Am53C974A PCscsi™ II

Technical Manual

Revision 1.0

ADVANCED MICRO DEVICES

TABLE OF CONTENTS

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 on a PCI adapter card or 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 **S**mall **C**omputer **S**ystems **I**nterface, 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 and the specification has been finalized as of August 1990.

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 controls 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. Sixteen-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
- Fiber 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
- Fiber 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 Am53C974A 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 Am53C974A 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 Am53C974A also complies with ANSI standards X3.131–1986 (SCSI-1) and X3.131–199X (SCSI-2).

The Am53C974A offers a glueless interface to the PCI Bus, making it an ideal choice for motherboard as well as adapter card 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 Am53C974A 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
- Boot ROM support
- Level 1 SCAM support
- 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 and status bit
- Fully static design for low frequency operation
- SCSI clock disconnect capability

SCSI Bus Reliability

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

Other Features

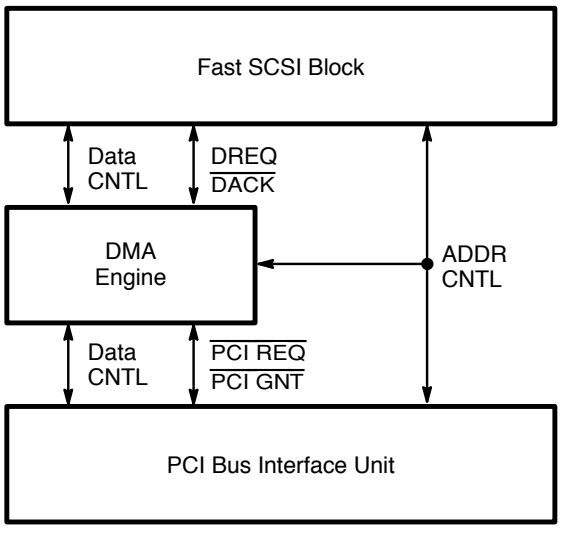
- 132-pin PQFP
- Uses state of the art advanced CMOS technology

The Am53C974A 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 Am53C974A 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 Am53C974A.

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



1.2.1 Fast SCSI Block

The Am53C974A's Fast SCSI block supports SCSI transfer rates of up to 10 MBytes/s synchronously, and up to 7 MByte/s asynchronously. The Am53C974A 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 Am53C974A has an 8-bit SCSI data interface and can operate as either an Initiator or a Target to support all SCSI applications. The SCSI block is designed to minimize host intervention by implementing common SCSI sequences in hardware. Selection, Reselection, Information Transfer and Disconnection commands are directly supported. For example, functions such as Target Selection/Initiator Reselection, Command, Message, and data transfers between the SCSI bus and the SCSI FIFO are internal processes that the Am53C974A handles without microprocessor intervention. An on-chip state machine reduces protocol overhead by performing the required sequences in response to a single command from the host.

Additionally, a 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 checking on data received from the SCSI bus is optional. 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 Am53C974A also handles without microprocessor intervention.

1.2.1.1 Features

Key features in the Am53C974A Fast SCSI block are highlighted below:

1.2.1.1.1 Access FIFO Command

The Target command set for the Am53C974A includes the Access FIFO command. This command allows the host or DMA controller to remove remaining FIFO data following the host's issuance of a Target abort DMA command, or following an abort due to parity error. This command facilitates data recovery and thereby minimizes the need to re-transmit data. For more details, refer to the Target Command Set.

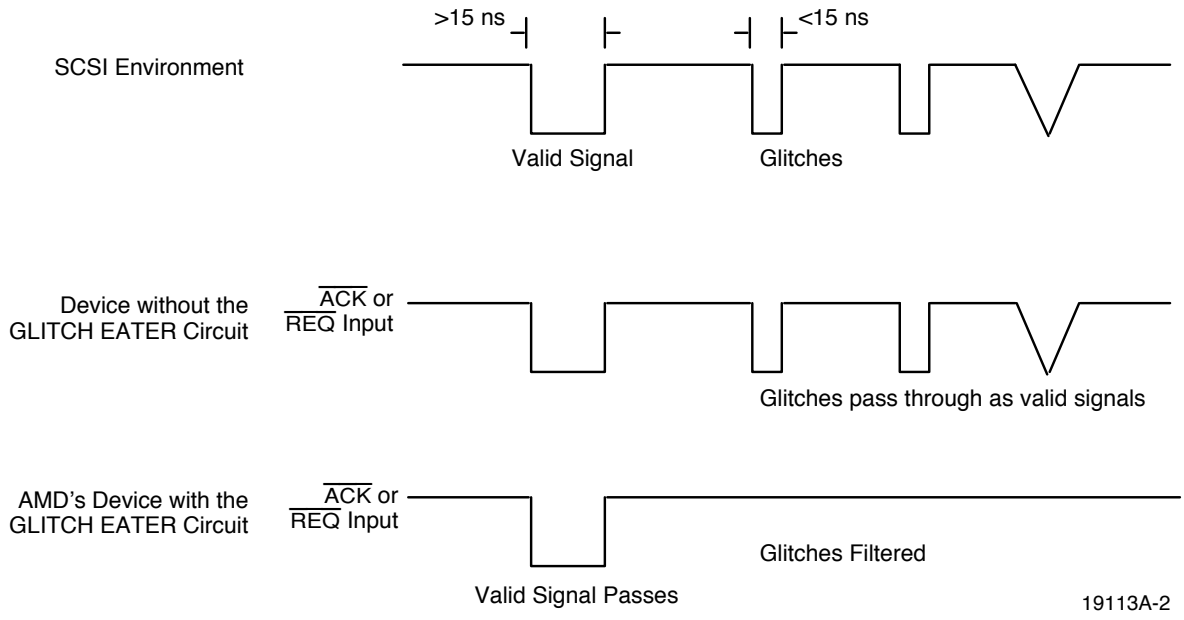
1.2.1.1.2 Reduced Power Mode

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. Additionally, the clock to the SCSI core can be disconnected for further power reduction.

1.2.1.1.3 Programmable GLITCH EATER Circuitry

The patented GLITCH EATER Circuitry in the Am53C974A PC$_{SCSI}$ II 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 $\overline{REQ}$ and $\overline{ACK}$ inputs since these lines are most susceptible to electrical anomalies such as reflections and voltage spikes. Such signal inconsistencies can trigger false $\overline{REQ}$/$\overline{ACK}$ 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



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 Am53C974A's ability to meet Fast SCSI timings.

	0 ns Window	12 ns Window
Data to $\overline{REQ}$ or $\overline{ACK}$ Setup Time:		
Fast SCSI ANSI Requirement:	25 ns	
Normal SCSI ANSI Requirement:	55 ns	
Data to $\overline{REQ}$ Setup Time (Async Initiator Receive Mode):	0 ns	12 ns
Data to $\overline{ACK}$ Setup Time (Async Target Receive Mode):	0 ns	12 ns
Data to $\overline{REQ}$ or $\overline{ACK}$ Setup Time (Sync Initiator/Target Receive Mode):	5 ns	12 ns

1.2.1.1.4 Programmable Active Negation

AMD offers programmable active negation, a feature which, when implemented, will actively drive the $\overline{REQ}$, $\overline{ACK}$ 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 $\overline{REQ}$ and $\overline{ACK}$, or on $\overline{REQ}$, $\overline{ACK}$, 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 Am53C974A 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 Am53C974A. This combination represents a powerful PCI systems solution which enhances the flexibility of your system.

1.3.1 AMD's PCscsi II Software Solution

AMD's PCscsi II 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:

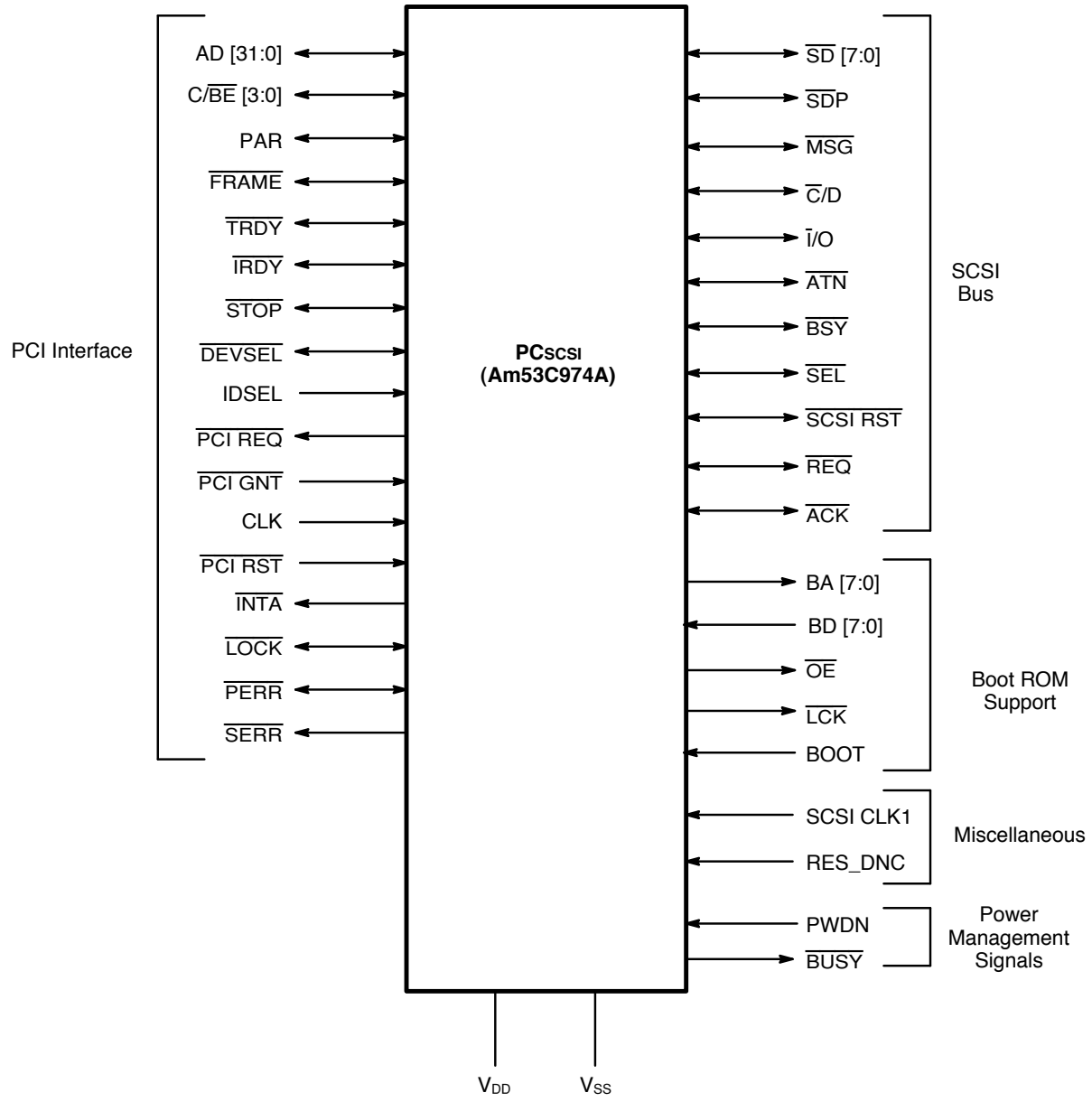
- Device level overlapped/multithreaded operation
- Tagged-queuing
- Automatic request sense
- Scatter-gather operations
- Synchronous Transfers (including Fast SCSI)

CHAPTER

2 SIGNAL DESCRIPTIONS

2.1 LOGIC SYMBOL

Figure 2-1 Am53C974A Logic Symbol



19113A-3

2.2 QUICK REFERENCE PIN DESCRIPTIONS

Pin Name	Pin Type	Description
PCI		
AD [31:00]	IN/OUT	Address/Data Bus
C/$\overline{BE}$ [3:0]	IN/OUT	Command/Byte Enable signals
PAR	IN/OUT	Parity Signal
$\overline{FRAME}$	IN/OUT	Cycle Frame
$\overline{TRDY}$	IN/OUT	Target Ready
$\overline{IRDY}$	IN/OUT	Initiator Ready
$\overline{STOP}$	IN/OUT	Stop
$\overline{LOCK}$	IN/OUT	Lock
IDSEL	IN	Initialization Device Select
$\overline{DEVSEL}$	IN/OUT	Device Select
$\overline{PCI\ REQ}$	OUT	PCI Request
$\overline{PCI\ GNT}$	IN	PCI Grant
CLK	IN	PCI Clock
$\overline{PCI\ RST}$	IN	PCI Reset
$\overline{PERR}$	IN/OUT	Parity Error
$\overline{SERR}$	OUT	System Error
$\overline{INTA}$	OUT	Interrupt
SCSI Interface		
$\overline{SD}$ [7:0]	IN/OUT	SCSI Data
$\overline{SDP}$	IN/OUT	SCSI Data Parity
$\overline{MSG}$	IN/OUT	Message
$\overline{C}$/D	IN/OUT	Command/Data
$\overline{I}$/O	IN/OUT	Input/Output
$\overline{ATN}$	IN/OUT	Attention
$\overline{BSY}$	IN/OUT	Busy
$\overline{SEL}$	IN/OUT	Select
$\overline{SCSI\ RST}$	IN/OUT	SCSI Bus Reset
$\overline{REQ}$	IN/OUT	Request
$\overline{ACK}$	IN/OUT	Acknowledge
Boot ROM Interface		
BOOT	IN	Boot ROM Enable
BA [7:0]	OUT	Boot ROM Address
BD [7:0]	IN	Boot ROM Data
$\overline{OE}$	OUT	ROM Output Enable
$\overline{LCK}$	OUT	High Address Byte Latch Clock
Miscellaneous		
SCSI CLK1	IN	SCSI Core Clock
RES_DNC	IN	Reserved, DO NOT CONNECT
Power Management		
PWDN	IN	Power Down Indicator
$\overline{BUSY}$	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)

Address and Data are multiplexed on the same PCI pins. During the first clock of a transaction the AD (31:00) contains the physical address (32 bits). During subsequent clocks AD (31:00) may contain 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)

Command and Byte Enables 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 the least significant byte (byte 0) and C/$\overline{\text{BE}}$(3) applies to the most significant byte (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 Am53C974A 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 high 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 Am53C974A 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 Am53C974A 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.

$\overline{\text{IRDY}}$
Initiator Ready (Input/Output, Active Low)

When the Am53C974A 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 Am53C974A from the initiating (master) device.

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

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

$\overline{\text{STOP}}$
Stop (Input/Output, Active Low)

In the slave role the Am53C974A drives the $\overline{\text{STOP}}$ signal to indicate to the bus master to stop the current transaction. In the bus master role the Am53C974A receives the $\overline{\text{STOP}}$ signal and stops the current transaction.

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

$\overline{\text{LOCK}}$
Lock (Input/Output, Active Low)

In the master role the Am53C974A drives the $\overline{\text{LOCK}}$ signal to indicate to the slave device that multiple transactions may be necessary to complete an operation. When $\overline{\text{LOCK}}$ is asserted, non-exclusive transactions may proceed. Control of $\overline{\text{LOCK}}$ is obtained under its own protocol in conjunction with $\overline{\text{PCI GNT}}$. In the slave role the Am53C974A receives the $\overline{\text{LOCK}}$ signal from the master.

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

***Note**: In the current implementation, the Am53C974A as a master will never generate a* $\overline{\text{LOCK}}$*. However in slave role, the chip will respond to a* $\overline{\text{LOCK}}$ *asserted by a master.*

IDSEL
Initialization Device Select (Input, Active High)

This signal is used as a chip select for the Am53C974A in lieu of the 24 address lines during configuration read and write transaction.

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

$\overline{\text{DEVSEL}}$
Device Select (Input/Output, Active Low)

This signal when actively driven by the Am53C974A 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 $\overline{\text{PCI RST}}$ is active, $\overline{\text{DEVSEL}}$ 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 Am53C974A 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 Am53C974A. 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 Am53C974A. All signals are sampled on the rising edge of CLK and all parameters are defined with respect to this edge. The Am53C974A 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 Am53C974A sequencer to a known state. All Three-State bi-directional signals are forced to a high impedance state and all Sustained Open Drain signals are allowed to float high. The Am53C974A will tristate $\overline{\text{PCI REQ}}$, and completely reset the Am53C974A. $\overline{\text{PCI RST}}$ 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)

This signal may be pulsed by the Am53C974A 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 Am53C974A 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 Am53C974A 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 ((B)+70). Interrupts caused by the DMA engine may be cleared in two ways. When the Write Erase feature is not set in the SBAC register ((B)+54), the $\overline{\text{INTA}}$ signal will be cleared when the Status Register ((B)+54) is read. When the Write Erase feature is set, the $\overline{\text{INTA}}$ signal will only be cleared when a '1' is written to the bit associated with the interrupting condition. 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, Schmitt Trigger, Open Drain/Active Negation)

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

$\overline{\text{SDP}}$
SCSI Data Parity
(Input/Output, Active Low, Schmitt Trigger, Open Drain/Active Negation)

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

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

$\overline{\text{MSG}}$ is a bi-directional signal which is asserted during a MESSAGE phase. It is a schmitt triggered input in the Initiator role and an output with a 48 mA driver in the Target role.

$\overline{\text{C}}$/D
Command/Data (Input/Output, Active Low, Schmitt Trigger, Open Drain)

$\overline{\text{C}}$/D is a bi-directional signal which is used to indicate whether CONTROL or DATA information is on the SCSI data bus. It is a schmitt trigger input in the Initiator role and an output with a 48 mA driver in the Target role.

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

$\overline{\text{I}}$/O is a bi-directional signal which controls the direction of data movement on the SCSI data bus with respect to the initiator. It is a schmitt triggered input in the Initiator role and an output with a 48 mA driver in the Target role.

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

$\overline{\text{ATN}}$ is a bi-directional signal which is used to indicate the ATTENTION condition. It is a schmitt triggered input in the Target role and an output with a 48 mA driver in the Initiator role.

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

$\overline{\text{BSY}}$ is a bi-directional signal which is asserted when the Am53C974A is arbitrating for the SCSI bus or when it is connected as a Target. As an input it has a schmitt trigger and as an output it has a 48 mA driver.

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

$\overline{\text{SEL}}$ is a bi-directional signal which is asserted when the Am53C974A is attempting to select or reselect another SCSI device. As an input it has a schmitt trigger and as an output it has a 48 mA driver.

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

$\overline{\text{SCSI RST}}$ is a bi-directional SCSI bus reset signal. As a schmitt triggered input to the Am53C974A, this signal when asserted will reset portions of the SCSI logic (see Soft Reset). As a 48 mA output driver, this signal when asserted will cause all other devices on the SCSI bus to be reset. The Reset SCSI command will also cause the $\overline{\text{SCSI RST}}$ pin to be driven active for 25–40 ms, depending on the SCSI clock frequency and conversion factor.

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

$\overline{\text{REQ}}$ is a bi-directional SCSI bus signal which indicates a request for a $\overline{\text{REQ/ACK}}$ data transfer. It is a schmitt triggered input in the Initiator role and an output with a 48 mA driver in the Target role.

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

$\overline{\text{ACK}}$ is a bi-directional SCSI bus signal which is used to indicate an acknowledgment for a $\overline{\text{REQ/ACK}}$ data transfer handshake. It is a schmitt triggered input in the Target role and a output with a 48 mA driver in the Initiator role.

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 $\overline{\text{BSY}}$ ORed with $\overline{\text{SEL}}$ signals. 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 Boot ROM Support Pins

BOOT
Boot ROM Present (Input)

The state of this pin determines whether or not the Boot ROM interface on the Am53C974A is enabled. When this pin is connected to V_{CC}, the interface is enabled to support the Boot ROM feature. When this pin is connected to ground, all input buffers on BD (7:0) are disabled, and BA (7:0), $\overline{\text{OE}}$, and $\overline{\text{LCK}}$ pins are tri-stated. Since this pin was V_{SS} on the Am53C974, the Boot ROM interface on the Am53C974A is automatically disabled in existing Am53C974 designs.

BD (7:0)
Boot ROM Data (Input)

When the Am53C974A is configured for Boot ROM support (Pin 100 tied to V_{CC}), BD (7:0) carries data from the ROM to the Am53C974A. When not configured to support a Boot ROM (Pin 100 tied to ground), the input buffers on these pins are disabled.

BA (7:0)
ROM Address (Output)

When the Am53C974A is configured for Boot ROM support (Pin 100 tied to V_{CC}), BA (7:0) carries the Boot ROM address from the Am53C974A. When Boot ROM support is disabled (Pin 100 tied to ground), these pins are tri-stated.

$\overline{\text{OE}}$
ROM Output Enable (Output)

When the Am53C974A is configured for Boot ROM support (Pin 100 tied to V_{CC}), this pin is used as the output enable for the ROM. When Boot ROM support is disabled (Pin 100 Tied to ground), this pin is tri-stated.

$\overline{\text{LCK}}$
Latch Clock (Output)

When the Am53C974A is configured for Boot ROM support (Pin 100 tied to V_{CC}), this pin is used as a clock to latch the high byte of the ROM address. When Boot ROM support is disabled (Pin 100 tied to ground), this pin is tri-stated.

