

This approach uses a "pizza-box" type of motherboard, which has been incorporated into PC systems and workstations. This design reduces the system's height so that its casing resembles a pizza box. This is partly the result of a riser card that enables cards to rest on their side instead of upright. This approach requires the following and is illustrated in Figure 7-4:

- An external connector mounted on the motherboard with routings that connect to the Am53C974.
- The Am53C974 must be as close as possible to this external connector since this part of the SCSI bus consists of motherboard routings (not just ribbon cable).
- An internal connector on the motherboard to accommodate internal drives.
- On-board regulated terminators for SCSI drives not mounted within the system. However, should the user decide to connect a drive internally, terminators are not needed.
- If on-board regulated terminators are used, they should be placed within .1 m (3.937 in) from the Am53C974.

Figure 7-4 Motherboard Layout #2


Riser Card
Keyboard Connector
High-Density SCSI-2 Connectors
To DC Power
To SCSI CLK1 and CLK2
Ferrite Bead
External Peripheral
25
Am53C974
Termination
Internal Drive (Terminated)
Termination
CPU
External SCSI Bus Cable
to PCI Bus
Mounted High-Density SCSI-2 Connector
High-Density SCSI-2 Termination
This is optional. Use accordingly to terminate both ends of the SCSI Bus.
Memory
18264B-30

Motherboard designs which place an internal and external connector on either side of the Am53C974 are discouraged since:

- Two SCSI connectors are required on the motherboard—one more than what the other two approaches call for.
- When the number of internal and external bus components increase, the on-board terminators would have to be turned off either through software or hardware, which may be undesirable to a board designer.
- The possibility for creating stubs exists if the connectors and Am53C974 are not routed correctly. See Figure 7-1 for the ideal routing scheme.

7.3 NOISE CONSIDERATIONS

Some areas of a PCI motherboard or host adapter design (which include the Am53C974) are more susceptible to noise. They are:

- the 40 MHz crystal oscillator
- the SCSI cables
- the DC power planes
- pins on the ICs, specifically the Am53C974.

7.3.1 Electromagnetic Interference (EMI)

There are several ways to reduce the amount of noise present in these areas:

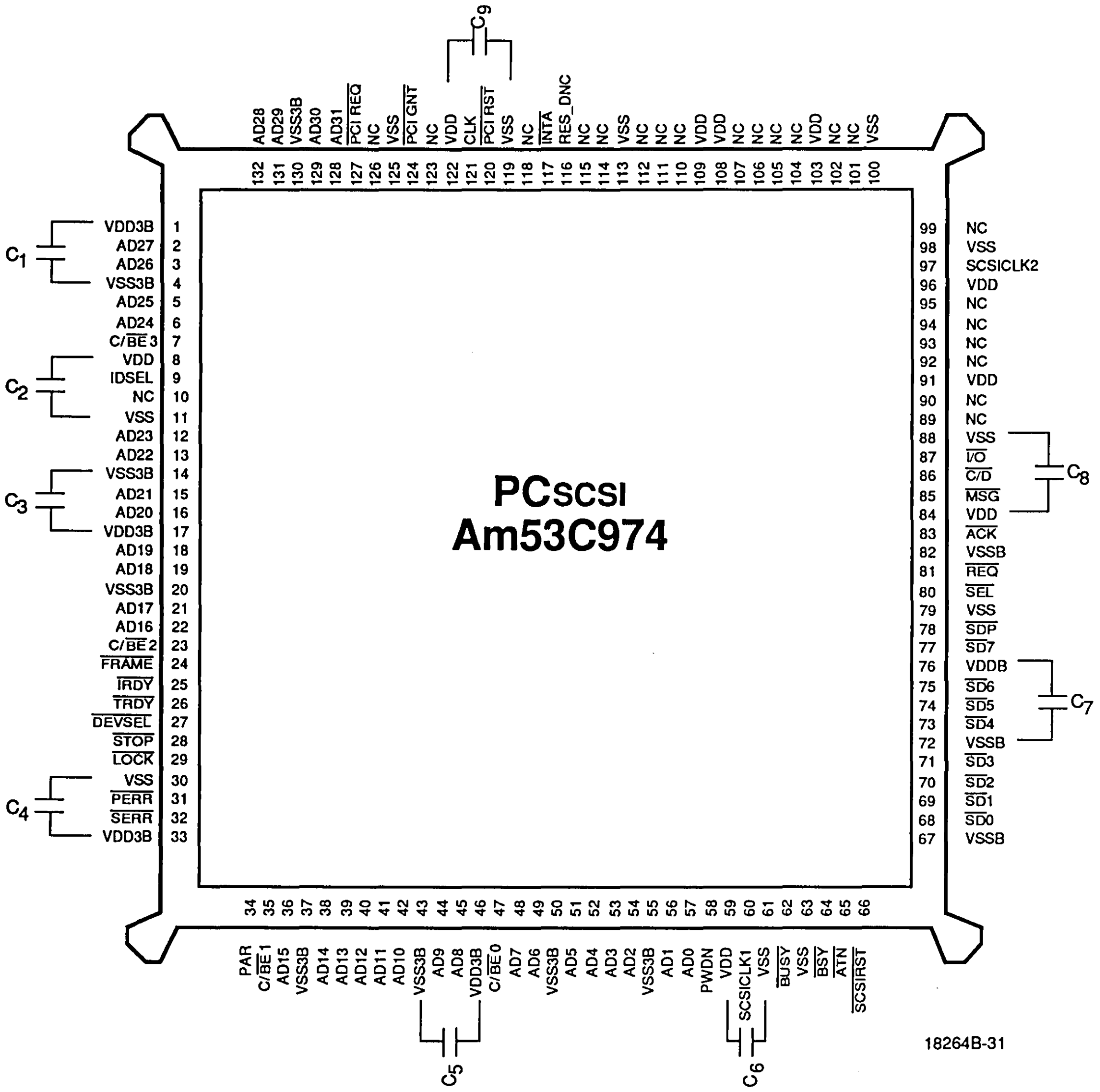
- A 40 MHz crystal oscillator must be used in order to have a 10 MB/s SCSI data rate. Use of this 40 MHz crystal oscillator, which drives the SCSI CLK1 and CLK2 pins on the Am53C974, introduces the possibility of unwanted high frequency components, including harmonics, coupling with Am53C974 signals. To help prevent this, a resistor should be placed in series with the oscillator to form a low pass filter with the input capacitance (10 pF) of the SCSI CLK1 and SCSI CLK2 pins. This filter reduces the edge rate of the clock waveform, increasing both rise and fall times by 2 ns. This removes higher frequencies, specifically harmonics from the crystal oscillator waveform and thus reduces the amount of noise introduced into the Am53C974.
- A ferrite bead, which is essentially an inductor, may be used with the oscillator to prevent coupling of higher frequencies with DC power supply signals. The ferrite bead blocks high frequency noise while acting like a short to the DC components.
- From an EMI standpoint, SCSI-2 high-density cables should be used instead of a SCSI-1 cables. Unlike the SCSI-1 flat ribbon cable, the SCSI-2 cable is electrically more substantial. It is shielded and signal wires are strategically placed for better signal travel.

7.3.2 Decoupling Methods

Decoupling capacitors should be used across the V_{DD} and V_{SS} pins on the motherboard There are pairs of V_{DD} and V_{SS} pins on the Am53C974 that should each have their own decoupling capacitor. The following decoupling method should be used for the V_{DD}/V_{SS} pairs:

- Connect the capacitor directly between a V_{DD}/V_{SS} pair of the Am53C974 so that it sits on the component side of the board. This configuration will allow the capacitor to filter undesired high frequency components directly at the Am53C974 and not only at the power planes. This not only minimizes the noise on the power planes that the chip sees, but it also filters any noise generated by the chip before it reaches the power planes. The leads to each end of the capacitor should be wide and may contain several feed-throughs to the V_{DD} and V_{SS} planes to reduce the inductance present. Refer to Figure 7-5.

Figure 7-5 Decoupling Capacitor Placement



where C1 – C9 are decoupling capacitors.