2.3.10 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.

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.**

2.3.11 Power Supply Pins

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

These inputs provide power necessary to operate the Am53C974A. 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 Am53C974A. 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	BOOT
2	AD27	35	C/$\overline{BE}$1	68	$\overline{SD}$0	101	BD0
3	AD26	36	AD15	69	$\overline{SD}$1	102	NC
4	V_{SS3B}	37	V_{SS3B}	70	$\overline{SD}$2	103	V_{DD}
5	AD25	38	AD14	71	$\overline{SD}$3	104	$\overline{OE}$
6	AD24	39	AD13	72	V_{SSB}	105	$\overline{LCK}$
7	C/$\overline{BE}$3	40	AD12	73	$\overline{SD}$4	106	BA0
8	V_{DD}	41	AD11	74	$\overline{SD}$5	107	NC
9	IDSEL	42	AD10	75	$\overline{SD}$6	108	V_{DD}
10	NC	43	V_{SS3B}	76	V_{DDB}	109	V_{DD}
11	V_{SS}	44	AD9	77	$\overline{SD}$7	110	NC
12	AD23	45	AD8	78	$\overline{SDP}$	111	BA1
13	AD22	46	V_{DD3B}	79	V_{SS}	112	BA2
14	V_{SS3B}	47	C/$\overline{BE}$0	80	$\overline{SEL}$	113	V_{SS}
15	AD21	48	AD7	81	$\overline{REQ}$	114	BA3
16	AD20	49	AD6	82	V_{SSB}	115	BA4
17	V_{DD3B}	50	V_{SS3B}	83	$\overline{ACK}$	116	RES_DNC
18	AD19	51	AD5	84	V_{DD}	117	$\overline{INTA}$
19	AD18	52	AD4	85	$\overline{MSG}$	118	BA5
20	V_{SS3B}	53	AD3	86	$\overline{C}$/D	119	V_{SS}
21	AD17	54	AD2	87	I/O	120	$\overline{PCI\ RST}$
22	AD16	55	V_{SS3B}	88	V_{SS}	121	CLK
23	C/$\overline{BE}$2	56	AD1	89	BD7	122	V_{DD}
24	$\overline{FRAME}$	57	AD0	90	BD6	123	BA6
25	$\overline{IRDY}$	58	PWDN	91	V_{DD}	124	$\overline{PCI\ GNT}$
26	$\overline{TRDY}$	59	V_{DD}	92	NC	125	V_{SS}
27	$\overline{DEVSEL}$	60	SCSICLK1	93	BD5	126	BA7
28	$\overline{STOP}$	61	V_{SS}	94	BD4	127	$\overline{PCI\ REQ}$
29	$\overline{LOCK}$	62	$\overline{BUSY}$	95	BD3	128	AD31
30	V_{SS}	63	V_{SS}	96	V_{DD}	129	AD30
31	$\overline{PERR}$	64	$\overline{BSY}$	97	BD2	130	V_{SS3B}
32	$\overline{SERR}$	65	$\overline{ATN}$	98	V_{SS}	131	AD29
33	V_{DD3B}	66	$\overline{SCSI\ RST}$	99	BD1	132	AD28

NC = No Connect

RES_DNC = Reserved_DO NOT CONNECT.

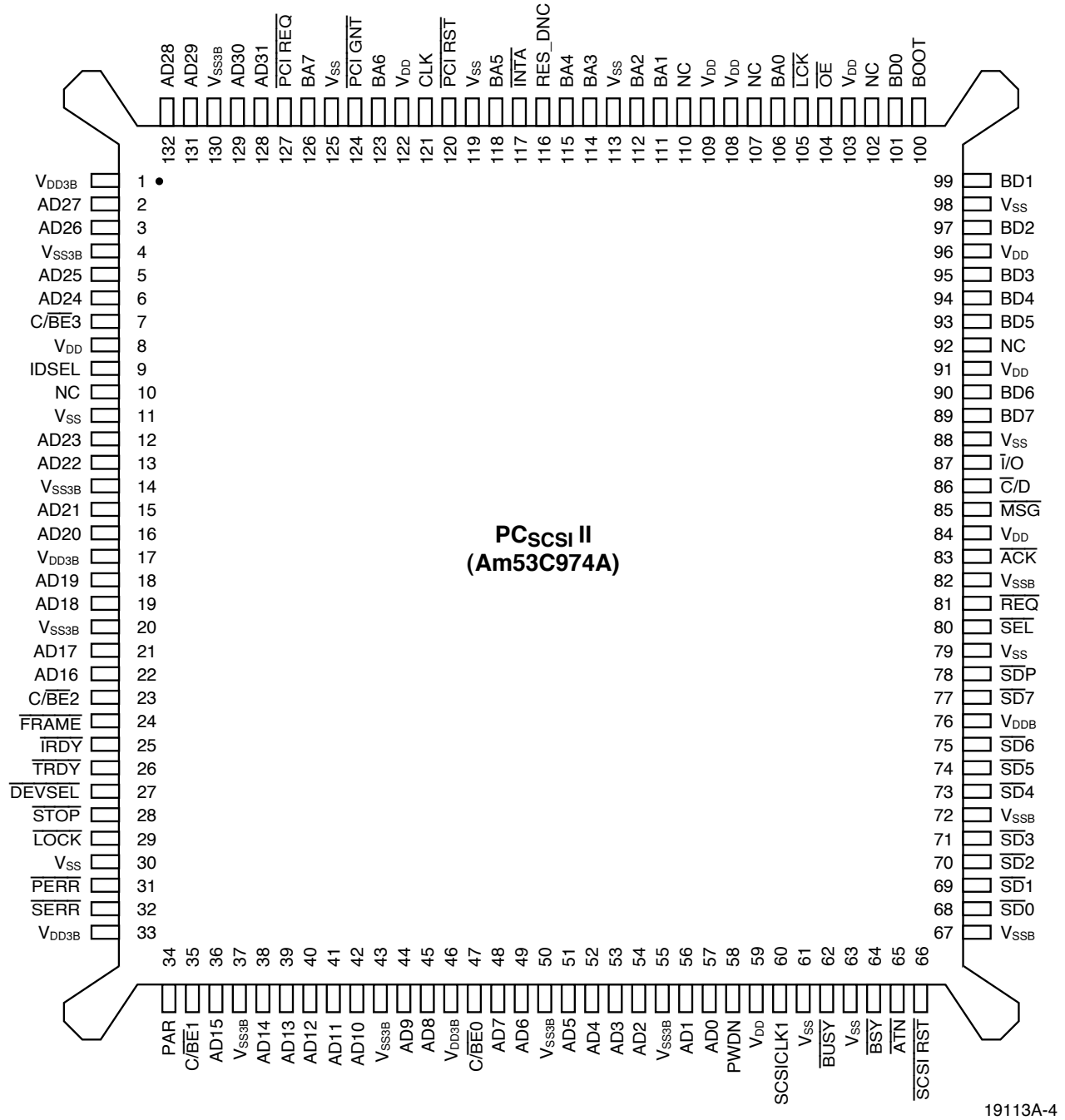
2.4.2 Listed by Pin Name

Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.
$\overline{ACK}$	83	$\overline{ATN}$	65	$\overline{LOCK}$	29	V_{DD}	84
AD0	57	BA0	106	$\overline{MSG}$	85	V_{DD}	91
AD1	56	BA1	111	NC	10	V_{DD}	96
AD2	54	BA2	112	NC	92	V_{DD}	103
AD3	53	BA3	114	NC	102	V_{DD}	108
AD4	52	BA4	115	NC	107	V_{DD}	109
AD5	51	BA5	118	NC	110	V_{DD}	122
AD6	49	BA6	123	$\overline{OE}$	104	V_{DD3B}	1
AD7	48	BA7	126	PAR	34	V_{DD3B}	17
AD8	45	BD0	101	$\overline{PCI\ GNT}$	124	V_{DD3B}	33
AD9	44	BD1	99	$\overline{PCI\ REQ}$	127	V_{DD3B}	46
AD10	42	BD2	97	$\overline{PCI\ RST}$	120	V_{DDB}	76
AD11	41	BD3	95	$\overline{PERR}$	31	V_{SS}	11
AD12	40	BD4	94	PWDN	58	V_{SS}	30
AD13	39	BD5	93	$\overline{REQ}$	81	V_{SS}	61
AD14	38	BD6	90	RES_DNC	116	V_{SS}	63
AD15	36	BD7	89	$\overline{SCSI\ RST}$	66	V_{SS}	79
AD16	22	BOOT	100	SCSICLK1	60	V_{SS}	88
AD17	21	$\overline{BSY}$	64	$\overline{SD0}$	68	V_{SS}	98
AD18	19	$\overline{BUSY}$	62	$\overline{SD1}$	69	V_{SS}	113
AD19	18	C/$\overline{BE0}$	47	$\overline{SD2}$	70	V_{SS}	119
AD20	16	C/$\overline{BE1}$	35	$\overline{SD3}$	71	V_{SS}	125
AD21	15	C/$\overline{BE2}$	23	$\overline{SD4}$	73	V_{SS3B}	4
AD22	13	C/$\overline{BE3}$	7	$\overline{SD5}$	74	V_{SS3B}	14
AD23	12	$\overline{C}$/D	86	$\overline{SD6}$	75	V_{SS3B}	20
AD24	6	CLK	121	$\overline{SD7}$	77	V_{SS3B}	37
AD25	5	$\overline{DEVSEL}$	27	$\overline{SDP}$	78	V_{SS3B}	43
AD26	3	$\overline{FRAME}$	24	$\overline{SEL}$	80	V_{SS3B}	50
AD27	2	$\overline{I}$/O	87	$\overline{SERR}$	32	V_{SS3B}	55
AD28	132	IDSEL	9	$\overline{STOP}$	28	V_{SS3B}	130
AD29	131	$\overline{INTA}$	117	$\overline{TRDY}$	26	V_{SSB}	67
AD30	129	$\overline{IRDY}$	25	V_{DD}	8	V_{SSB}	72
AD31	128	$\overline{LCK}$	105	V_{DD}	59	V_{SSB}	82

NC = No Connect

RES_DNC = Reserved_DO NOT CONNECT.

2.5 PIN OUT MAP



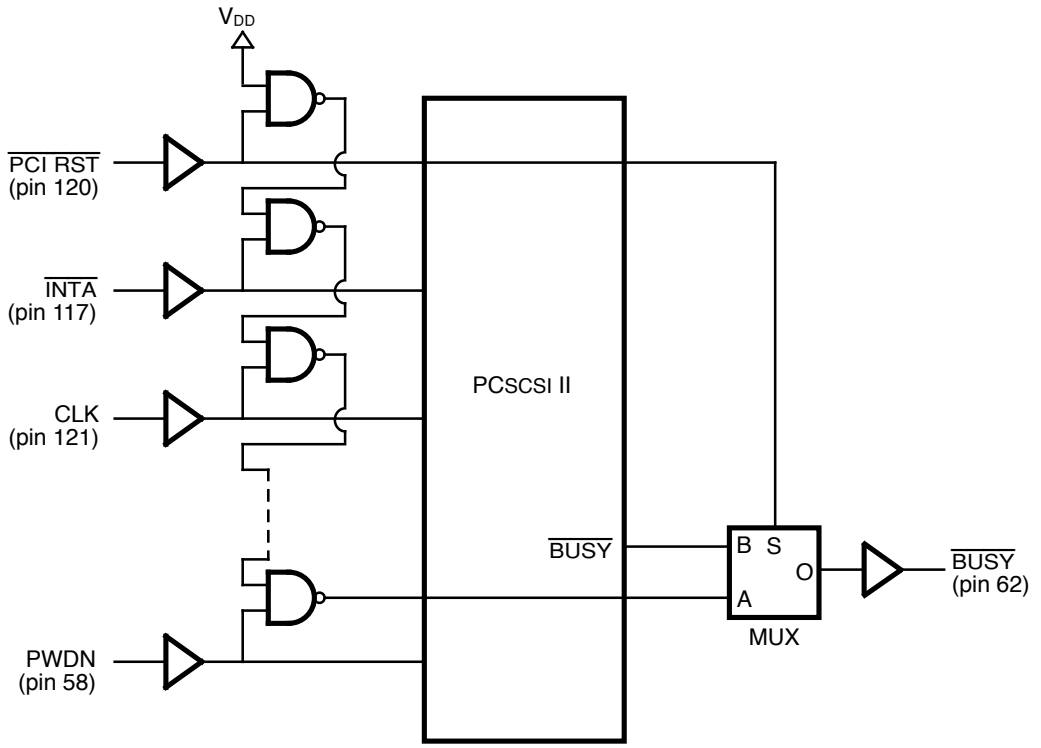
19113A-4

2.6 NAND TREE TESTING

The Am53C974A PCSCSI II 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



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, counter-clockwise, 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.

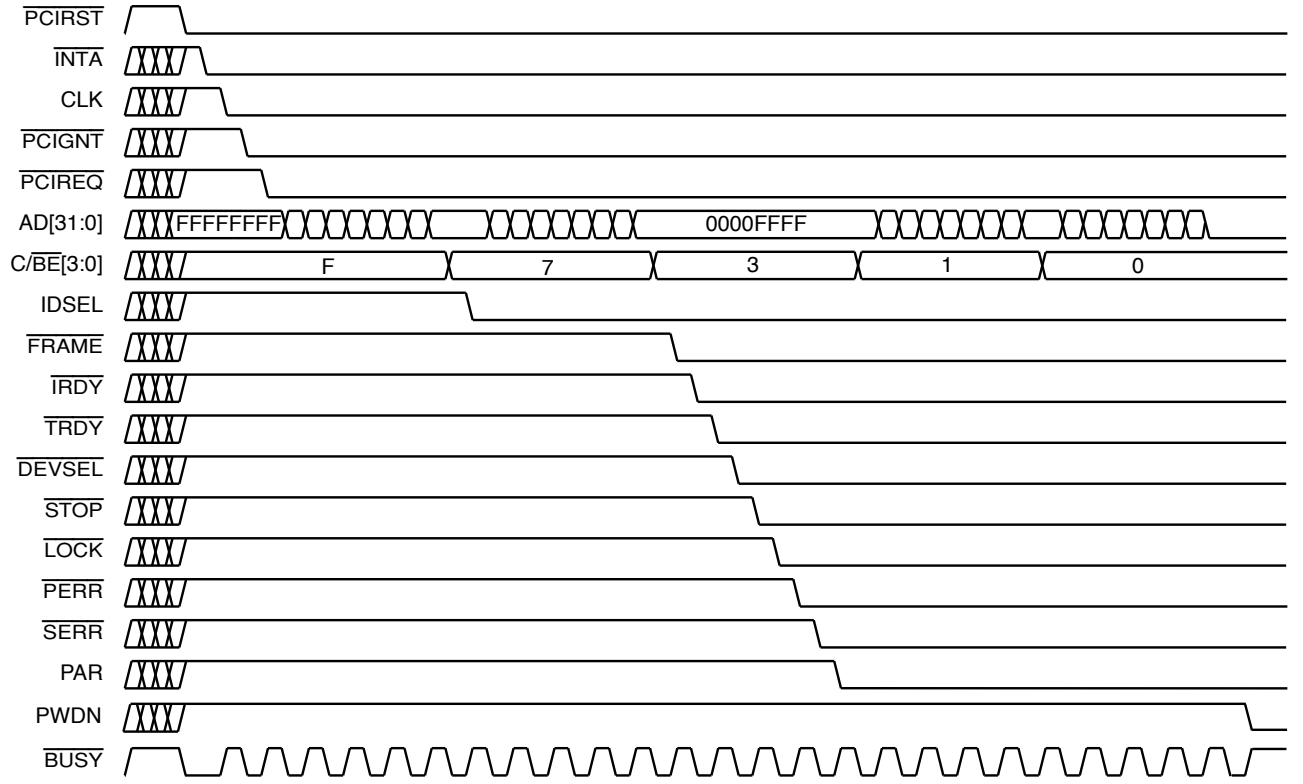
NAND Tree Input #	Pin #	Name	NAND Tree Input #	Pin #	Name	NAND Tree Input #	Pin #	Name
1	120	$\overline{\text{PCI RST}}$	19	16	AD20	37	39	AD13
2	117	$\overline{\text{INTA}}$	20	18	AD19	38	40	AD12
3	121	CLK	21	19	AD18	39	41	AD11
4	124	$\overline{\text{PCI GNT}}$	22	21	AD17	40	42	AD10
5	127	$\overline{\text{PCI REQ}}$	23	22	AD16	41	44	AD9
6	128	AD31	24	23	C/$\overline{\text{BE}}$2	42	45	AD8
7	129	AD30	25	24	$\overline{\text{FRAME}}$	43	47	C/$\overline{\text{BE}}$0
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{BE}}$3	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{BE}}$1	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 high 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 PC$_{SCSI}$ controller is configured for NAND tree testing.*

Figure 2-3 NAND Tree Waveform



19113A-6

2.7 Am53C974A REGISTER MAP

Configuration Register Map

31	16	15	0	
Device ID		Vendor ID		00h
Status		Command		04h
Base Class	Sub Class	Prog. If.	Revision ID	08h
BIST*	Header Type*	Latency Timer	Cache Line Size*	0Ch
Base Address				10h
Reserved*				14h – 38h
Expansion ROM Base Address				30h
Reserved*				34h – 38h
Max_Lat	Min_Gnt	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 Am53C974A. 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	R
STCREG	(B)+00	Start Transfer Count Register Low	W
CTCREG	(B)+04	Current Transfer Count Register Middle	R
STCREG	(B)+04	Start Transfer Count Register Middle	W
FFREG	(B)+08	SCSI FIFO Register	R/W
CMDREG	(B)+0C	SCSI Command Register	R/W
STATREG	(B)+10	SCSI Status Register	R
SDIDREG	(B)+10	SCSI Destination ID Register	W
INSTREG	(B)+14	Interrupt Status Register	R
STIMREG	(B)+14	SCSI Timeout Register	W
ISREG	(B)+18	Internal State Register	R
STPREG	(B)+18	Synchronous Transfer Period Register	W
CFIREG	(B)+1C	Current FIFO/Internal State Register	R
SOFREG1	(B)+1C	Synchronous Offset Register	W
CNTLREG1	(B)+20	Control Register One	R/W
CLKFREG	(B)+24	Clock Factor Register	W
RES	(B)+28	Reserved	W
CNTLREG2	(B)+2C	Control Register Two	R/W
CNTLREG3	(B)+30	Control Register Three	R/W
CNTLREG4	(B)+34	Control Register Four	R/W
CTCREG	(B)+38	Current Transfer Count Register High/Part-Unique ID Code	R
STCREG	(B)+38	Start Current Transfer Count Register High	W
RES	(B)+3C	Reserved	W

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
SBAC	(B)+70	SCSI Bus and Control	*

**Certain bits are Read/Write, certain bits are Read Only. Refer to the SBAC (SCSI Bus and Control) register for more detail.*

CHAPTER

3 POWER MANAGEMENT FEATURES

3.1 INTRODUCTION

As a leader in low-voltage technology, AMD has incorporated power-saving features into the Am53C974A. Through hardware and software or just software alone, the Am53C974A 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 INDICATORS

The SCSI Bus activity can be monitored through hardware or software. Through the hardware, the SCSI Bus activity is reflected by the $\overline{\text{BUSY}}$ output pin. This pin, when active, indicates that the SCSI Bus is in use and therefore the Am53C974A should not be powered down. Similarly, the SCSI activity can also be monitored through software by polling the SBSY bit (bit 20) in the SBAC register ((B)+70). Both these indicators are the logical equivalent to the SCSI bus signal $\overline{\text{BSY}}$ ORed with $\overline{\text{SEL}}$. However, they are not physically connected to the $\overline{\text{BSY}}$ signal on the SCSI bus. To correctly identify the Bus Free State on the SCSI bus, either the $\overline{\text{BUSY}}$ pin or the SBSY bit must be inactive for at least 250 ms (Selection Timeout period). Once this condition is valid, the SCSI software can safely commence the power down sequence. Note that if the $\overline{\text{BUSY}}$ pin is used to detect SCSI bus free condition, then external logic on the host must drive the PWDN pin to notify the SCSI software to commence the power down sequence.

3.2.1 Reduced Power Mode

When the SCSI Bus is free and there are no pending commands, the Am53C974A may be powered down by turning off the input buffers on the SCSI Bus lines. This is done by setting bit 5 in Control Register Four (B)+34h. Additionally, for further power reduction, the internal registers may be programmed to a predetermined state, and the clock to the SCSI core disconnected via the PWD bit (bit 21) in the SBAC register ((B)+70). However before disconnecting the clock from the SCSI core, the state of the SCSI bus should first be saved. This can be done through use of the Scratch registers in the PCI configuration space starting at address 40h.