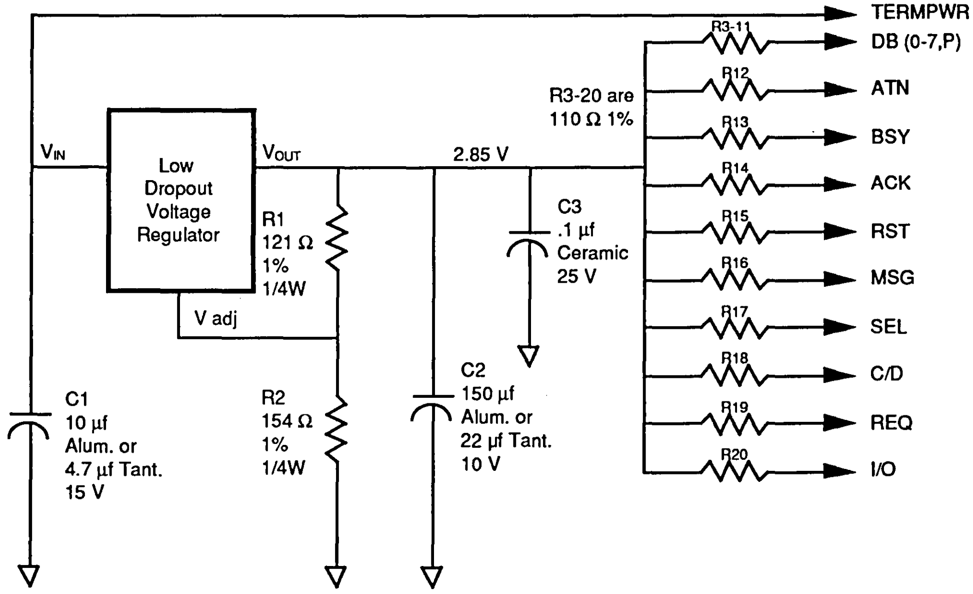
Decoupling methods which are NOT recommended include:

- connecting a capacitor only between the V_{DD} and V_{SS} planes so that it sits on the component side of the board. This doesn't allow the capacitor to reduce noise directly at the chip, but it does allow for a reduction of power plane noise and of board components. (capacitors).
- connecting a capacitor only between the V_{DD} and V_{SS} planes so that it sits on the solder side of the board. This configuration also doesn't allow direct decoupling at the chip. It does save board space, but it requires an extra manufacturing step.

7.4 TERMINATION CONSIDERATIONS

The use of active or regulated termination for terminators on the motherboard is recommended (see Figure 7-6), while external SCSI terminators can be used to terminate other parts of the SCSI bus. Terminators must match the impedance seen by a signal at the end of the SCSI bus to the characteristic impedance of the SCSI bus. This impedance is typically 84 +/– 12 Ω, but can vary greatly with PC board characteristics and cabling.

Figure 7-6 Regulated Termination



18264B-32

7.4.1 Termination

There are three general termination schemes that apply to motherboard or host adapter setups when using regulated terminators. Each scheme recognizes that the SCSI bus must be terminated on both ends. Therefore, as more components and peripherals are added to the bus, terminators must be relocated accordingly. Each scheme is also based on an ideal routing situation, that is one where the internal peripherals, external peripherals and the Am53C974 chip are connected by a linear SCSI bus. See Figure 7-1.

7.4.1.1 Scheme #1

In this case, the system uses only SCSI internal peripherals. The regulated terminators on board should be activated. Peripherals can be added to the bus by connecting them to a 50-pin ribbon cable. A SCSI terminator should be attached to the last peripheral to terminate the other end of the bus.

7.4.1.2 Scheme #2

In this case, the system uses only external SCSI peripherals. As in Scheme #1, the on-board regulated terminators should be activated. In this scheme, a 50-pin high density SCSI-2 cable should connect the external port on the motherboard or host adapter to the first external peripheral. Peripherals may be added with more cables and the last peripheral should be terminated.

7.4.1.3 Scheme #3

In this case, both internal and external SCSI peripherals are used. The regulated terminators should be deactivated (since the Am53C976 will sit in the middle of the SCSI bus). This may be accomplished through hardware or software. The hardware approach involves developing a mechanism to detect peripherals connected to the SCSI bus. This mechanism must then activate a signal to turn the regulated terminators on or off accordingly. The software approach involves having the user tell the system interactively that the regulated terminators should be turned on or off. Care should be taken to ensure that each end of the Bus is terminated.

7.5 OTHER CONSIDERATIONS

The following are motherboard considerations for routing and layout. They should be taken into account along with SCSI considerations.

7.5.1 Motherboards

1. High speed signals should be referenced only to the ground plane or exclusively to one of the power planes. If not, the power planes should be decoupled.
2. For a PCI CLK of 33 MHz, the maximum round trip time of any shared mother board signal should be 10 ns.

CHAPTER

8 AMD's PCscsi SOFTWARE

8.1 INTRODUCTION

Reduced time to market, support for all major operating systems, and the means to harness the performance and flexibility of the PCI interface — this is what AMD's PCscsi Software Solution is all about.

At the heart of this Solution is a SCSI software architecture which provides maximum flexibility in low level SCSI protocol chip software. The architecture represents a modular approach to software development and is the key factor which allows AMD to meet the aggressive time schedules set forth by our customers.

8.2 PCscsi SOFTWARE ARCHITECTURE

AMD's SCSI software architecture is divided into two layers:

a) *Hardware (H/W) dependent layer*

- provides low level programming of SCSI protocol chip.
- utilizes functions and services provided by the *Operating System dependent layer*
- independent of the operating system
- written only once for each SCSI chip

b) *Operating System (OS) dependent layer*

- acts as a manager for the *H/W dependent layer* by providing functions which the *H/W dependent layer* requires to execute desired requests
- written for specific operating system platforms
- provides DMA services, request completion, interrupt services, memory allocation, I/O port, and address translation

The operating system device drivers generate I/O requests which are translated into SCSI requests that are executed by the *O/S dependent layer.* This *O/S dependent layer* then performs any operating system dependent functions and passes the requests to the *H/W dependent layer* for hardware delivery.

AMD's PCscsi Software Architecture features:

- Device level overlapped/multi-threaded operation
- Tagged-queuing
- Automatic request sense
- Scatter-gather operation
- Synchronous Transfers (including Fast SCSI)

8.3 OPERATING SYSTEM SUPPORT

AMD PCscsi Software supports the following operating systems:

- DOS 5.0, 6.0
- Netware 3.1x, 4.x
- Windows 3.1
- OS/2 2.x
- Windows NT
- SCO Unix 3.2.4/ODT 3.0

A brief description of AMD's device drivers and support tools under the above Operating Systems are highlighted below.

8.3.1 DOS

a) Installation Program

- Installs DOS device drivers and application programs
- Update options in **config.sys and autoexec.bat** files.
- Installs appropriate command line switches for device drivers based on user's choices.
- Supports Windows 3.1 driver
- Supports for Updating installation floppies of OS/2 2.x

b) ASPI Device Driver

- Central execution point for ASPI based DOS device drivers
- Compliant with ASPI specification
- Multi-threaded execution, virtual DMA services, INT13H interceptor/handler, and Windows 3.1 support
- Manages host adapter resources

Provides hardware independent ASPI for SCSI applications and drivers

c) CD-ROM Device Driver

- Complies with Microsoft CD-ROM extensions
- Operates via ASPI interface
- Supports data and audio functions

d) Removable Media Device Driver

- Supports for greater than two fixed disks under DOS versions 3.31–4.0
- Operates via ASPI interface
- Command line switches controllable by user applications

e) SCSI Low Level format Utility

- Issues SCSI commands via ASPI to format disk media
- 'Media scan' functions which ensure media integrity
- Allows user to select SCSI options pertaining to media defect handling (e.g. automatic write reallocation, number of retries etc.)

f) **Compact Disk Audio Play Utility**

- Operates via Microsoft CD-ROM extensions to provide support for audio tracks on CD's
- Provides play, up-track, down-track, pause and stop features

8.3.2 Netware

a) **Netware ASPI**

- Netware loadable module based on the ASPI interface for Netware 386 (dated August 4, 1991)
- Calls Disks Driver for Netware
- Accepts requests from ASPI clients and Parses them
- Performs Netware specific NLM initialization
- Detects error and converts to ASPI error codes

b) **Netware Server 3.1x + /4.x**

- Based on device driver function specification for Netware operating system version 3.1x
- Netware loadable module is accessible to users can server command line
- Accepts commands from mass storage control, IOCTL interface and I/O operations interface
- Parses and Dispatches commands to appropriate command routines

8.3.3 OS/2

a) **Installation Program**

- Detects the presence of the SCSI controller.
- Utilized to access the hardware.

b) **Setup Profile**

- Describes the installation data needed by the device driver utility provided with OS/2.

c) **Adapter Device Driver**

- Complies with IBM OS/2 2.x Original Equipment Manufacturers DASD and SCSI device driver support document.