3.3 POWER DOWN PIN (PWDN Pin)

When the PWDN pin is driven active, it sets the PWDN bit in the Status Register (Bit 0, DMA Status Register (B)+54h), signaling the Am53C974A that the host would like to power down the SCSI interface. An interrupt is generated when this bit is 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 SCSI input buffers as described in the previous section.*

CHAPTER

4 THE PCI BUS INTERFACE UNIT

4.1 INTRODUCTION

The Am53C974A 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 Am53C974A 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 Am53C974A PCI_{SCSI} II controller. The Am53C974A'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*.

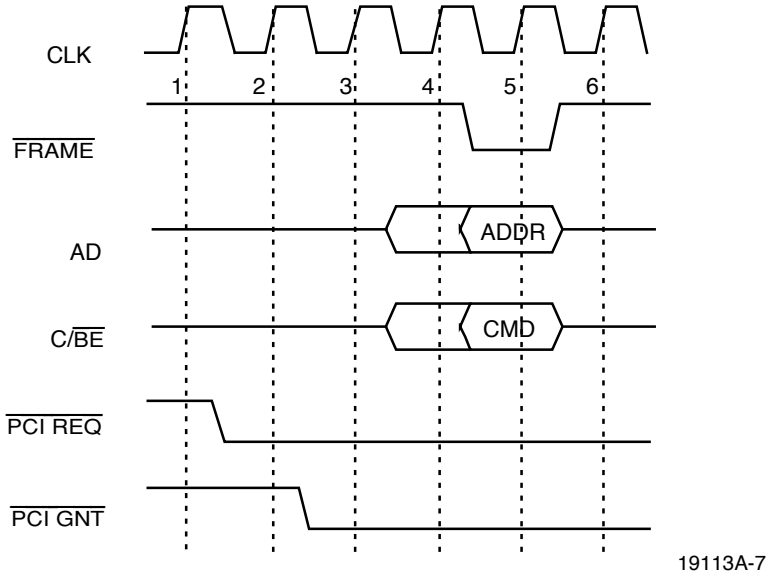
4.2 ADDRESSING

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. Am53C974A 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 base I/O location of the Am53C974A. Configuration accesses to the Am53C974A are done by issuing a Configuration Read/Write command with the Am53C974A IDSEL line asserted.

4.3 BUS ACQUISITION

The first step in any Am53C974A 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 Am53C974A's bus acquisition timing. In this figure, although bus ownership is granted on clock 3 with the assertion of $\overline{\text{PCI GNT}}$, the Am53C974A will not assert $\overline{\text{FRAME}}$ (indicating the start of bus cycle) until clock 5. Note, however, that although the Am53C974A 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 Am53C974A 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 Am53C974A supports only the relevant PCI bus cycles (eight of the sixteen cycles). These cycles are defined by the C/$\overline{BE}$ [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 Am53C974A. Note that Bus cycles with an asterisk (*) are ignored by the Am53C974A, while those with double asterisks (**) are aliased to the Slave Memory read cycle.

Table 4-1 PCI Bus Cycles Supported by the Am53C974A

C/$\overline{BE}$[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, Slave
0111	Memory Write	Master, Slave***
1000	Reserved	*
1001	Reserved	*
1010	Config. Read	Slave
1011	Config. Write	Slave
1100	Mem Read Multiple	**Slave
1101	Dual Address Cycle	*
1110	Mem Read Line	Master, **Slave
1111	Mem Write & Invalidate	*

** These cycles are ignored by the Am53C974A.*

*** Both the Slave Memory Read Line and Slave Memory Read Multiple Cycles are aliased to the Slave Memory Read cycle.*

**** Slave Memory Write commands to the Am53C974A will complete normally but the data is ignored by the device.*

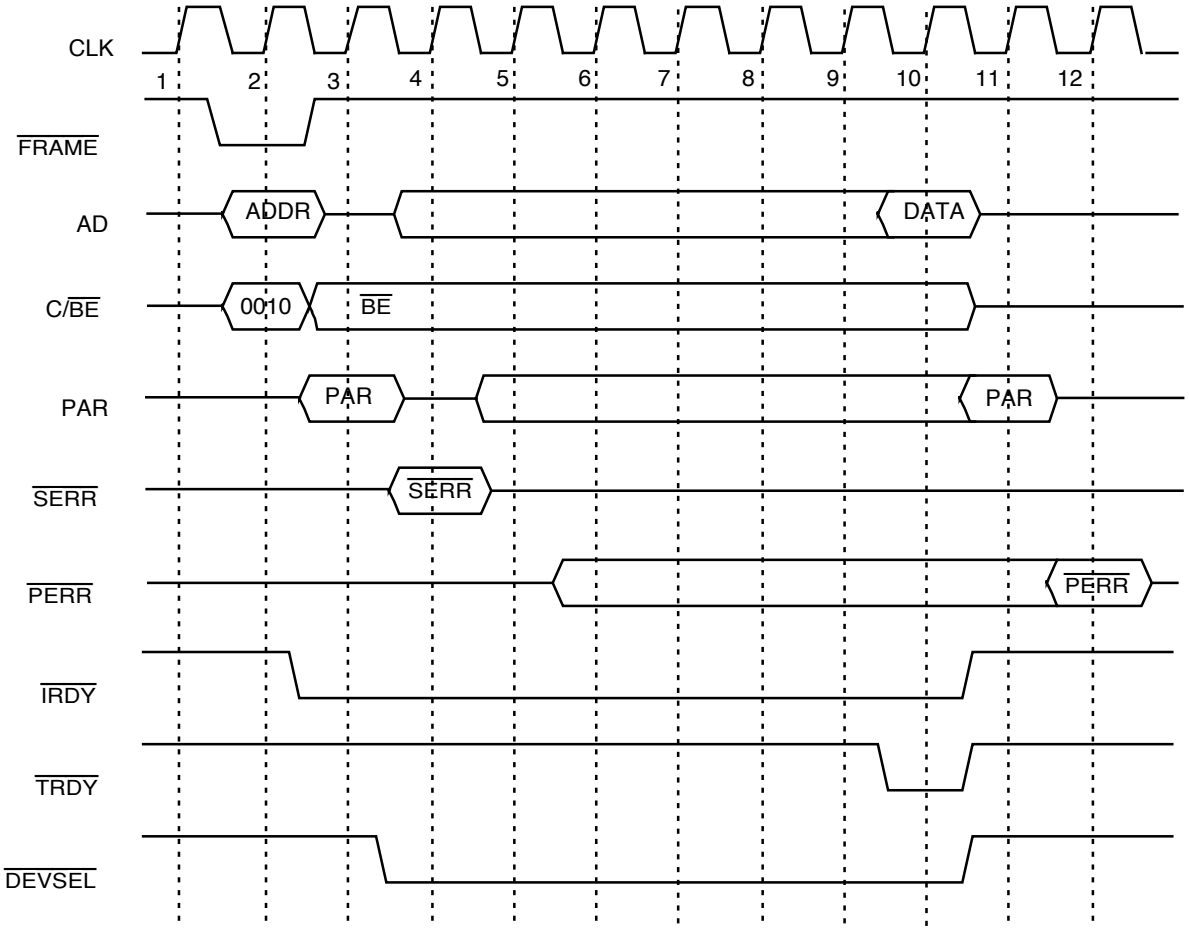
4.5 BUS CYCLE DIAGRAMS

The following are samples of the Am53C974A'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 Am53C974A. It is a single cycle, non_burst 8-bit, 16-bit, or 32-bit transfer which is initiated by the host CPU. The Am53C974A will not produce slave I/O Read commands while a bus master. Slave I/O Read cycles are fixed length cycles, i.e. the Am53C974A 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

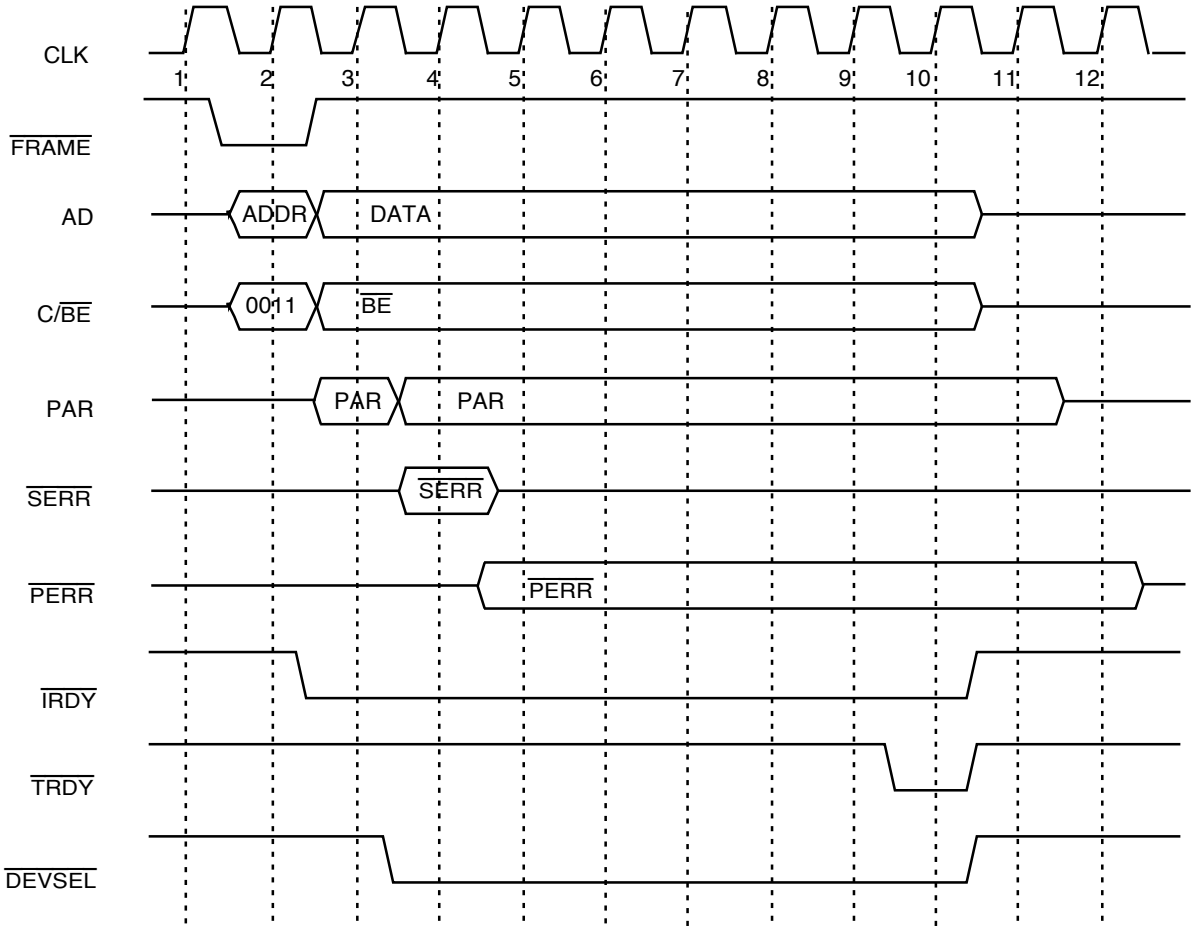


19113A-8

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 Am53C974A. It is a single cycle, non-burst 8-bit, 16-bit, or 32-bit transfer which is initiated by the host CPU. The Am53C974A will not produce Slave I/O Write commands while a bus master. Slave I/O Write cycles are fixed length cycles, i.e. the Am53C974A 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



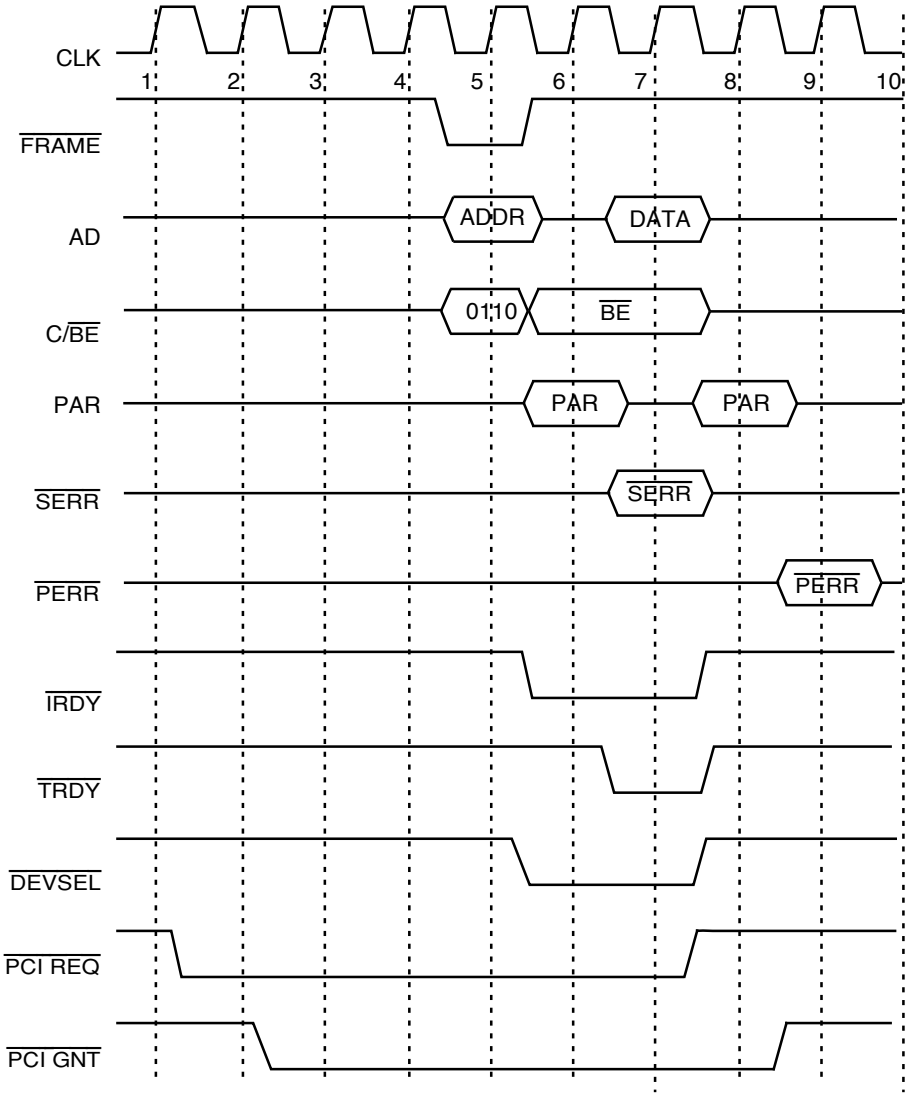
19113A-9

4.5.3 Master Memory Read

The Master Memory Read command is used by the Am53C974A when it will be reading 2 or less 32-bit locations of memory. If the device needs to read more than 2 Double-words (Dwords) of memory, the Master Memory Read Line command 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 Am53C974A will always activate all byte enables, even though some byte lanes may not contain "valid" data. In such instances, the Am53C974A will internally discard unnecessary bytes.

Figure 4-4 Master Memory Read Timing

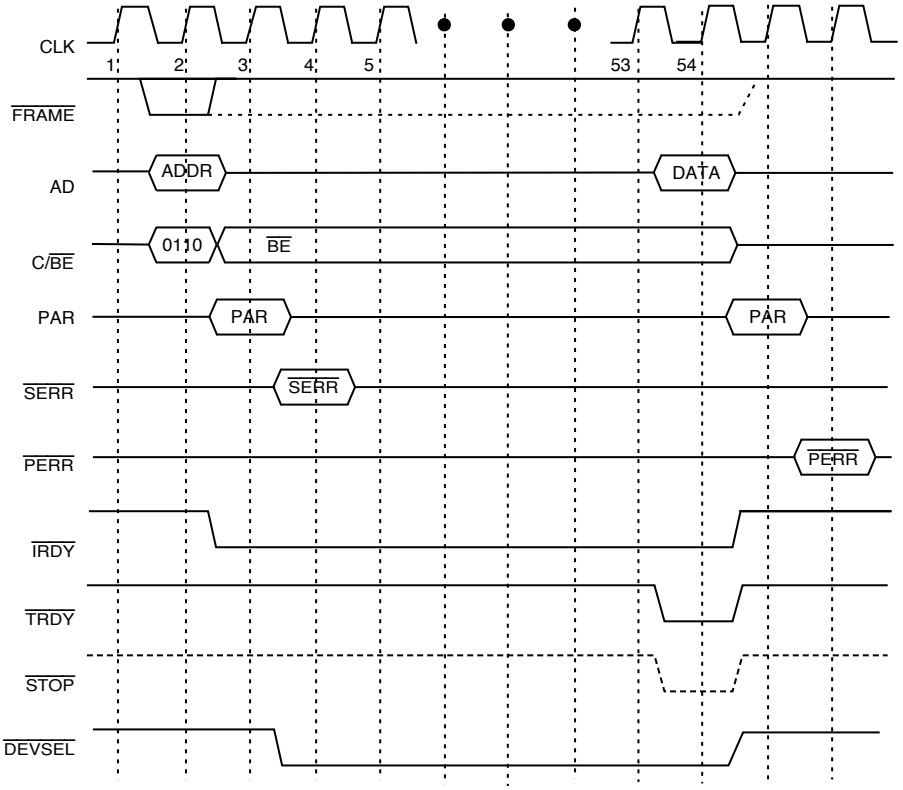


19113A-10

4.5.4 Slave Memory Read

The Slave Memory Read Command is used by the processor to read data from the Am53C974A's expansion ROM. This is a single cycle, non-burst 8-bit,16-bit, or 32-bit transfer which is initiated by the host CPU. The Am53C974A will always respond to this cycle by returning valid data on all byte lanes. Thus, it is the responsibility of the host to discard any unneccessary bytes via the C/$\overline{BE}$[3:0] lines. The Am53C974A will not produce Slave Memory Read commands while a bus master. Slave Memory Read cycles are fixed length cycles, i.e. the Am53C974A will return $\overline{TRDY}$ on the 54th bus cycle of the transfer. Figure 4-5 shows the timing for a Slave Memory Read bus cycle. Note that the Slave Memory Read Multiple and the Slave Memory Read Line commands are aliased to the Slave Memory Read command.

Figure 4-5 Slave Memory Read Timing

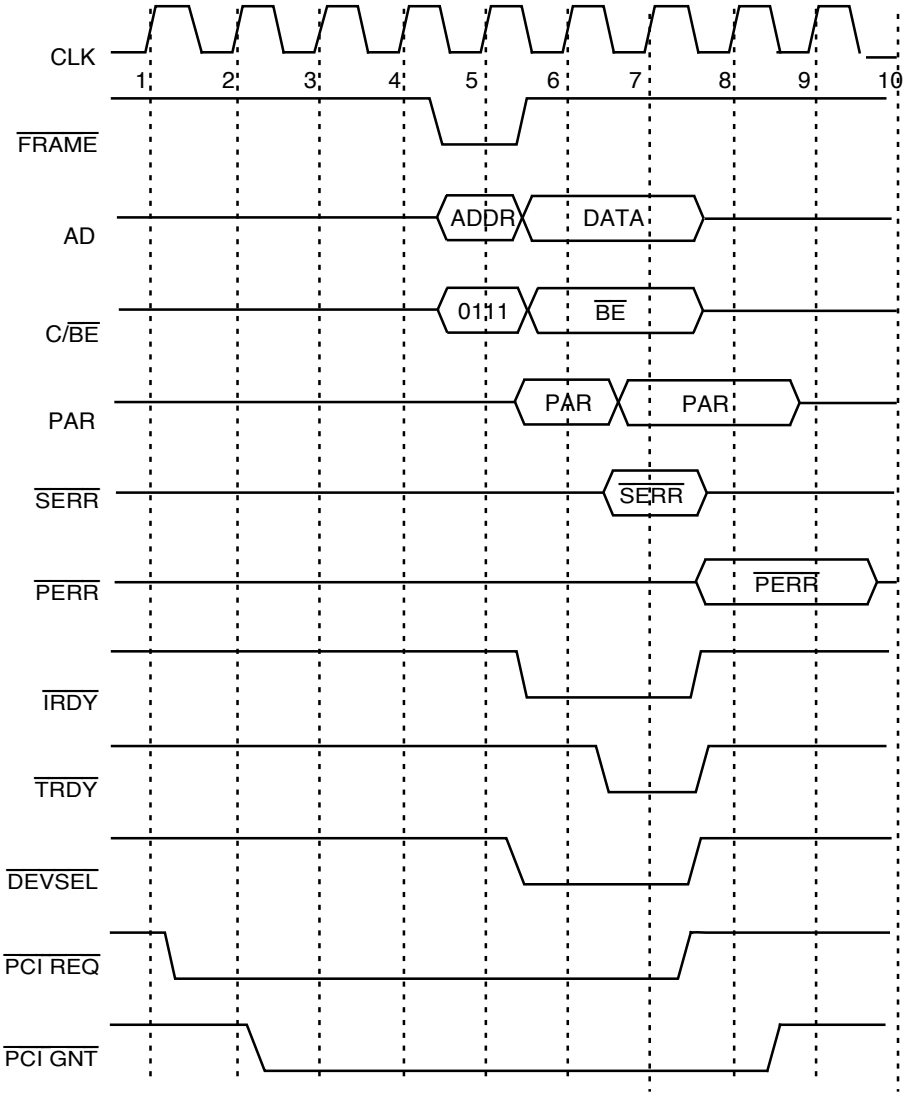


19113A-11

4.5.5 Master Memory Write

The Master Memory Write command is used by the Am53C974A when it will be writing to memory. Figure 4-6 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 $\overline{\text{TRDY}}$.

Figure 4-6 Master Memory Write Timing

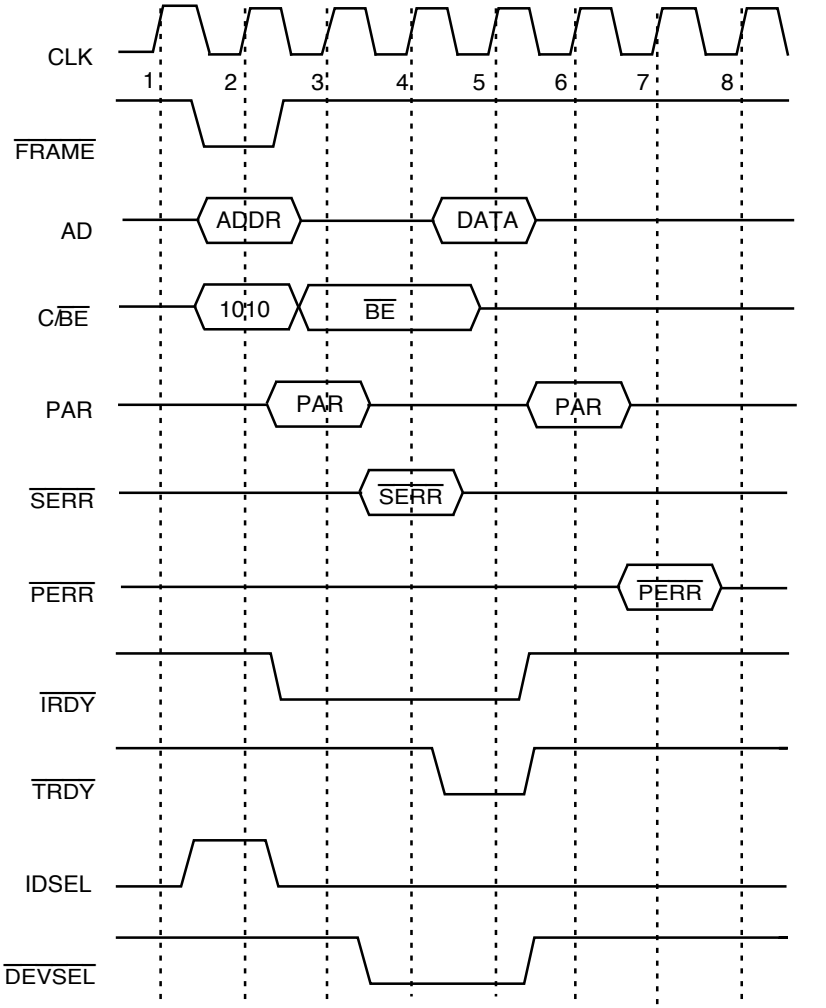


19113A-12

4.5.6 Slave Configuration Read

The Slave Configuration Read command is used by the host CPU to read the PCI configuration space in the Am53C974A. 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 Am53C974A will return TRDY on the 5th bus cycle of the transfer. Figure 4-7 shows the Configuration Read cycle timing.

Figure 4-7 Slave Configuration Read Timing

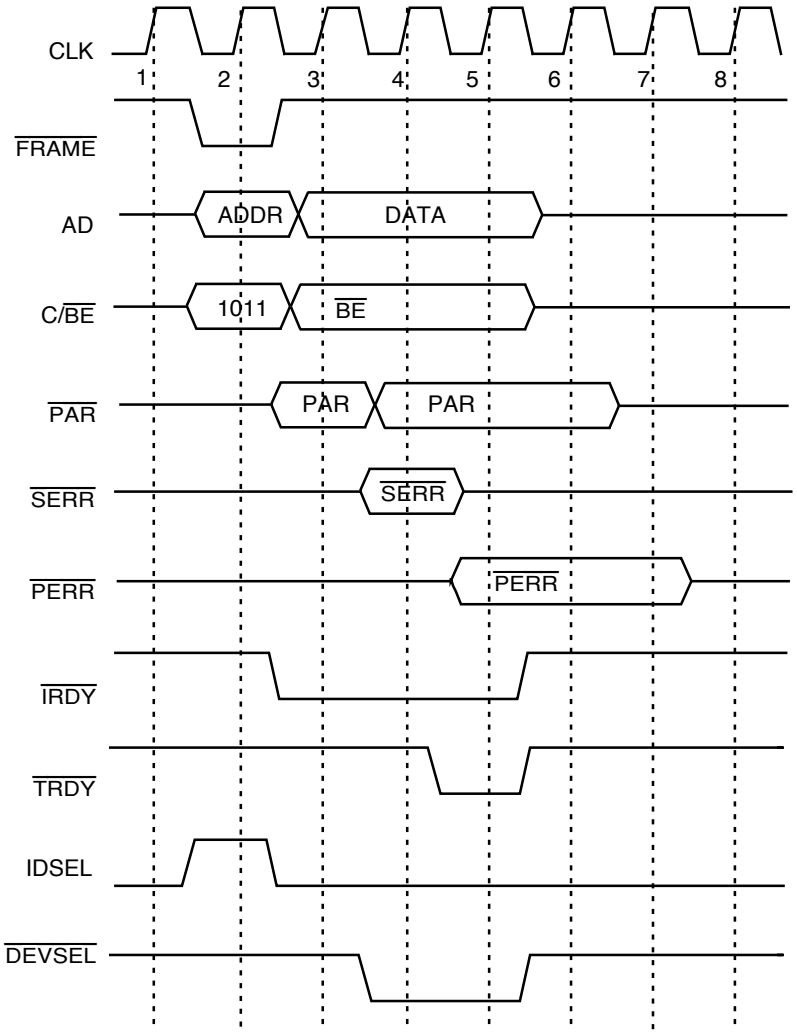


19113A-13

4.5.7 Slave Configuration Write

The Slave Configuration Write command is used by the host CPU to write the configuration space in the Am53C974A. 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 Am53C974A will return $\overline{\text{TRDY}}$ on the 5th bus cycle of the transfer. Figure 4-8 shows the Configuration Write cycle timing.

Figure 4-8 Slave Configuration Write Timing

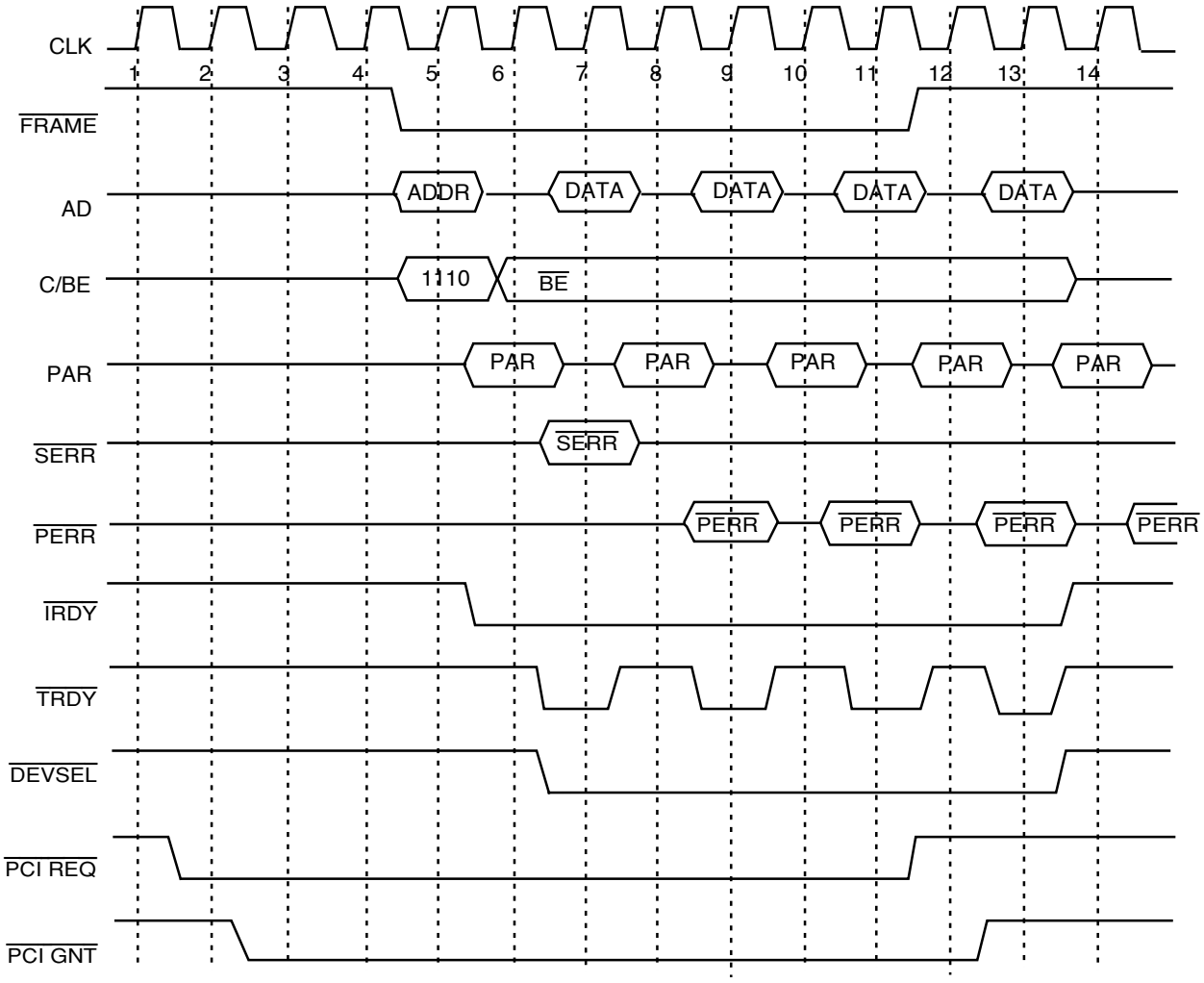


19113A-14

4.5.8 Master Memory Read Line

The Master Memory Read Line command is used by the Am53C974A 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 is used. Figure 4-9 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 $\overline{\text{TRDY}}$.

Figure 4-9 Master Memory Read Line Timing



19113A-15

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.

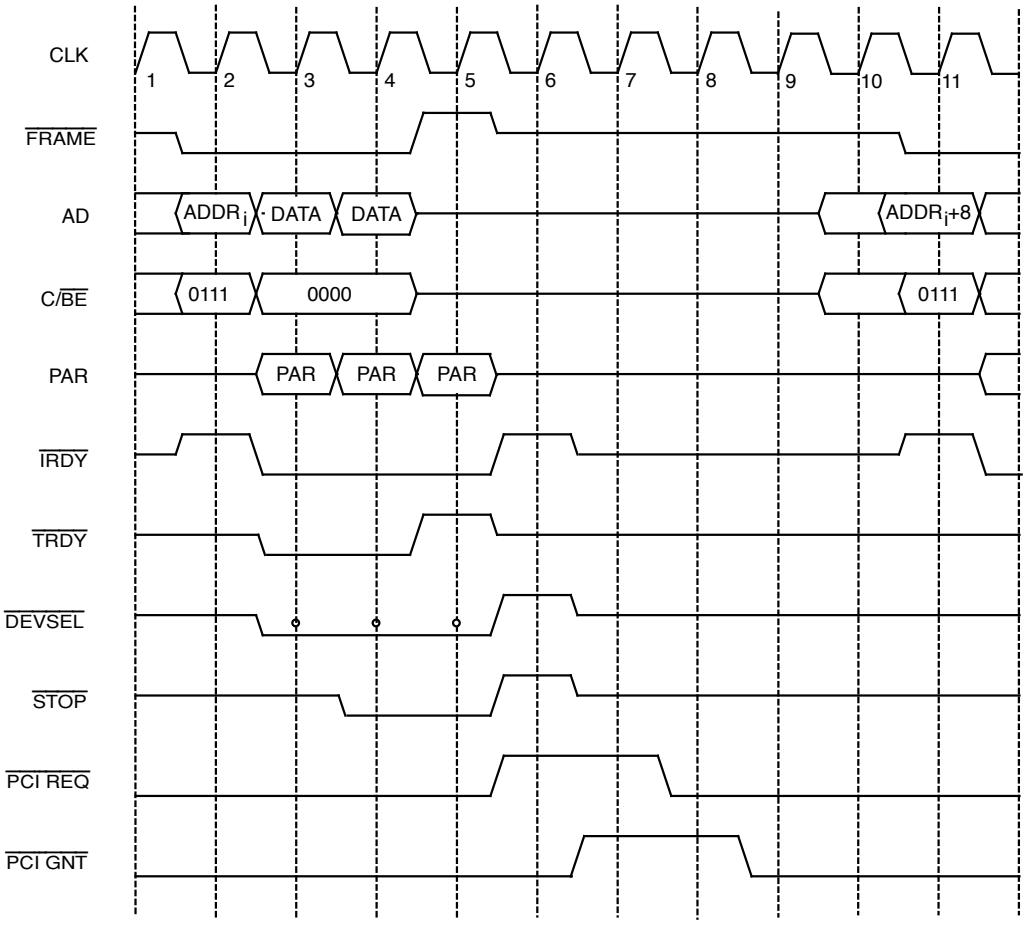
4.6.1 Target Initiated Termination

When the Am53C974A 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 4-10 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 Am53C974A 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 Am53C974A 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 4-10 Disconnect With Data Transfer



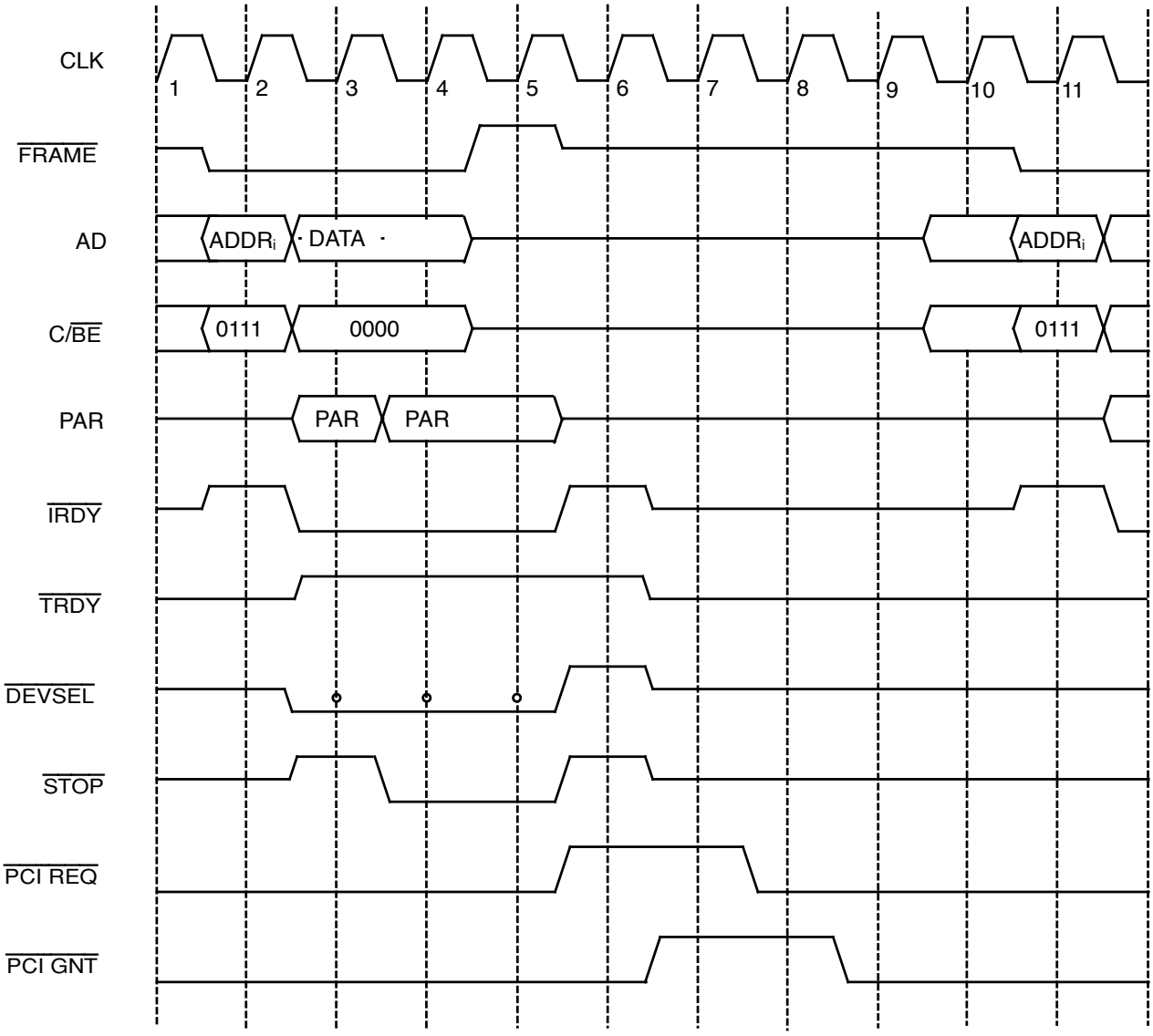
° $\overline{\text{DEVSEL}}$ *is sampled by the Am53C974A*

19113A-16

4.6.1.2 Disconnect Without Data Transfer

Figure 4-11 shows a target disconnect sequence during which no data is transferred. $\overline{STOP}$ is asserted on clock 4 without $\overline{TRDY}$ being asserted at the same time. The Am53C974A terminates the current transfer with the deassertion of $\overline{FRAME}$ on clock 5 and one clock cycle later with the deassertion of $\overline{IRDY}$. It finally releases the bus on clock 6. The Am53C974A 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 4-11 Disconnect Without Data Transfer



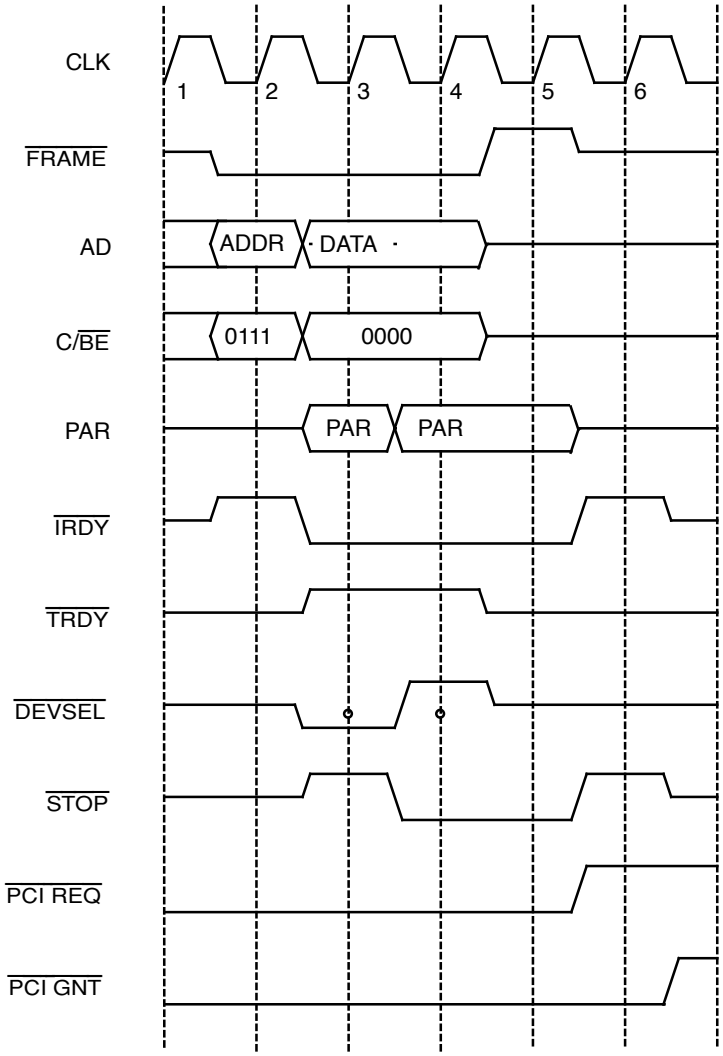
° $\overline{DEVSEL}$ *is sampled by the Am53C974A*

4.6.1.3 Target Abort

Figure 4-12 shows a target abort sequence. The target asserts DEVSEL for one clock. It then deasserts DEVSEL and asserts 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 Am53C974A cannot make any assumption about the success of the previous data transfers in the current transaction. The Am53C974A terminates the current transfer with the deassertion of FRAME on clock 5 and one clock cycle later with the deassertion of IRDY. It finally releases the bus on clock 6.

Since data integrity is not guaranteed, the Am53C974A 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 Am53C974A has received a target abort.