8.3.4 Windows

a) **Installation Program**

- Same DOS 5.0, 6.0 install program described previously.

b) **FastDisk Driver**

- Installed only for Windows version 3.1+
- Supports scatter-gather DMA operations.

c) **ASPI Virtual Device Driver**

- ASPI complaint Virtual Device Driver.

- Provides ASPI services to DOS based programs that are running in a DOS virtual machine within Windows 3.1+.
- Supports DOS virtual machines and multi-threaded SCSI execution.
- Communicates with DOS ASPI driver.

d) **Windows 3.1 ASPI DLL**

- ASPI compliant installable Windows device driver.
- The main function of this driver is to provide ASPI services to ASPI aware Windows applications.

e) **SCSI for Windows Utility**

- Shows status and inquiry data for all devices on the SCSI Bus

8.3.5 Windows NT

a) **Miniport**

- Miniport Driver for Windows NT for SCSI.
- Based on the port/miniport architecture

b) **Setup**

- Update file for SCSI.INF setup file for Microsoft Windows NT operating system.
- Installs the SCSI Miniport driver under Windows NT.

8.3.6 SCO UNIX

a) **Boot-Time Loadable Driver**

- Automatically loads device drivers into Unix kernel at boot time.

8.3.7 SCSI ROM BIOS

Supports INT 13h fixed disks interface to provide I/O capability to system

- 3 main modules
  - — INT13 I/O handler
  - — initialization code
  - — minimal SCSI handler
- Supports Virtual DMA services to provide compactibility with Windows 3.X enhanced mode and some DOS extenders
- Supports up to 2 Fixed Disks for DOS 5.X, 6.X
- Co-exists with other Disks Controllers (like ST506, ESDI, IDE etc.)
- Supports up to 7 Fixed Disks for DOS 5.X and non-DOS operating systems
- Developed assuming a minimum of 386 processor

8.4 PERIPHERAL SUPPORT

AMD software supports peripherals such as CD-ROMs, Tape Drives, Fixed Disk Drives, and Magneto-Optical Drives. A comprehensive list of the manufacturer names and model numbers is provided in AMD's PSSA document. Below is a brief list of supported manufacturers.

a) Fixed Disk Drives:

- Conner
- Compaq
- H-P
- IBM
- Quantum
- Seagate
- Maxtor
- Micropolis
- Fujitsu

b) CD-ROM:

- Chinon
- Panasonic
- DEC
- Denon
- Hitachi
- IBM
- LMSI
- NEC
- Toshiba
- Sony

c) Tape Drives

- Archives
- NCR
- Caliper
- Sankyo
- Cipher
- ServerDAT
- Sony
- Teac
- Emerald
- Tecmar
- Exabyte

- Tandberg
- Gigatrend
- Transitional Technologies
- H-P
- WangDAT
- IBM
- WangTek
- Maynard
- Ardat
- Mountain

d) Magneto Optical Drives:

- H-P
- Ricoh
- Sony
- Maxoptix
- Storage Dimensions

APPENDIX

A AM53C974 LITERATURE/TOOL SUPPORT

To enhance the ease of use for the Am53C974, Bus Mastering SCSI-2 Controller for PCI systems, AMD has provided the following list of literature/tool support:

- Am53C974 Data Sheet – PID #18248B
- Am53C974 Technical Manual – PID #18264B
- AMD PCscsi Software Solutions Brochure – PID #18203A
- Embedding SCSI on PCI Motherboards Brochure – PID #18179A
- Am53C974 PCscsi Product Evaluation Kit containing
  - — Hardware reference platform
  - — SCSI cable
  - — SCSI Driver licensing information
  - — Evaluation software drivers for
    - — SCSI BIOS
    - — DOS
    - — Windows
    - — Windows NT
    - — Novell
    - — SCO UNIX
    - — OS/2
  - — PCI Host Adapter Board User's Manual
  - — PCscsi Driver Installation Manual
  - — OS/Peripheral compatibility summary
  - — Am53C974 Data Sheet
  - — Am53C974 Technical Manual
  - — Performance benchmarking report
  - — PCscsi test report

To obtain any of the literature above, contact AMD literature center at (800) 222-9323 or (408) 749-5703. For more information on the Am53C974 PCscsi product evaluation kit, please contact your nearest AMD sales office.