Figure 4-12 Target Abort



° DEVSEL *is sampled by the Am53C974A*

19113A-18

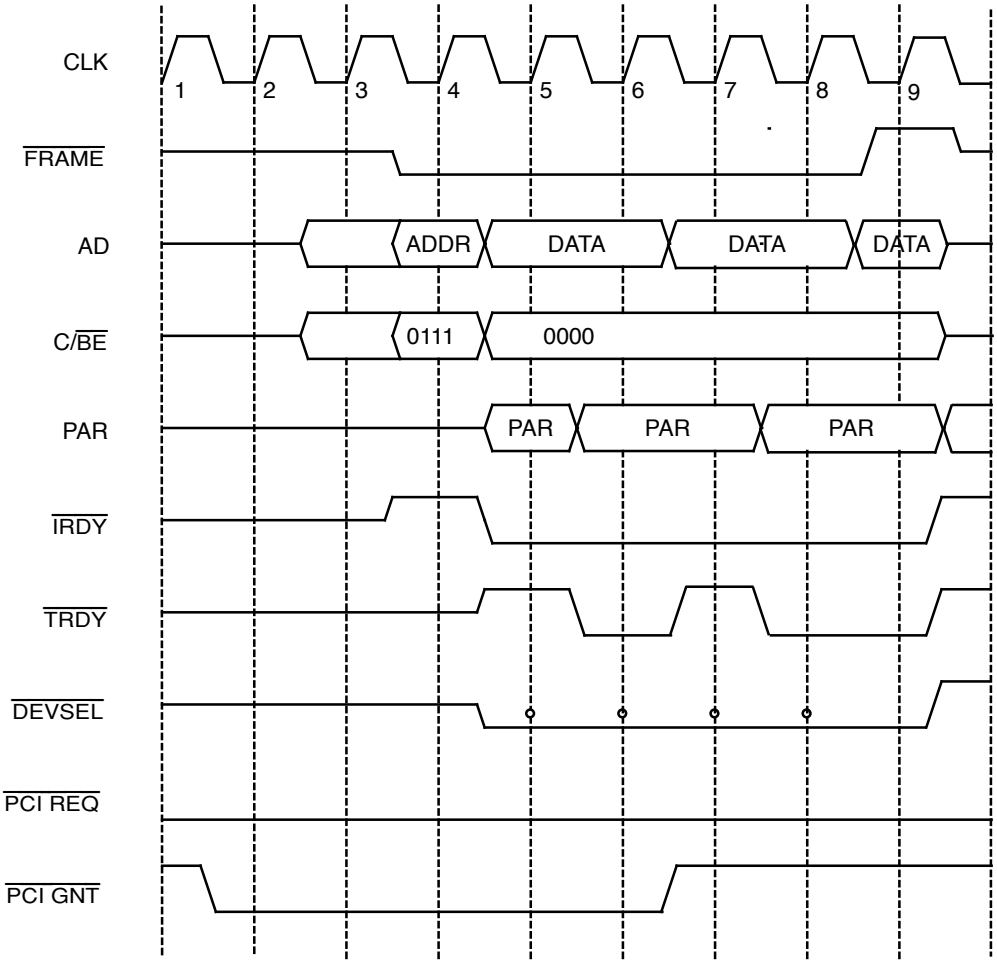
4.6.2 Master Initiated Termination

There are two scenarios besides normal completion of a transaction where the Am53C974A 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 Am53C974A away if the current bus operation takes too long. This may happen during DMA bursts. When $\overline{\text{PCI GNT}}$ is removed and the value in the Latency Timer register has counted to zero, the Am53C974A will finish the current transfer and then immediately release the bus. The Latency Timer in PCI configuration space of the Am53C974A is programmable. The Am53C974A will keep $\overline{\text{PCI REQ}}$ asserted to regain bus ownership as soon as possible.

Figure 4-13 Preemption



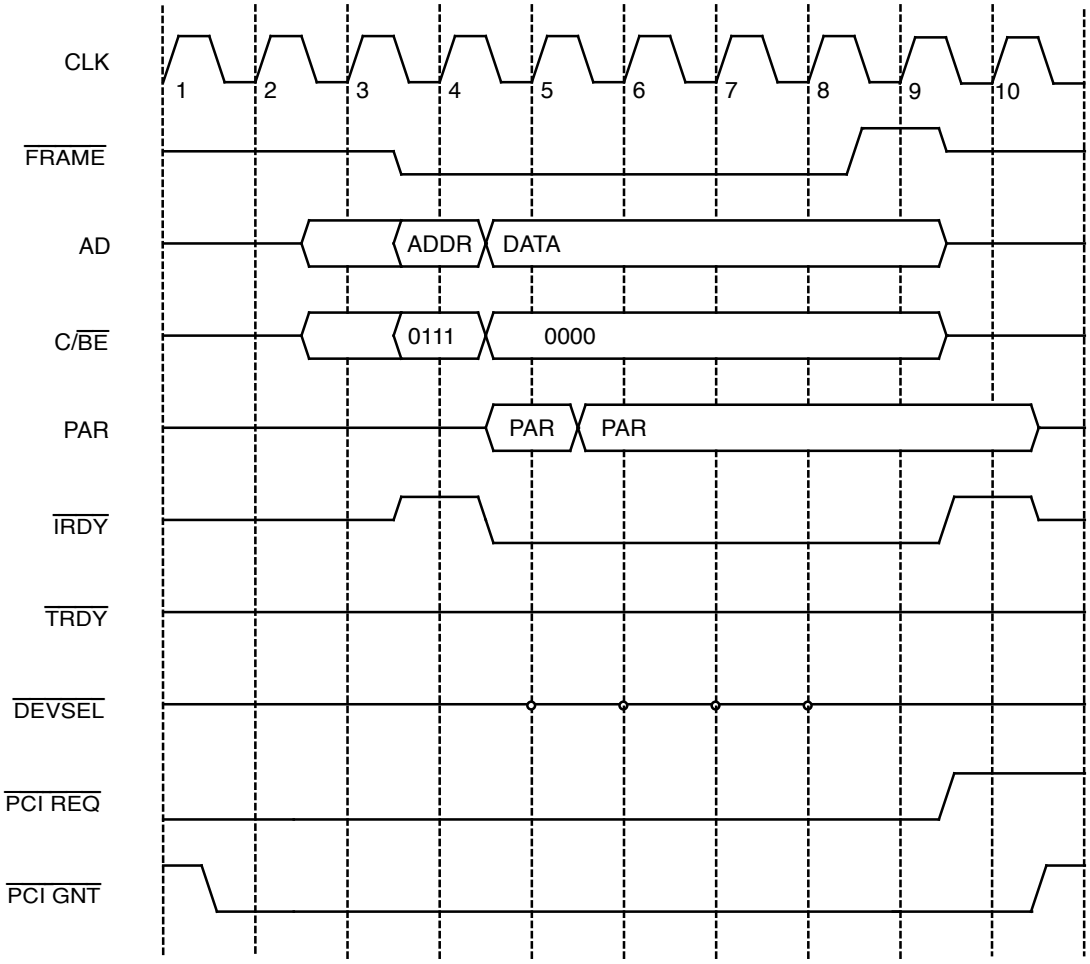
○ $\overline{\text{DEVSEL}}$ *is sampled by the Am53C974A*

19113A-19

4.6.2.2 Master Abort

The Am53C974A will terminate its cycle with a Master Abort sequence if $\overline{\text{DEVSEL}}$ is not asserted within 4 clocks after $\overline{\text{FRAME}}$ is asserted. Master Abort is treated as a fatal error by the Am53C974A. 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 Am53C974A has terminated its transaction with a master abort.

Figure 4-14 Master Abort



∘ $\overline{\text{DEVSEL}}$ *is sampled by the Am53C974A*

19113A-20

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 Am53C974A 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
Expansion ROM Base Address		30h
Reserved*		34h
Reserved*		38h
Max_Lat / Min_Gnt	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 Am53C974A. 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 Am53C974A, 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 Am53C974A 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 Am53C974A. 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 Am53C974A 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 Am53C974A 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 Am53C974A back-to-back transactions are only allowed to the same agent.

Bit 8 – $\overline{\text{SERREN}}$ – $\overline{\text{SERR}}$ Enable

This read/write bit is an enable bit for the $\overline{\text{SERR}}$ driver. When this bit is set to '1', the $\overline{\text{SERR}}$ driver is enabled. When this bit is reset to '0', the $\overline{\text{SERR}}$ 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 Am53C974A always does address stepping. The Am53C974A uses address stepping for the first address phase of each bus master period. $\overline{\text{FRAME}}$ will be asserted on the second clock following the assertion of $\overline{\text{PCI GNT}}$, indicating a valid address on the AD bus.

Bit 6 – PERREN – Parity Error Response Enable

This read/write bit controls the Am53C974A's response to parity errors. When PERREN is '0' and the Am53C974A detects a parity error, it only sets the Detected Parity Error bit in the PCI Configuration Space Status Register. When PERREN is '1', the Am53C974A asserts $\overline{\text{PERR}}$ 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 $\overline{\text{SERR}}$ pin and the $\overline{\text{SERR}}$ bit in the PCI Configuration Space Status Register. This bit must be reset to '0' after $\overline{\text{PCI RST}}$. 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 Am53C974A is not a graphics device.

Bit 4 – MWIEN – Memory Write & Invalidate Enable

This bit is hardwired to a value of '0' since the Am53C974A does not generate memory write & invalidate commands. Instead, the memory write command must be used.

Bit 3 – SCYCEN – Special Cycles Enable

The Am53C974A 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 Am53C974A'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 programmable to allow support for expansion ROM accesses. This bit must be set to '1' before the Expansion ROM can be accessed. When this bit is '0', access to the boot ROM is disabled.

Bit 0 – IOEN – I/O Space Enable

This read/write bit controls the Am53C974A'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 Am53C974A 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 Am53C974A samples the AD(31:00), C/$\overline{BE}$(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 Am53C974A.
- 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 Am53C974A sets the PERR bit if the target reports a data parity error by asserting the $\overline{PERR}$ signal. PERR is not affected 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 $\overline{SERR}$ 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 Am53C974A 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 Am53C974A 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 Am53C974A 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 $\overline{\text{DEVSEL}}$ signal. These bits are hardwired to a value of '0' and '1' (for bits 10 and 9 respectively) since the Am53C974A uses medium assertion timing for the $\overline{\text{DEVSEL}}$ signal. That is, DEVSEL is asserted two clocks after $\overline{\text{FRAME}}$ is asserted. These bits are read-only and indicate the time that the Am53C974A asserts $\overline{\text{DEVSEL}}$ for any bus command.

Bit 8 – DATAPERR – Data Parity Detected

DATAPERR is set when the Am53C974A 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 Am53C974A checks for parity errors by sampling the AD(31:00), C/$\overline{\text{BE}}$(3:0) and the PAR lines. During the data phase of all Memory Write commands, the Am53C974A checks the $\overline{\text{PERR}}$ input to detect whether the target has reported a parity error.

DATAPERR is set by the Am53C974A 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 Am53C974A 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. On the Am53C974A, the value of this register is '10h'.

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/WRITE**

This register specifies the maximum time the Am53C974A 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 Am53C974A 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 Am53C974A will finish the current data phase and then immediately release the bus.

The value for the Am53C974A Latency Timer Register is programmable.

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 Am53C974A. Bit 0 is hard-wired 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 Am53C974A in all of I/O space. IOBASE 26:0 must be written with a valid address before the Am53C974A 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 Am53C974A 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 Am53C974A will drive $\overline{\text{DEVSEL}}$ indicating it will respond to the access.

IOBASE 26:2 is read and written by the host. IOBASE 1:0 is hardwired to '0'.

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

IOSIZE 2:0 together with IOBASE 1:0 and bits 1 and 0 indicate the size of the I/O space the Am53C974A 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 6–1. This indicates an Am53C974A I/O space requirement of 128 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 Expansion ROM Base Address Register

Address 30h **READ/WRITE**

This is a 32-bit read/write register which is used to hold the expansion ROM Base Address. It is used to specify the size and alignment of the expansion ROM for the Am53C974A. It supports expansion ROMs of up to 64K.

31	30	29	28	27	26	25	24
ROM31	ROM30	ROM29	ROM28	ROM27	ROM26	ROM25	ROM24
X	X	X	X	X	X	X	X

23	22	21	20	19	18	17	16
ROM23	ROM22	ROM21	ROM20	ROM19	ROM18	ROM17	ROM16
X	X	X	X	X	X	X	X

15	14	13	12	11	10	9	8
ROM15	ROM14	ROM13	ROM12	ROM11	ROM10	ROM9	ROM8
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
ROM7	ROM6	ROM5	ROM4	ROM3	ROM2	ROM1	ROM0
0	0	0	0	0	0	0	0

Bit 31:16 – ROM31:16 – ROM Base Address

These bits are programmable for storing the expansion ROM base address. The values in these bits are unknown after power-on or reset.

Bit 15:11 – ROM15:11 – ROM Alignment

These bits are hardwired to '0', specifying a 64K alignment ROM support.

Bit 10:1 – ROM10:1 – Reserved

These bits are reserved as defined by the PCI specification, rev 2.0. These bits will return a value of 0 when read.

Bit 0 – ROM0 – ROM Address DecodeEnable

This bit is used to enable/disable the ROM address space decoding. When this bit is '0', the expansion ROM address space is disabled. When this bit is '1', address decoding is enabled using the value programmed into bits 31:16 of this register. This bit will default to '0' after power-on or reset.

4.7.1.13 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 Am53C974A'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.

4.7.1.14 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 Am53C974A only uses $\overline{\text{INTA}}$.

4.7.1.15 Min_Gnt Register

Address 3Eh **READ ONLY**

The Min_Gnt register is an 8-bit read only register. It is hardwired to a value of '04h'. This value equals a burst period of 1 μs calculated at a 33 MHz clock rate. The register value specifies the time in units of 1/4 microseconds. The host should use the value of this register to determine the setting of the Am53C974A's Latency Timer Register.

4.7.1.16 Max_Lat Register

Address 3Fh **READ ONLY**

The Max_Lat register is an 8-bit read only register. It is hardwired to a value of '28h'. This value equals a PCI bus latency of 10 μs calculated at a 33 MHz clock rate. The register value specifies the time in units of 1/4 microseconds. The host should use the value of this register to determine the setting of the Am53C974A's Latency Timer Register.

4.7.2 Device Dependent Register Description

The Am53C974A 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 Am53C974A 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 Am53C974A 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 Am53C974A 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	FSCSI	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 15:14 – Reserved

These 2 bits are reserved by AMD's SCSI software driver.

Bit 13 – FSCSI – Fast SCSI Drive Present

This bit identifies the Target as a Fast SCSI drive. A value of '1' indicates that a Fast drive is present, while a '0' indicates either that a drive is not present, or that the drive is not capable of sustaining fast synchronous SCSI transfers rates of 10 Mbytes/s. This bit will default to a value of '0'.

Note: *This register is defined by AMD's PCscsi Software and does not represent a change in the chip's functionality.*

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 Am53C974A 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 the following 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 Am53C974A are controlled by the FASTSCSI and FASTCLK bits in Control Register Three, as well as by the Extended Timing Feature in Control Register One. The chart below shows the effects of different bit configurations on minimum asynchronous and synchronous cycle times.

FASTCLK 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.

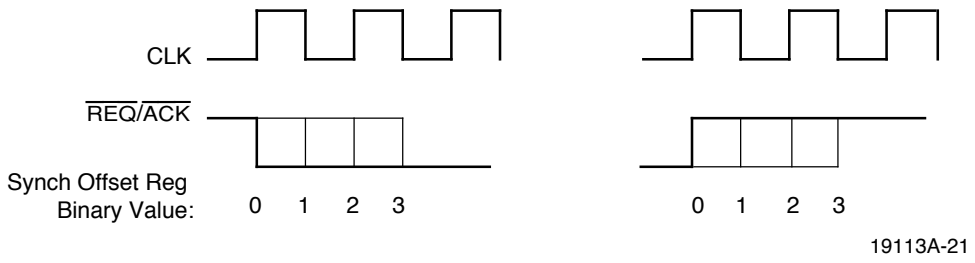
Bits 4:0 in the Synchronous Transfer Period Register ((B)+18h) specify the timing between the leading edges of consecutive $\overline{\text{REQ}}$ and $\overline{\text{ACK}}$ pulses during synchronous transfers. For programming information, refer to the register level descriptions.

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

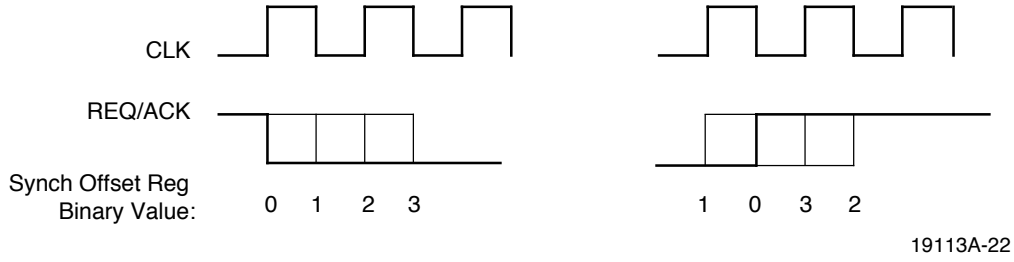
The assertion and deassertion time for the $\overline{\text{REQ}}$ and $\overline{\text{ACK}}$ signals may be controlled via the Synchronous Offset Register ((B)+1Ch). Bits 7:6 control $\overline{\text{REQ}}/\overline{\text{ACK}}$ deassertion delay, while bits 5:4 control $\overline{\text{REQ}}/\overline{\text{ACK}}$ assertion delay. The deassertion for $\overline{\text{REQ}}/\overline{\text{ACK}}$ may be moved ahead 0.5 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 $\overline{\text{REQ}}/\overline{\text{ACK}}$ can vary from 0 to 1.5 clock cycles.

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

FASTCLK Enabled



FASTCLK Disabled



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

5.1.5 Parity

Parity on the SCSI bus is such that the total number of logical ones on the data bus including the parity bit must be odd. 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 Am53C974A 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 Am53C974A 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 $\overline{ATN}$ 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 Am53C974A generates parity internally for each byte. Note that the parity on the PCI bus is generated internally and is distinct from the parity received from the SCSI bus.

5.1.6 Reset Levels

The Am53C974A has two reset pins and two reset commands that affect the SCSI block. The $\overline{\text{PCI RST}}$ pin resets the whole chip including the SCSI controller and the PCI interface.

The Reset Device command causes almost the same effect on the SCSI controller that the $\overline{\text{PCI RST}}$ pin does. However, the Reset Device command has no effect on the PCI interface. Also, after the Reset Device command has been issued, the user must issue a NOP command before another command can be executed.

The Action of the $\overline{\text{PCI RST}}$ signal or the Reset Device command is called Hard Reset.

The $\overline{\text{SCSI RST}}$ pin is a bidirectional signal on the SCSI bus that resets a portion of the SCSI logic when it is asserted by a device on the SCSI bus. Similarly, the Am53C974A can assert the $\overline{\text{SCSI RST}}$ signal to cause all of the other devices on the SCSI bus to reset.

The Reset SCSI command causes the same effect on the SCSI controller that the $\overline{\text{SCSI RST}}$ pin does, except that this command also causes the $\overline{\text{SCSI RST}}$ pin to be asserted so that all other (external) devices on the SCSI bus are also reset. Once a SCSI Reset command has been executed, the $\overline{\text{SCSI RST}}$ signal will remain asserted until a Hard Reset has occurred.

The action of the $\overline{\text{SCSI RST}}$ signal and the Reset SCSI command is called Soft Reset.

In addition there is a third type of reset, called Disconnected Reset, that is caused by certain sequences on the SCSI bus. These three types of reset are described in the following sections.

5.1.6.1 Hard Resets: (H)

This reset occurs at power up, when the $\overline{\text{PCI RST}}$ pin is asserted through external hardware, or when the Reset Device command is issued by writing 02h to the SCSI command register at ((B)+0Ch). Hard reset causes all chip functions to halt and resets all internal state machines. It leaves the SCSI block in the disconnected state. It leaves all SCSI registers in their default states.

In addition, if the Hard Reset is caused by the assertion of the $\overline{\text{PCI RST}}$ pin, the following actions occur.

- The Command register in the PCI configuration space is cleared to zero. (No other register in the configuration space is affected.)
- The DMA CCB registers are set to their default values.

5.1.6.2 Soft Reset: (S)

This reset occurs either when the $\overline{\text{SCSI RST}}$ pin on the SCSI bus is asserted or when the Reset SCSI Bus command is issued (by writing 03h to the SCSI command register at ((B)+0Ch). Soft reset causes the following actions to occur:

- All SCSI bus signals except $\overline{\text{SCSI RST}}$ are released.
- The chip is returned to the Disconnected state.
- An interrupt is generated 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 Am53C974A to disconnect from the SCSI bus. Disconnected reset occurs when any of the following conditions occur:

- Target Disconnect, Disconnect Steps, and Terminate Steps
- The Am53C974A is the Initiator and the SCSI bus moves to a Bus Free state.
- The Selection or Reselection command terminates due to selection timeout.

Disconnected reset causes the following actions to occur:

- All SCSI signals except $\overline{\text{SCSI RST}}$ are deasserted.
- The SCSI Command Register is initialized to empty.
- The IS1 and IS0 bits in the Internal State Register, ((B)+18h), are cleared to 0.
- Resets bit 6:4 in the Command Register.

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 bits 6:4 in the Command Register	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
Target (IS 0 = ‘0’)	HSD
Initiator (IS 1:0 = ‘00’)	HSD
Reset command sequence module	HS
Reset DMA Interface	HS
Reset bus-initiated selection/reselection 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 Register = ‘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 Am53C974A 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 Am53C974A is stored at register address (10h) in the PCI Configuration space.

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	FIF0 (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	GCV	PHASE MSG	PHASE C/D	PHASE I/O
Interrupt (B)+14h	SCSI RST	INVALID CMD	DISC	SVC REQ	SUCCESS OP	RESEL	SEL W/ATN	SEL
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	SCSI-2 FEATURES	RES	RES	RES
Control 3 (B)+30h	ADD'L ID CHK	QTAG EN	GROUP2 CMD	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	FIF0 (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 Am53C974A's 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 Am53C974A is 16 bytes deep and is used to transfer SCSI data to and from the Am53C974A. The FIFO may be accessed via a read or write to this register. This is the only register that can be accessed with $\overline{REQ}$ or $\overline{ACK}$. 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 Am53C974A 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
  Reselect command (if ATN is asserted)
  Target disconnect or Terminate command
- Selection or reselection timeout
- Bad parity received in Target mode
- ATN asserted in Target mode
- Receiving a message while in Target mode and ATN is deasserted before completion
- 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.

Bits 6:0 – CMD 6:0 – Command 6:0

These command bits decode the commands that the device needs to perform. There are a total of 31 commands grouped into four categories. The groups are Initiator Commands, Target Commands, Selection/Reselection Commands and General Purpose Commands. See Section 5.3 for descriptions of these 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	GCV	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 determine 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 – GCV – Group Code Valid

The GCV bit is set if the group code field in the Command Descriptor Block (CDB) is one that is defined by the ANSI Committee in their document X3.131 – 1986. If the SCSI-2 Features Enable (S2FE) bit in the Control Register 2 ((B)+2Ch) is set, Group 2 commands will be treated as 10-byte commands and the GCV bit will be set. If S2FE is reset then Group 2 commands will be treated as reserved commands. Group 3 and 4 commands will always be considered reserved commands. The device will treat all reserved commands as 6-byte commands. Group 6 commands will always be treated as vendor unique 6-byte commands and Group 7 commands will always be treated as vendor unique 10-byte commands.

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

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	SELA	SEL
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 in the Target or the Initiator mode when the device disconnects from the SCSI bus. In the Target mode this bit will be set if a Terminate or a Command Complete steps causes the device to disconnect 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 in the Target or the Initiator mode when another device on the SCSI bus has a service request. In the Target mode, this bit will be set when the Initiator asserts the ATN signal. In the Initiator mode, this bit is set when a Command Steps Successfully Completed Command is issued.

Bit 3 – SO – Successful Operation

The SO bit can be set in the Target or the Initiator mode when an operation has successfully completed. In the Target mode this bit will be set when any Target or Idle state command is 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 – SELA – Selected with Attention

The SELA bit is set at the end of the selection phase indicating that the device has been selected as a Target by the Initiator and that the ATN signal was active during Selection.

Bit 0 – SEL – Selected

The SEL bit is set at the end of the selection phase indicating that the device has been selected as a Target by the Initiator and that the ATN signal was inactive during Selection.

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) = 250×10^{-3} s.

Clock Frequency: 40 MHz. (assume) = 40×10^{6} Hz.

Clock Factor: 8 (See Clock Factor Register)

STIM 7:0 = (250 x 10–3) X (40 x 106) / (8192 (8)) = 152.59 decimal

The decimal value of **152.59 must be rounded up to 153** (the next integer value), and its hexadecimal value of '99h' 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) indicate the completion status of certain device commands. Certain commands cause the contents of the 3-bit Internal State register to be changed at several steps in the execution process. The value left in this register when the command terminates along with the contents of the Interrupt Status register indicate how far the execution had proceeded prior to the command termination. The following status decode tables show how to interpret the Internal State register after these commands have terminated.

Status Decode:

Initiator Select without $\overline{ATN}$ 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 $\overline{ATN}$ 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 $\overline{ATN}$ 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 $\overline{ATN}$ 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 $\overline{ATN}$ true, then released $\overline{ATN}$; 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; $\overline{ATN}$ still driven by the Am53C974A
Initiator Select with $\overline{ATN3}$ 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 $\overline{ATN3}$ 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; $\overline{ATN}$ released only if third message byte was sent
0	18	Arbitration and selection completed; sequence halted because Target did not assert message out phase; $\overline{ATN}$ still driven by the Am53C974A

Status Decode (continued):

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 Am53C974A
1	18	Message out completed; one message byte sent; $\overline{\text{ATN}}$ on
Target Selected without $\overline{\text{ATN}}$ Steps		
Internal State Register ((B)+18h) Bits 2:0 (Hex)	Interrupt Status Register ((B)+14h) Bits 7:0 (Hex)	Explanation
2	11	Selected; received entire CDB; check group code valid bit; Initiator asserted $\overline{\text{ATN}}$ in command phase
1	11	Sequence halted in command phase due to parity error; some CDB bytes may not have been received; check FIFO flags; Initiator asserted $\overline{\text{ATN}}$ in command phase
2	01	Selected; received entire CDB; check group code valid bit
1	01	Sequence halted in command phase because of parity error; some CDB bytes may not have been received; check FIFO flags
0	01	Selected; loaded bus ID into FIFO; null-byte message loaded into FIFO
Target Select with $\overline{\text{ATN}}$ Steps, SCSI-2 Bit NOT SET		
Internal State Register ((B)+18h) Bits 2:0 (Hex)	Interrupt Status Register ((B)+14h) Bits 7:0 (Hex)	Explanation
2	12	Selection complete; received one message byte and entire CDB; Initiator asserted $\overline{\text{ATN}}$ during command phase
1	12	Halted in command phase; parity error and $\overline{\text{ATN}}$ true
0	12	Selected with $\overline{\text{ATN}}$; stored bus ID and one message byte; sequence halted because $\overline{\text{ATN}}$ remained true after first message byte
2	02	Selection completed; received one message byte and the entire CDB
1	02	Sequence halted in command phase because of parity error; some CDB bytes not received; check group code valid bit and FIFO flags
0	02	Selected with $\overline{\text{ATN}}$; stored bus ID and one message byte; sequence halted because of parity error or invalid ID message

Status Decode (continued):

Target Select with $\overline{ATN}$ Steps, SCSI-2 Bit SET		
Internal State Register ((B)+18h) Bits 2:0 (Hex)	Interrupt Status Register ((B)+14h) Bits 7:0 (Hex)	Explanation
6	12	Selection completed; received three message bytes and entire CDB. $\overline{ATN}$ is true
5	12	Halted in command phase; parity error and $\overline{ATN}$ true
4	12	$\overline{ATN}$ remained true after third message byte
2	12	Selection completed; Initiator deasserts $\overline{ATN}$ after receipt of one message byte; entire CDB received. $\overline{ATN}$ asserted during command phase
1	12	Sequence halted during command phase; Initiator deasserts $\overline{ATN}$ after receipt of one message byte; parity error and $\overline{ATN}$ true
0	12	Selected with $\overline{ATN}$; stored bus ID and one message byte; sequence halted because of parity error or invalid ID message; $\overline{ATN}$ is true
6	02	Selection completed; received three message bytes and the entire CDB
5	02	Received three message bytes then halted in command phase because of parity error; some CDB bytes not received; check group code valid bit and FIFO flags
4	02	Parity error during second or third message byte
2	02	Selection completed; Initiator deasserts $\overline{ATN}$ after receipt of one message byte; entire CDB received
1	02	Sequence halted during command phase because of parity error; Initiator deasserts $\overline{ATN}$ after receipt of one message byte; some bytes of CDB not received; check FIFO flags and group code valid bit
0	02	Selected with $\overline{ATN}$; stored bus ID and one message byte; sequence halted because of parity error or invalid ID message
Target Receive Command Steps		
Internal State Register ((B)+18h) Bits 2:0 (Hex)	Interrupt Status Register ((B)+14h) Bits 7:0 (Hex)	Explanation
2	18	Received entire CDB; Initiator asserted $\overline{ATN}$
1	18	Sequence halted during command transfer due to parity error; $\overline{ATN}$ asserted by Initiator
2	08	Received entire CDB
1	08	Sequence halted during command transfer due to parity error; check FIFO flags

Status Decode (continued):

Target Disconnect Steps		
Internal State Register ((B)+18h) Bits 2:0 (Hex)	Interrupt Status Register ((B)+14h) Bits 7:0 (Hex)	Explanation
2	28	Disconnect steps fully executed; disconnected; bus is free
1	18	Two message bytes sent; sequence halted because Initiator asserted $\overline{ATN}$
0	18	One message byte sent; sequence halted because Initiator asserted $\overline{ATN}$
Target Terminate Steps		
Internal State Register ((B)+18h) Bits 2:0 (Hex)	Interrupt Status Register ((B)+14h) Bits 7:0 (Hex)	Explanation
2	28	Terminate steps fully executed; disconnected; bus is free
1	18	Status and message bytes sent; sequence halted because Initiator asserted $\overline{ATN}$
0	18	Status byte sent; sequence halted because Initiator asserted $\overline{ATN}$
Target Command Complete Steps		
Internal State Register ((B)+18h) Bits 2:0 (Hex)	Interrupt Status Register ((B)+14h) Bits 7:0 (Hex)	Explanation
0	18	Status byte sent; sequence halted because Initiator set $\overline{ATN}$
2	08	Command complete steps fully executed

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 Am53C974A 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 $\overline{REQ}/\overline{ACK}$ 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 $\overline{REQ}$ (or $\overline{ACK}$). Bytes exceeding the threshold will be sent one byte at a time (asynchronously). That is, each byte will require an $\overline{REQ}/\overline{ACK}$ 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 – $\overline{REQ}/\overline{ACK}$ Deassertion

These bits may be programmed to control the deassertion delay of the $\overline{REQ}$ and $\overline{ACK}$ 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 $\overline{REQ}/\overline{ACK}$ 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 – $\overline{REQ}/\overline{ACK}$ Assertion

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

SOFREG Bits 5:4	Assertion Delay $\overline{REQ}/\overline{ACK}$ 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 $\overline{ACK}$ (or $\overline{REQ}$) signal. A zero value designates Asynchronous transfers, while a non-zero value designates the byte offset for synchronous transfers. The Am53C974A 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 Am53C974A.

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. These bits encode the decimal value to be used in calculating the SCSI Timeout Register value.

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. For this case, a value of 8 should be used to calculate the SCSI Timeout Register value. See the SCSI Timeout Register.*

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	SCSI-2	Reserved	Reserved	Reserved
0	0	0	0	0	0	0	0

Control Register Two (CNTLREG2) programs various operating parameters for the Am53C974A.

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 Am53C974A'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 – S2FE – SCSI–2 Features Enable

The S2FE bit allows the device to recognize two SCSI-2 features: the extended message feature and the Group 2 command recognition. (These features can also be controlled independently by bits 6:5 in CNTLREG3).

Extended Message Feature: When the S2FE bit is set and the device is selected with attention, the device will monitor the ATN signal at the end of the first message byte. If the ATN signal is active, the device will request two more message bytes before switching to the command phase. If the ATN signal is inactive the device will switch to the Command phase. When the S2FE bit is reset as a Target the device will request a single message byte.

Group 2 Command Recognition: When the S2FE bit is set, Group 2 commands are recognized as 10-byte commands. When the S2FE bit is reset, the device will interpret Group 2 commands as reserved commands. Thus the Am53C974A will only request 6-byte commands when the S2FE bit is reset.

Bit 2:0 – 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	QTAG	G2CB	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 Am53C974A 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 – QTAG – QTAG Control

This bit controls the Queue Tag feature in the Am53C974A. When enabled, the Am53C974A is capable of receiving 3-byte messages during bus-initiated Select/ Reselect with ATN. The 3-byte message consists of one byte Identify Message and two bytes of Queue Tag message. The Am53C974A will check the second byte for values of 20h, 21h, and 22h. If this condition is not satisfied, the sequence halts and the Am53C974A generates an interrupt.

When the QTAG feature is not enabled, the Am53C974A halts the Selected with ATN sequence following the receipt of one ID message byte if ATN is still true.

Bit 3, Control Register Two also enables this feature.

Bit 5 – G2CB – Group 2 Command Block

When this bit is set, the Am53C974A is capable of recognizing 10-byte Group 2 Commands as valid CDBs (Command Descriptor Blocks). (This feature is also controlled by bit 3 of CNTLREG2). When this feature is enabled, the Target receives 10 bytes of Group 2 commands, and sets the group code valid bit (bit 3) in the Status Register (STATREG). When this feature is disabled, the Target receives only 6 bytes of command code, and does not set bit 3 in the Status Register ((B)+10h).

This bit may be programmed in conjunction with bit 6 (described above) to send 1 or 3 byte messages with 6 or 10 byte CDBs. The following table illustrates the transmission options:

CNTLREG3 Bit 6 QTAG	CNTLREG3 Bit 5 G2CB	CNTLREG2 Bit 3 S2FE	Enabled Features
X	X	1	10-byte CDB, 3-byte message
1	0	0	3-byte message
0	1	0	10-byte CDB
1	1	0	10-byte CDB, 3-byte message
0	0	0	Features disabled

X is don't care

Bit 4 – FASTSCSI – Fast SCSI

Bit 3 – FASTCLK – Fast SCSI Clocking

These bits configure the Am53C974A'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 Am53C974A'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)	RADE	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 $\overline{\text{REQ}}$ and $\overline{\text{ACK}}$ 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 using bit 21 in the SBAC register ((B)+ 70h).

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'.

5.2.3.20 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 Am53C974A. 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

The ID value in this register is 12h.

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 SCSI 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.

Before a DMA device command is issued, the software must initialize the DMA Starting Transfer Count and Starting Physical Address registers. Then it must issue a DMA Start command. Once this is done, it can issue the device command.

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.

Some of the non-DMA commands are output commands that transfer data from the SCSI FIFO to the SCSI bus. Before these commands are executed, the SCSI FIFO must be loaded with the bytes to be sent. Other non-DMA commands are input commands that transfer data from the SCSI bus to the SCSI FIFO. After these commands are executed, the data received can be accessed by reading one byte at a time from the SCSI FIFO.

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	18	98
Set $\overline{\text{ATN}}$*	1A	–
Reset $\overline{\text{ATN}}$*	1B	–
Target Commands		
Send Message	20	A0
Send Status	21	A1
Send Data	22	A2
Disconnect Steps	23	A3
Terminate Steps	24	A4
Target Command Complete Steps	25	A5
Disconnect	27	–
Receive Message Steps	28	A8
Receive Commands	29	A9
Receive Data	2A	AA
Receive Command Steps	2B	AB
DMA Stop	04	–
Access FIFO	–	85
Idle State Commands		
Reselect Steps	40	C0
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 $\overline{\text{ATN3}}$ Steps	46	C6
Reselect with $\overline{\text{ATN3}}$ Steps	47	C7
General Commands		
No Operation*	00	80
Clear FIFO*	01	81
Reset Device*	02	82
Reset SCSI Bus**	03	83

** These commands do not generate interrupt.*

*** An interrupt is generated when SCSI bus reset interrupt reporting is not disabled (see Control Register1/DISR bit 6).*

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 latch 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 Am53C974A, 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 Am53C974A 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 $\overline{ACK}$ is still asserted.
- A Selection or Reselection command is issued with the DMA bit enabled, if the Selection/Reselection 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 or Reselection begins, has been eliminated. This prevents a false Successful Operation Interrupt from being generated when the Selection/Reselection 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 $\overline{BSY}$ signal line while the Am53C974A (Initiator) is waiting for the Target to assert $\overline{REQ}$, a Disconnected Interrupt will be issued 1.5 to 3.5 clock cycles following $\overline{BSY}$ going false.

Upon receipt of the last byte during Message In phase, $\overline{ACK}$ 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 Am53C974A 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 Am53C974A proceeds to Disconnect.

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

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 Am53C974A clears the Command Register (CMDREG), and generates a Service Request interrupt when the Target asserts $\overline{REQ}$.
- Transfer has successfully completed. If the phase is Message Out, the Am53C974A deasserts $\overline{ATN}$ before asserting $\overline{ACK}$ for the last byte of the message. When the Target asserts $\overline{REQ}$, a Service Request interrupt is generated.
- In the Message In phase when the device receives the last byte. The Am53C974A keeps the $\overline{ACK}$ signal asserted and generates a Successful Operation interrupt.

During Synchronous Data transfers the Target may send up to the maximum synchronous threshold number of $\overline{REQ}$ pulses to the Initiator. If it is the Synchronous Data-In phase then the Target sends the data and the $\overline{REQ}$ 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 $\overline{ATN}$ 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 $\overline{ATN}$ 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 $\overline{REQ}$ 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 $\overline{ACK}$ signal asserted to allow the Initiator to examine the message and assert the $\overline{ATN}$ 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 $\overline{ACK}$ 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 $\overline{REQ}$ is asserted by the Target.

After the device has received the last byte of message, it keeps the $\overline{ACK}$ 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 $\overline{ATN}$ signal must be asserted (with the help of the Set $\overline{ATN}$ Command) before issuing the Message Accepted Command. In either case, the Message Accepted Command has to be issued to release the $\overline{ACK}$ 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 Am53C974A 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 $\overline{ACK}$ 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 Target Commands

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

A SCSI bus reset during any Target command will cause the device to abort the command sequence, flag a SCSI bus reset interrupt (if the interrupt is enabled) and disconnect from the SCSI bus.

Normal or successful completion of a Target command will cause a Successful Operation interrupt to be generated. If the ATN signal is asserted during a Target command sequence, the Service Request bit is asserted in the Interrupt Status Register (INSTREG). If the ATN signal is asserted when the device is in an Idle state, a Service Request interrupt will be generated, the Successful Operation bit in the Interrupt Status Register (INSTREG) will be reset, and the Command Register (CMDREG) cleared.

During a command sequence, the Am53C974A decodes bits 7–5 of the 1st byte received during the command phase to determine the Group Code and CDB length. The following table shows Group Codes and their corresponding block lengths.

Group Code Bits 7–5	CDB Length	Command Group Status
000	6	Valid
001	10	Valid
010	6	Reserved (SCSI-2 mode or G2CB disabled)
010	10	Reserved (SCSI-2 mode or G2CB enabled)
011	6	Reserved
100	6	Reserved
101	12	Valid
110	6	Valid
111	10	Valid

5.3.5.1 Send Message Command (Command Code 20H/A0H)

The Send Message Command is used by the Target to inform the Initiator to receive a message. The SCSI bus phase lines are set to the Message In Phase and message bytes are transferred from the SCSI FIFO to the SCSI bus.

5.3.5.2 Send Status Command (Command Code 21H/A1H)

The Send Status Command is used by the Target to inform the Initiator to receive status information. The SCSI bus phase lines are set to the Status Phase and status bytes are transferred from the SCSI FIFO to the SCSI bus.

5.3.5.3 Send Data Command (Command Code 22H/A2H)

The Send Data Command is used by the Target to inform the Initiator to receive data bytes. The SCSI bus phase lines are set to the Data-In Phase and data bytes are transferred from the SCSI FIFO to the SCSI bus.

5.3.5.4 Disconnect Steps Command (Command Code 23H/A3H)

The Disconnect Steps Command is used by the Target to disconnect from the SCSI bus. This command is executed in two steps. In the Message In phase, the Target sends two bytes of the Save Data Pointers commands. Following transmission, the Target disconnects from the SCSI bus. Successful Operation and Disconnected bits are set in the Interrupt Status Register (INSTREG) upon command completion. If $\overline{\text{ATN}}$ signal is asserted by the Initiator then Successful Operation and Service Request bits are set in the INSTREG, the Command Register (CMDREG) is cleared and Disconnect Steps Command terminates without disconnecting.

5.3.5.5 Terminate Steps Command (Command Code 24H/A4H)

The Terminate Steps Command is used by the Target to disconnect from the SCSI bus. This command is executed in three steps. While in Status phase, the Target first sends a 1 byte status message. Following the Status phase the Target moves to the Message In phase and sends another 1 byte message. Lastly, the Target disconnects from the SCSI bus. The Disconnected bit is set in the Interrupt Status Register (INSTREG) upon command completion. If $\overline{\text{ATN}}$ signal is asserted by the Initiator, then the Successful Operation and Service Request bits are set in the INSTREG, an interrupt is generated and the Command Register (CMDREG) is cleared and Terminate Steps Command terminates without disconnecting.