For information on additional SCSI software products contact:

Sequoia Advanced Technologies, Inc.
(415) 459-7978
(415) 459-7988 FAX
71332,1020 CompuServe ID

For further SCSI Standards information, contact

John Lohmeyer
Chair X3T9.2
NCR Corporation
1635 Aeroplaza Drive
Colorado Springs CO 80916
(719) 573-3362

For SCSI Standard documentation, contact

Global Engineering Documents
15 Inverness Way East
Englewood, CO 80112-5704
(800) 854-7179
(303) 792-2181

Sales Offices

North American

ALABAMA	(205) 882-9122
ARIZONA	(602) 242-4400
CALIFORNIA,	
Culver City	(310) 645-1524
Newport Beach	(714) 752-6262
Sacramento(Roseville)	(916) 786-6700
San Diego	(619) 560-7030
San Jose	(408) 452-0500
Woodland Hills	(818) 878-9988
CANADA, Ontario,	
Kanata	(613) 592-0060
Willowdale	(416) 222-7800
COLORADO	(303) 741-2900
CONNECTICUT	(203) 264-7800
FLORIDA,	
Clearwater	(813) 530-9971
Boca Raton	(407) 361-0050
Orlando (Longwood)	(407) 862-9292
GEORGIA	(404) 449-7920
IDAHO	(208) 377-0393
ILLINOIS,	
Chicago (Itasca)	(708) 773-4422
Naperville	(708) 505-9517
MARYLAND	(301) 381-3790
MASSACHUSETTS	(617) 273-3970
MINNESOTA	(612) 938-0001
NEW JERSEY,	
Cherry Hill	(609) 662-2900
Parsippany	(201) 299-0002
NEW YORK,	
Brewster	(914) 279-8323
Rochester	(716) 425-8050
NORTH CAROLINA	
Charlotte	(704) 875- 3091
Raleigh	(919) 878- 8111
OHIO,	
Columbus (Westerville)	(614) 891-6455
Dayton	(513) 439-0268
OREGON	(503) 245-0080
PENNSYLVANIA	(215) 398-8006
TEXAS,	
Austin	(512) 346-7830
Dallas	(214) 934-9099
Houston	(713) 376-8084

International

BELGIUM, Antwerpen	TEL	(03) 248 43 00
	FAX	(03) 248 46 42
FRANCE, Paris	TEL	(1) 49-75-10-10
	FAX	(1) 49-75-10-13
GERMANY,		
Bad Homburg	TEL	(06172)-24061
	FAX	(06172)-23195
München	TEL	(089) 45053-0
	FAX	(089) 406490
HONG KONG,	TEL	(852) 865-4525
Wanchai	FAX	(852) 865-4335
ITALY, Milano	TEL	(02) 3390541
	FAX	(02) 38103458
JAPAN,		
Atsugi	TEL	(0462) 29-8460
	FAX	(0462) 29-8458
Kanagawa	TEL	(0462) 47-2911
	FAX	(0462) 47-1729

International *(Continued)*

Tokyo	TEL	(03) 3346-7550
	FAX	(03) 3342-5196
Osaka	TEL	(06) 243-3250
	FAX	(06) 243-3253
KOREA, Seoul	TEL	(82) 2-784-0030
	FAX	(82) 2-784-8014
LATIN AMERICA,		
Ft. Lauderdale	TEL	(305) 484-8600
	FAX	(305) 485-9736
SINGAPORE	TEL	(65) 3481188
	FAX	(65) 3480161
SWEDEN,		
Stockholm area	TEL	(08) 98 61 80
(Bromma)	FAX	(08) 98 09 06
TAIWAN, Taipei	TEL	(886) 2-7153536
	FAX	(886) 2-7122183
UNITED KINGDOM,		
Manchester area	TEL	(0925) 830380
(Warrington)	FAX	(0925) 830204
London area	TEL	(0483) 740440
(Woking)	FAX	(0483) 756196