5.3.5.6 Target Command Complete Steps Command (Command Code 25H/A5H)

The Target Command Complete Steps Command is used by the Target to inform the Initiator of a linked command completion. This command consists of two steps. In the first step, the Target sends one status byte to the Initiator in the Status Phase. The Target then sends one message byte to the Initiator in the Message In Phase. The Successful Operation bit is set in the Interrupt Status Register (INSTREG) upon command completion. If $\overline{ATN}$ signal is asserted by the Initiator, then the Successful Operation and Service Request bits are set in the INSTREG, the Command Register (CMDREG) is cleared and Target Command Complete Steps Command terminates prematurely.

5.3.5.7 Disconnect Command (Command Code 27H)

The Disconnect Command is used by the Target to disconnect from the SCSI bus. All SCSI bus signals except $\overline{SCSIRST}$ are released and the device returns to the Disconnected state. The $\overline{SCSIRST}$ signal is driven active for about 25 ms (depending on clock frequency and clock factor). Interrupt is not generated to the microprocessor.

5.3.5.8 Receive Message Steps Command (Command Code 28H/A8H)

The Receive Message Steps Command is used by the Target to request message bytes from the Initiator. The Target receives the message bytes from the Initiator while the SCSI bus is in the Message Out Phase. The Successful Operation bit is set in the Interrupt Status Register (INSTREG) upon command completion. If $\overline{ATN}$ is asserted by the Initiator, the Successful Operation and Service Request bits are set in the INSTREG, and the Command Register (CMDREG) is cleared. But if a parity error is detected, the Am53C974A ignores the received message bytes until the $\overline{ATN}$ signal is deasserted. Then the Successful Operation bit is set in the INSTREG, and the CMDREG is cleared.

5.3.5.9 Receive Commands Command (Command Code 29H/A9H)

The Receive Commands Command is used by the Target to request command bytes from the Initiator. The Target receives the command bytes from the Initiator while the SCSI bus is in the Command Phase. The Successful Operation bit is set in the Interrupt Status Register (INSTREG) upon command completion. If the $\overline{ATN}$ signal is asserted by the Initiator, then the Successful Operation and Service Request bits are set in the INSTREG, the Command Register (CMDREG) is cleared, and the command terminates prematurely. If a parity error is detected, the device continues to receive command bytes until the transfer is complete. However, if the Abort on Command Data/Parity Error (ACDPE) bit in Control Register Two (CNTLREG2) bit is set, the command is terminated immediately. The Parity Error (PE) bit in the Status Register (STATREG) is set and CMDREG is cleared.

5.3.5.10 Receive Data Command (Command Code 2AH/AAH)

The Receive Data Command is used by the Target to request data bytes from the Initiator. During this command the Target receives the data bytes from the Initiator while the SCSI bus is in the Data-Out Phase. The Successful Operation bit is set in the Interrupt Status Register (INSTREG) upon command completion. If $\overline{ATN}$ signal is asserted by the Initiator then Successful Operation and Service Request bits are set in the INSTREG, the Command Register (CMDREG) is cleared and the command terminates prematurely. If a parity error is detected, the device continues to receive data bytes until the transfer is complete (Abort on Command/Data Parity Error (ACDPE) bit in Control Register Two (CNTLREG2) is reset). If the ACDPE bit is set, the command is terminated immediately. The Parity Error (PE) bit in the Status Register (STATREG) is set and CMDREG is cleared.

5.3.5.11 Receive Command Steps Command (Command Code 2BH/ABH)

The Receive Command Steps Command is used by the Target to request command information bytes from the Initiator. During this command the Target receives the command information bytes from the Initiator while the SCSI bus is in the Command Phase.

The Target device determines the command block length from the first byte. If an unknown length is received, the Start Transfer Count Register (STCREG) is loaded with five and the Group Code Valid (GCV) bit in the Status Register (STATREG) is reset. If a valid length is received, the STCREG is loaded with the appropriate value and the GCV bit in the STATREG is set. If $\overline{\text{ATN}}$ signal is asserted by the Initiator then the Service Request bit is set in the Interrupt Status Register (INSTREG), and the Command Register (CMDREG) is cleared. If a parity error is detected, the command is terminated prematurely and the CMDREG is cleared.

5.3.5.12 DMA Stop Command (Command Code 04H)

The DMA Stop Command is used by the Target to allow the microprocessor to terminate a Target data transfer due to a lack of activity on the DMA channel. This command is executed from the top of the command queue. If there is a queued command waiting execution, it will be overwritten and the Illegal Operation Error (IOE) bit in the Status Register (STATREG) will be set. This command is cleared from the command queue once it is decoded.

Caution must be exercised when using this command since the removal of DREQ by this command may confuse the system DMA controller. Verify this condition before proceeding further.

The DMA Stop Command may be executed when all of the following conditions are satisfied:

- DMA Target Send Data Command or DMA Target Receive Data Command is in execution. In both cases the DMA controller and the Am53C974A must be in a steady state.
- During a DMA Target Send Data Command: the FIFO is empty or the Current FIFO (CF 4:0) bits in the Current FIFO/Internal State Register (CFISREG) are zero.
- During a DMA Synchronous Target Receive Data Command: the Current Transfer Count Register (CTCREG) is zero, (indicated by the Count to Zero (CTZ) bit of the Status Register (STATREG)), or the Synchronous Offset Register (SOFREG) has reached its maximum value (indicated by the Synchronous Offset Flag (SOF) bit of the Internal State Register (ISREG)).
- During a DMA Asynchronous Target Receive Data Command: the FIFO is full (CF 4:0 set to '1' in the Current FIFO/Internal State Register (CFISREG)), or Current Transfer Count Register (CTCREG) is zero (indicated by the Count to Zero (CTZ) bit of the Status Register (STATREG)).

When these conditions are satisfied, the Am53C974A halts, asserts a DMA request to its internal DMA engine, and then waits for the DMA channel. If the Am53C974A halted during Synchronous Transfer, the ACK pulses not received from the SCSI bus remain outstanding.

Upon receipt of the DMA Stop Command, the Am53C974A resets the DMA interface to its internal DMA engine and then terminates the command in progress. Ongoing SCSI sequences are completed as follows:

- Synch Data Send: completes when CTZ bit in Status Register is '1'.
- Synch Data Receive: when all outstanding ACKs received, command completes
- Asynchronous Data Send: immediately completes
- Asynchronous Data Receive: immediately completes. Remaining data in FIFO should be removed by host.

The host is interrupted only by the command that was in progress. Other bits in the Interrupt and Status Registers are left untouched, and the Command Register is cleared.

5.3.5.13 Access FIFO Command (Command Code 85H)

The host may issue the Access FIFO command following a Target Abort DMA or Abort due to parity error. This command will give the DMA controller access to the data remaining in the SCSI FIFO.

5.3.6 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.

5.3.6.1 Reselect Steps Command (Command Code 40H/C0H)

The Reselect Steps Command is used by the Target device to reselect an Initiator device. When this command is issued, the device arbitrates for the control of the SCSI bus. If the device wins arbitration, it Reselects the Initiator device and transfers a single byte identify 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 one. If DMA is not enabled, the single byte identify message 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 sequence terminates normally, a Successful Operation interrupt will be issued. This command also resets the Internal State Register (ISREG).

5.3.6.2 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.

5.3.6.3 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 a 1-byte message followed by 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 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.

5.3.6.4 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 and interrupts the host processor. $\overline{\text{ATN}}$ 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.

5.3.6.5 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 Am53C974A 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.

5.3.6.6 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 Am53C974A also generates a Selected or Reselected interrupt when the sequence is complete.

5.3.6.7 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 Am53C974A 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 three message bytes followed by 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 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.

5.3.6.8 Reselect with ATN3 Steps Command (Command Code 47H/C7H)

The Queue Tag feature of the Select with ATN3 command has been implemented in the Reselection command. Therefore, a Target reselecting an Initiator can use the QTAG feature of ATN3. Following Reselection, one message byte and 2 QTAG bytes will be sent. The three message bytes must be loaded into the FIFO before this command is issued if DMA is not enabled.

5.3.7 General Commands

5.3.7.1 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.

5.3.7.2 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.

5.3.7.3 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.

5.3.7.4 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

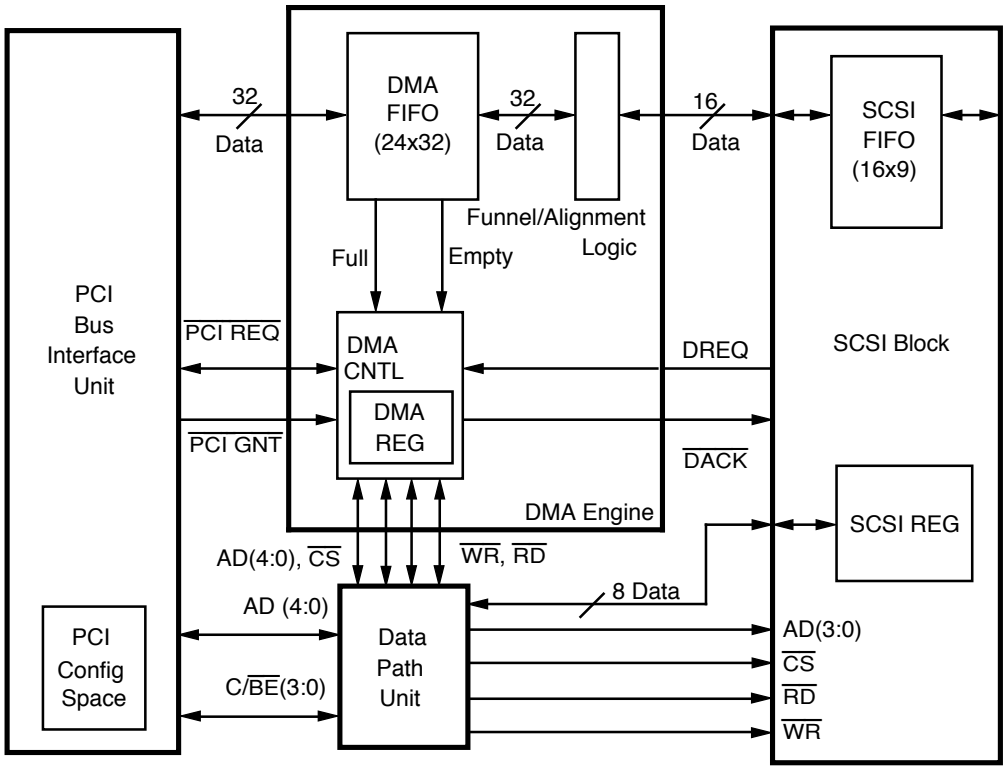
6 DMA ENGINE

6.1 INTRODUCTION

The DMA Engine in the Am53C974A 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



19113A-23

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(6:0) and C/$\overline{BE}$ (3:0) inputs from the PCI bus through the PCI Bus Interface Unit, and routes appropriate addresses, $\overline{RD}$, $\overline{WR}$, and $\overline{CS}$ control lines to either the DMA engine or the SCSI block, depending on the state of the AD6 address line. If AD6 is '0', then register accesses are to the SCSI registers. However, if AD6 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 $\overline{DACK}$ 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 ($\overline{INTA}$ 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*

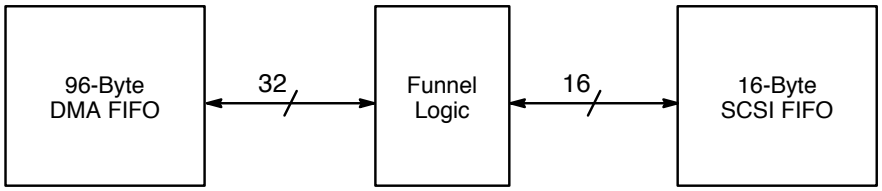
3. The DMA BLAST command is executed by writing '01' to bits 1:0 in the DMA Command Register. When issued, this command will move the DMA FIFO contents into the system memory.
4. Completion of the BLAST operation is signaled when bit 5 in the DMA Status Register is set to '1'.
5. Set the DMA engine to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



19113A-24

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 Am53C974A is stored at register address (10h) in the PCI Configuration Space.

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
SBAC	(B)+70	SCSI Bus and Control	*

**Certain bits are Read/Write; certain bits are Read Only. Refer to the SBAC register description for more details.*

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 = '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 through 4 must be programmed as they were during the first PCI I/O write cycle, and must be preserved throughout the operation. 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	0	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 of a DMA transfer (DONE condition), or receipt of an error condition during a DMA transfer. This bit is cleared by a hard reset.

Bit 5 – Reserved

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

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 (Bits 11:0), will contain the offset for the first of the translated MDL entries. When this bit is '0', the Am53C974A will operate in non-MDL mode, in which case the SPA register will contain actual 32-bit physical memory addresses. This bit is cleared by a hard reset.

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 Am53C974A 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 this value are considered NOPs if there are no DMA operations in progress. Issuing this command while a DMA transaction is in progress will reset the DMA Engine to the IDLE state and halt the current transfer, INTA will not be asserted.
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 only after the DMA transfer has completed. 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 are 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 only after the DMA transfer has completed.

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 Register (SPA, Register (B)+48) 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	PABORT	BCMBLT	SCSINT	DONE	ABORT	ERROR	PWDN
0	0	0	0	0	0	0	X

Bit 31:8 – Reserved

These bits are reserved for future implementation.

Bit 7 – Reserved

This bit is reserved and read as zero.

Bit 6 – PABORT – PCI Master/Target Abort

This bit is used in conjunction with bit 25 in register (B)+70h to indicate that a PCI Master or Target abort condition was detected. When bit 25 is set to '1' and a PCI abort condition is detected, this bit will be set to '1'. When bit 25 is reset to '0', detection of an abort condition will not be reported by this bit. PABORT is read-only and is cleared when read if bit 24 (DMA Status Write Erase control) in register (B)+70h is reset to '0'. If the Write Erase feature is set, PABORT is cleared when a '1' is written to this location.

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 in conjunction with bit 6 (INTE_D) in the DMA Command Register ((B)+40h), and the Working Byte Count (WBC23:0) is zero, an interrupt is generated. Reading this bit will clear it and its associated interrupt. Refer to the section 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 register is read, this bit will be cleared.

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'.

6.7.9 SCSI Bus and Control (SBAC) Address (B)+70h READ ONLY

Control (CNTRL)
Address (B)+70

This is a 32-bit register which is used to control the enhancements in the Am53C974A. These enhancements inclued the Write Erase feature for the DMA Status register, the SCSI Powerdown and clock selection feature, and the SCAM support.

31	30	29	28	27	26	25	24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	PABTEN	STATUS
0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16
Reserved	Reserved	PWD	SBSY	SCLK	SCAM	REQ	ACK
0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8
RST	BSY	SEL	ANT	MSG	C/D	I/O	SDP
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0
0	0	0	0	0	0	0	0

Bit 31:26 – Reserved

These bits are reserved and will return undefined values when read.

Bit 25 – PABTEN – Enable Interrupt on PCI Abort

This bit controls the interrupt and status bit for the PCI Abort condition (Master or Target). When this bit is set to '1' and a PCI Abort condition is detected, an interrupt will be generated, and bit 6 of the DMA Status register ((B)+54h) will be set to '1'. When this bit is reset to '0' and a PCI Abort condition is detected, an interrupt will not be generated and bit 6 in the DMA Status register will be '0'.

Bit 24 – STATUS – Write Erase Control

This bit controls the Write Erase feature on bits 3:1 and bit 6 of the DMA Status Register ((B)+54h). When this bit is programmed to '1', the state of bits 3:1 are preserved when read. Bits 3:1 are only cleared when a '1' is written to the corresponding bit location. For example, to clear bit 1, the value of '0000_0010b' should be written to the register. When the DMA Status Preserve bit is '0', bits 3:1 are cleared when read.

Bit 23:22 – Reserved

These bits are reserved for internal testing and must always be set to '0'.

Bit 21 – PWD – Power Down SCSI Core

This bit is used to remove the SCSI clock input to the Am53C974A during power-down sequencing. When this bit is '1', the SCSI clock input is disconnected from the SCSI core to save power. When this bit is '0', the clock is connected for SCSI operation.

Bit 20 – SBSY – SCSI Busy

This is a read only bit which monitors the BUS FREE condition on the SCSI bus. When this bit is '1' , the SCSI bus is in use. When this bit is '0', the SCSI bus is free. This bit is used in conjunction with the PWD bit for power-down mode.

Bit 19 – SCLK – Select Clock

This bit is read-only and indicates whether the SCSI core is connected to the PCI clock, or if it is connected to an external SCSI clock source. When an external SCSI clock is not provided on pin 60 (SCSICLK1), the Am53C974A will automatically use the PCI clock pin to drive the SCSI logic. Under this condition, this bit will be read as '0'. When an external SCSI clock is present, the SCSI logic will automatically be driven by the SCSI clock and this bit will be read as '1'.

Note: *For Fast Synchronous (10 MBytes/sec) operation, the SCSI core must be driven by a 40 MHz source. This requires an external SCSI clock source since the PCI clock operates at 33 MHz or less.*

Bit 18 – SCAM – SCSI Configure AutoMagically Mode

This bit is used to enable or disable SCAM mode. When this bit is set to '1', bits 17:0 may be programmed to support SCAM protocol. When in SCAM mode, bits 17:0 are read/write and directly affect the logical state of the SCSI bus. For example, if SCAM is enabled and bit 17 (REQ) is set to '1', the $\overline{\text{REQ}}$ line on the SCSI bus will go low. When bit 18 is reset to '0', bits 17:0 are read-only and represent the logical state of the SCSI bus.

Bit 17 – REQ – Request

When the SCAM bit is enabled (bit 18 set to '1'), this bit is read/write and logically affects the state of the $\overline{\text{REQ}}$ line on the SCSI bus. When SCAM is disabled (bit 18 reset to '0'), this bit is read-only and represents the logical state of the $\overline{\text{REQ}}$ line.

Bit 16 – ACK – Acknowledge

When the SCAM bit is enabled (bit 18 set to '1'), this bit is read/write and logically affects the state of the $\overline{\text{ACK}}$ line on the SCSI bus. When SCAM is disabled (bit 18 reset to '0'), this bit is read-only and represents the logical state of the $\overline{\text{ACK}}$ line.

Bit 15 – RST – Reset

When the SCAM bit is enabled (bit 18 set to '1'), this bit is read/write and logically affects the state of the $\overline{\text{RST}}$ line on the SCSI bus. When SCAM is disabled (bit 18 reset to '0'), this bit is read-only and represents the logical state of the $\overline{\text{RST}}$ line.

Bit 14 – BSY – Busy

When the SCAM bit is enabled (bit 18 set to '1'), this bit is read/write and logically affects the state of the $\overline{\text{BSY}}$ line on the SCSI bus. When SCAM is disabled (bit 18 reset to '0'), this bit is read-only and represents the logical state of the $\overline{\text{BSY}}$ line.

Bit 13 – SEL – Select

When the SCAM bit is enabled (bit 18 set to '1'), this bit is read/write and logically affects the state of the $\overline{\text{SEL}}$ line on the SCSI bus. When SCAM is disabled (bit 18 reset to '0'), this bit is read-only and represents the logical state of the $\overline{\text{SEL}}$ line.

Bit 12 – ATN – Attention

When the SCAM bit is enabled (bit 18 set to '1'), this bit is read/write and logically affects the state of the $\overline{\text{ATN}}$ line on the SCSI bus. When SCAM is disabled (bit 18 reset to '0'), this bit is read-only and represents the logical state of the $\overline{\text{SEL}}$ line.

Bit 11 – MSG – Message

When the SCAM bit is enabled (bit 18 set to '1'), this bit is read/write and logically affects the state of the $\overline{\text{MSG}}$ line on the SCSI bus. When SCAM is disabled (bit 18 reset to '0'), this bit is read-only and represents the logical state of the $\overline{\text{MSG}}$ line.

Bit 10 – C/D – Command/Data

When the SCAM bit is enabled (bit 18 set to '1'), this bit is read/write and logically affects the state of the $\overline{\text{C}}$/D line on the SCSI bus. When SCAM is disabled (bit 18 reset to '0'), this bit is read-only and represents the logical state of the $\overline{\text{C}}$/D line.

Bit 9 – I/O – Input/Output

When the SCAM bit is enabled (bit 18 set to '1'), this bit is read/write and logically affects the state of the $\overline{\text{I}}$/O line on the SCSI bus. When SCAM is disabled (bit 18 reset to '0'), this bit is read-only and represents the logical state of the $\overline{\text{I}}$/O line.

Bit 8 – SDP – SCSI Data Parity

When the SCAM bit is enabled (bit 18 set to '1'), this bit is read/write and logically affects the state of the $\overline{\text{SDP}}$ line on the SCSI bus. When SCAM is disabled (bit 18 reset to '0'), this bit is read-only and represents the logical state of the $\overline{\text{SDP}}$ line.