North American Representatives

CANADA	
Burnaby, B.C. – DAVETEK MARKETING	(604) 430-3680
Kanata, Ontario – VITEL ELECTRONICS	(613) 592-0060
Mississauga, Ontario – VITEL ELECTRONICS	(416) 564-9720
Lachine, Quebec – VITEL ELECTRONICS	(514) 636-5951
ILLINOIS	
Skokie – INDUSTRIAL REPRESENTATIVES,INC	(708) 967-8430
IOWA	
LORENZ SALES	(319) 377-4666
KANSAS	
Merriam – LORENZ SALES	(913) 469-1312
Wichita – LORENZ SALES	(316) 721-0500
MICHIGAN	
Holland – COM-TEK SALES, INC	(616) 335-8418
Brighton – COM-TEK SALES, INC	(313) 227-0007
MINNESOTA	
Mel Foster Tech. Sales, Inc.	(612) 941-9790
MISSOURI	
LORENZ SALES	(314) 997-4558
NEBRASKA	
LORENZ SALES	(402) 475-4660
NEW MEXICO	
THORSON DESERT STATES	(505) 883-4343
NEW YORK	
East Syracuse – NYCOM, INC	(315) 437-8343
Hauppauge – COMPONENT CONSULTANTS, INC	(516) 273-5050
OHIO	
Centerville – DOLFUSS ROOT & CO	(513) 433-6776
Columbus – DOLFUSS ROOT & CO	(614) 885-4844
Westlake – DOLFUSS ROOT & CO	(216) 899-9370
PENNSYLVANIA	
RUSSELL F. CLARK CO.,INC.	(412) 242-9500
PUERTO RICO	
COMP REP ASSOC, INC	(809) 746-6550
UTAH	
Front Range Marketing	(801) 288-2500
WASHINGTON	
ELECTRA TECHNICAL SALES	(206) 821-7442
WISCONSIN	
Brookfield – INDUSTRIAL REPRESENTATIVES,INC	(414) 574-9393

Advanced Micro Devices reserves the right to make changes in its product without notice in order to improve design or performance characteristics. The performance characteristics listed in this document are guaranteed by specific tests, guard banding, design and other practices common to the industry. For specific testing details, contact your local AMD sales representative. The company assumes no responsibility for the use of any circuits described herein.

RECYCLED & RECYCLABLE

Advanced Micro Devices, Inc. 901 Thompson Place, P.O. Box 3453, Sunnyvale, CA 94088, USA
Tel: (408) 732-2400 • TWX: 910-339-9280 • TELEX: 34-6306 • TOLL FREE: (800) 538-8450
APPLICATIONS HOTLINE & LITERATURE ORDERING • TOLL FREE: (800) 222-9323 • (408) 749-5703

18264B 9/27/93
BAN-14M-11/93-0 Printed in USA

ADVANCED
MICRO
DEVICES, INC.
901 Thompson Place
P.O. Box 3453
Sunnyvale,
California 94088-3453
(408) 732-2400
TWX: 910-339-9280
TELEX: 34-6306

APPLICATIONS HOTLINE &
LITERATURE ORDERING
USA (800) 222-9323
USA (408) 749-5703
JAPAN 011-81-3-3346-7561
UK & EUROPE 44-(0)256-811101
TOLL FREE
USA (800) 538-8450
FRANCE 0590-8621
GERMANY 0130-813875
ITALY 1678-77224

RECYCLED &
RECYCLABLE

Printed in USA
BAN-14M-11/93-0
18264B

***ADVANCED
MICRO
DEVICES, INC.***
901 Thompson Place
P.O. Box 3453
Sunnyvale,
California 94088-3453
(408) 732-2400
TWX: 910-339-9280
TELEX: 34-6306

APPLICATIONS HOTLINE &
LITERATURE ORDERING
USA (800) 222-9323
USA (408) 749-5703
JAPAN 011-81-3-3346-7561
UK & EUROPE 44-(0)256-811101
TOLL FREE
USA (800) 538-8450
FRANCE 0590-8621
GERMANY 0130-813875
ITALY 1678-77224

RECYCLED &
RECYCLABLE

Printed in USA
BAN-14M-11/93-0
18264B