Bit 7:0 – SD7:0 – SCSI Data

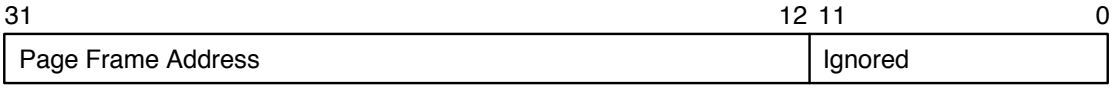
When the SCAM bit is enabled (bit 18 set to '1'), these bits are read/write and logically affect the state of the $\overline{\text{SD}}$7:0 lines on the SCSI bus. When SCAM is disabled (bit 18 reset to '0'), this bit is read-only and represents the logical state of the $\overline{\text{SD}}$7:0 lines.

6.8 DMA SCATTER-GATHER MECHANISM

The Am53C974A 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 required 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:



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

The scatter-gather mechanism described below assumes 4k page alignment and size 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).

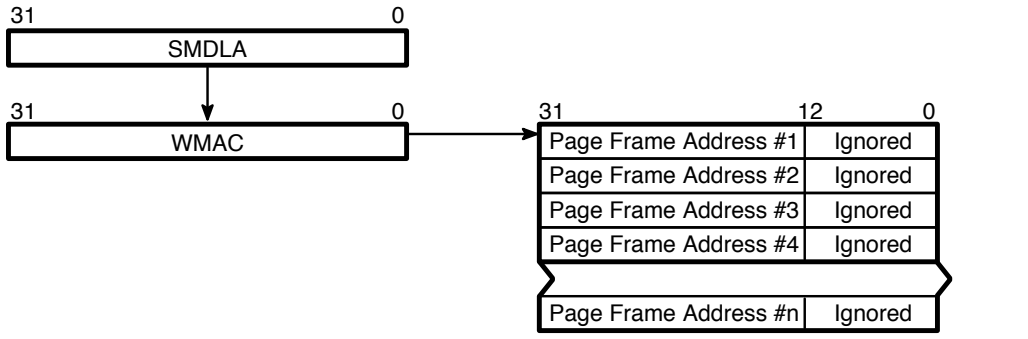
1. a) Prepare the Memory Descriptor List (MDL) through software and store it in system memory.

   b) Load the address of the starting entry in the Memory Descriptor List (MDL) 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 entry.

   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.

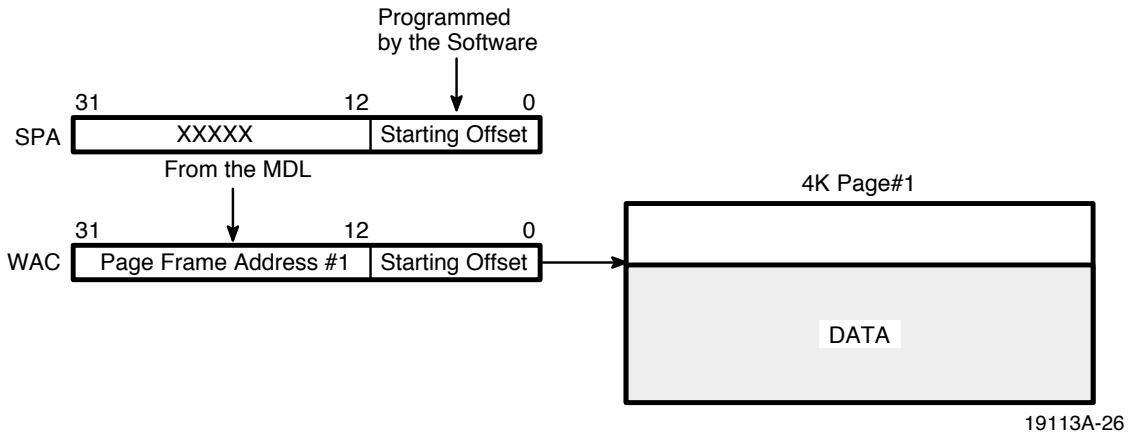
Note: *The value in the SMDLA register is double word aligned. Therefore, read/write transactions will always begin on a double word boundary.*



19113A-25

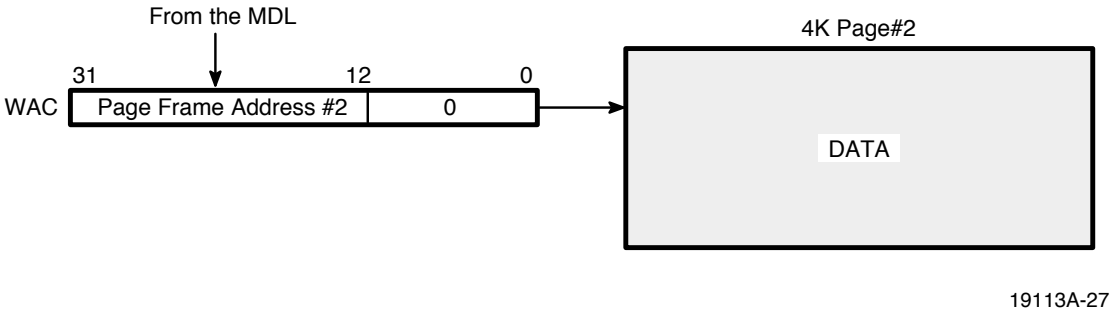
In this example, the contents of the WMAC register is pointing to page frame address #1. When the first entry in the MDL is read, the WMAC register is incremented to point to the next page entry (page frame address #2).

2. The Am53C974A reads only the page frame address (bits 31:12) from the MDL entry and combines 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.



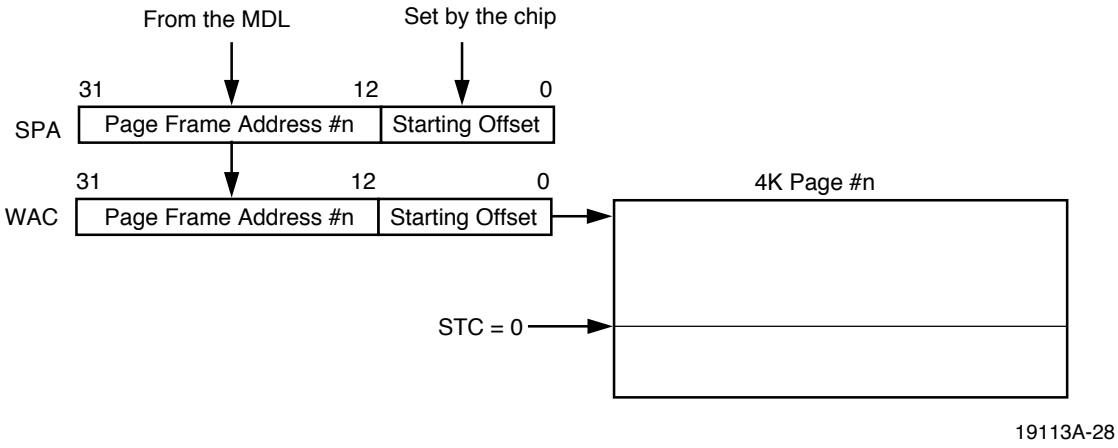
When the WAC register (bits 11:0) reaches the end of the first page, the Am53C974A reads the next MDL entry, and increments the WMAC register.

3. The chip reads the page frame address (bits 31:12) from the second entry of the MDL and combines it with the WAC register (bits 11:0). This becomes the physical address for page#2. The WAC register (bits 11:0) are '000h' since it now points to the beginning of Page# 2 as shown below.



Again when WAC (bits 11:0) reaches the maximum page length count, the Am53C974A 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.



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. If the interrupt source is from the SCSI block, the SCSI Status, Internal State, and Interrupt Status Registers should be read to obtain information about the SCSI interrupt. SCSI interrupts will only be cleared when the Interrupt Status Register is 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 PWDN pin is first asserted

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 Am53C974A has been reselected

In addition, under certain conditions, an interrupt will be generated upon receipt of a PCI parity error.

The following matrix explains the conditions which cause a PCI parity error interrupt. PCI parity error detection is controlled by the following items:

a) Bit 6, PCI Command Register at address 04h
b) Bit 6, DMA Command Register at address (B)+40h
c) The state of the $\overline{\text{PERR}}$ pin

PCI parity errors are reported in three locations:

a) Bit 15 and bit 8, PCI Status Register at address 06h
b) Bit 1, DMA Status Register at address (B)+54h
c) Interrupt asserted

Conditions of Parity Error Detection (When all Conditions are True)	**Method of Parity Error Reporting (These Bits will be Set)**		
	PCI Status Reg (06h)	**DMA Status Reg (B)+54h**	**Interrupt Generated?**
PCI CMD Reg (04h) Bit 6 = 1 and DMA CMD Reg (B)=40h Bit 6 = 0 and $\overline{\text{PERR}}$ asserted	Bit 15 = 1 Bit 8 = 1	Bit 1 = 1	No
PCI CMD Reg (04h) Bit 6 = 1 and DMA CMD Reg (B)=40h Bit 6 = 1 and $\overline{\text{PERR}}$ asserted	Bit 15 = 1 Bit 8 = 1	Bit 1 = 1	Yes

Please note that the parity error detection and reporting scheme described above is true only during DMA operations, when PCSCSI is a Master.

CHAPTER

7 EXPANSION ROM SUPPORT

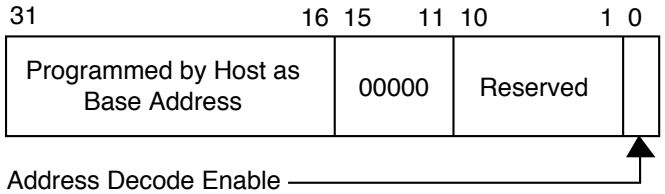
7.1 INTRODUCTION

This chapter describes the ROM support feature of the Am53C974A. The boot ROM for the Am53C974A can be implemented using a standard PROM or EPROM and one external latch. This chapter shows a sample boot ROM implementation as well as a typical ROM access cycle for clarity. In addition, the expansion ROM mapping as defined by the PCI specfication, rev 2.0 is also described briefly.

7.2 ROM BASE ADDRESS REGISTER

The Am53C974A supports expansion ROMs of up to 64 Kbytes, with speed as slow as 250 ns, if the BOOT pin (pin 100) is tied high. This is done through use of the Expansion ROM Base Address Register located at address 30h in the PCI Configuration Space and built-in logic to simplify the ROM interface. The Base Address Register is 4 bytes wide and defines the base address and size of the expansion ROM. Figure 7-1 shows the Base Address Register implemented on the Am53C974A. In this Figure, only bits 31:16 and bit 0 are programmable while bits 15:11 are hardwired to '0'. As a result, the Am53C974A specifies support for ROMs aligned to 64 Kbyte boundaries. Refer to the Expansion ROM Base Address Register for the bit-level discription.

Figure 7-1 Expansion ROM Base Address Register

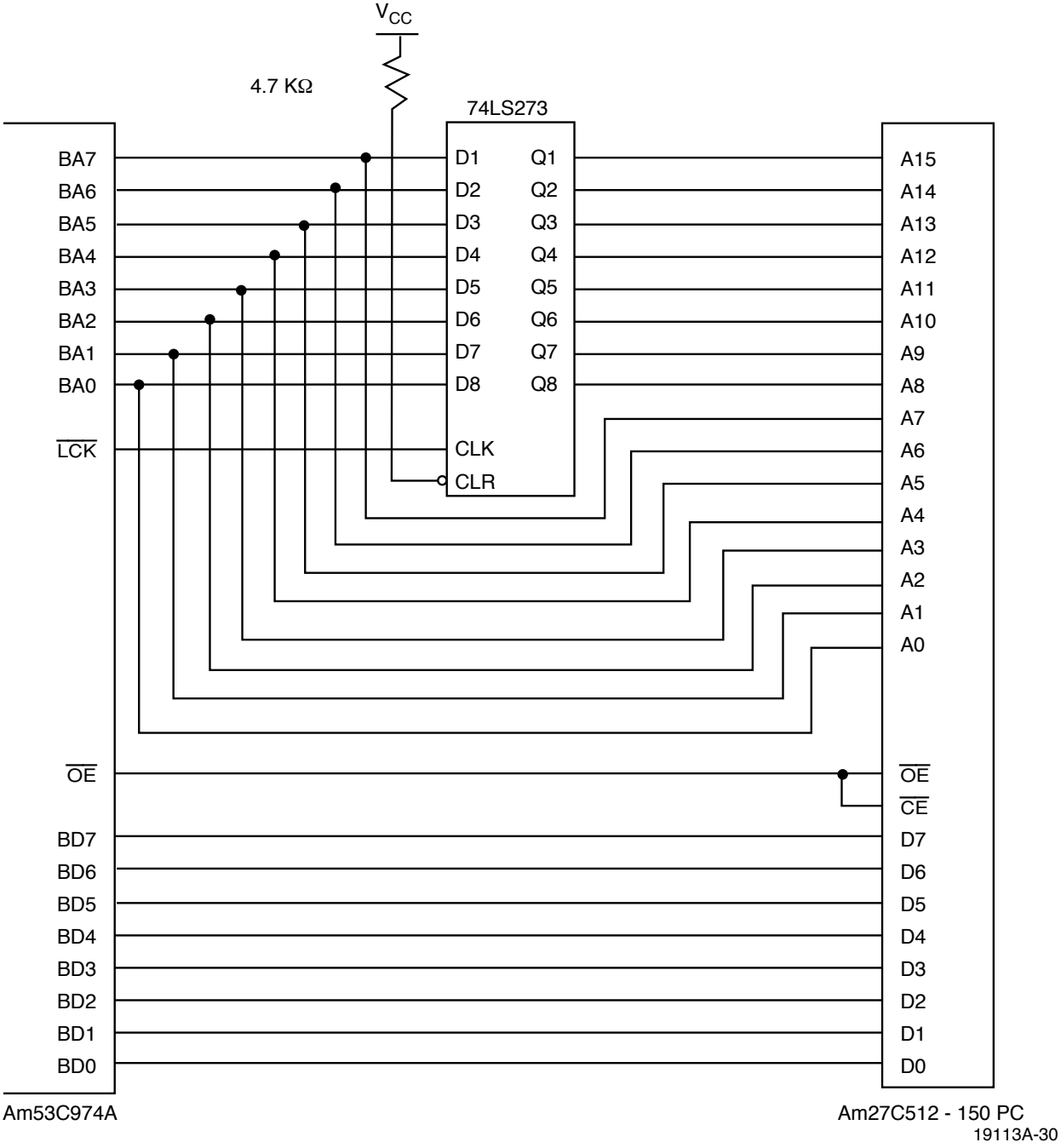


19113A-29

7.3 SAMPLE IMPLEMENTATION

Figure 7-2 shows a sample implementation of an Am53C974A to boot ROM interface. In Figure 7-2, the implementation uses an Am27C512 - 150 PC EPROM and a 74LS273 positive edge-triggered octal DFF to latch the high address byte. Note that the ROM's $\overline{CE}$ signal is tied to its $\overline{OE}$ signal to save power consumed by the ROM when it is not being accessed (t_{CE} to Data Valid is 150 ns). However, This solution will cause the ROM's access time to be slightly extended. Another method is to connect the ROM's $\overline{CE}$ to ground to minimize the access time (t_{CE} to Data Valid is 50 ns). This is done at the expense of power consumed since the ROM is always enabled.

Figure 7-2 Example of Am53C974A to ROM Interface



7.4 ROM ACCESS CYCLE

PCI Slave Memory Reads at the ROM addresses initiate the Am53C974A read sequence to the expansion ROM. Figure 7-3 shows the ROM access timing when the PCI Slave Memory Read cycle is initiated to the ROM. In this Figure, it is assumed that the ROM's $\overline{OE}$ is connected to its $\overline{CE}$. Note that the entire cycle requires 54 PCI clocks from the time $\overline{FRAME}$ is sampled asserted to the time TRDY is sampled asserted. Note that if $\overline{FRAME}$ is kept asserted by the host after the Am53C974A asserts $\overline{TRDY}$, the Am53C974A will assert $\overline{STOP}$. This is shown by the dashed lines.

Figure 7-3 PCI Slave Memory Read and ROM Access Cycle

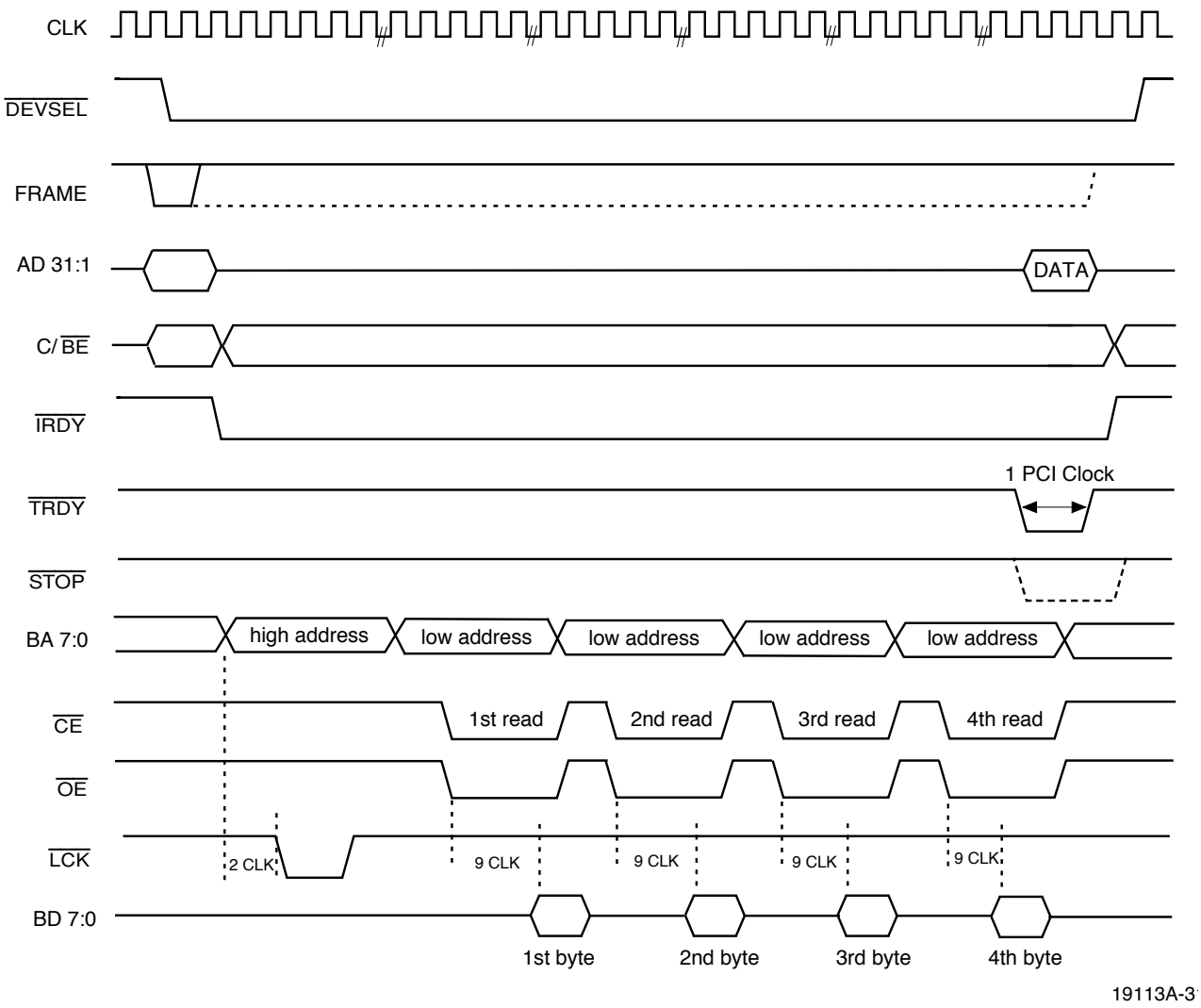
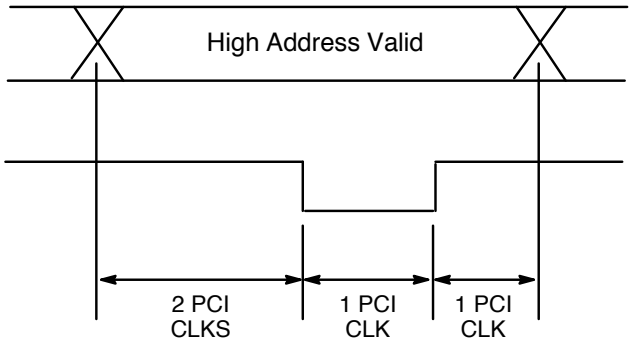


Figure 7-4 shows an expanded view of the $\overline{LCK}$ signal relative to the valid high address byte on BA7:0. Note that the time from the high address valid to the negative edge of $\overline{LCK}$ is approximately 2 PCI clocks and the time from the positive edge of $\overline{LCK}$ to the high address invalid is approximately 1 PCI clock. Assuming a PCI clock of 33 MHz (30 ns period), the recommended worst case setup and hold times for negative and positive edge triggered latches are as follows:

Latch Type	Setup Time	Hold Time
Negative Edge Triggered	55 ns	55 ns
Positive Edge Triggered	85 ns	25 ns

For example, a negative edge triggered latch with 50 ns setup and hold times will work with the Am53C974A.

Figure 7-4 Expanded Picture of $\overline{\text{LCK}}$ Relative to High Address Valid

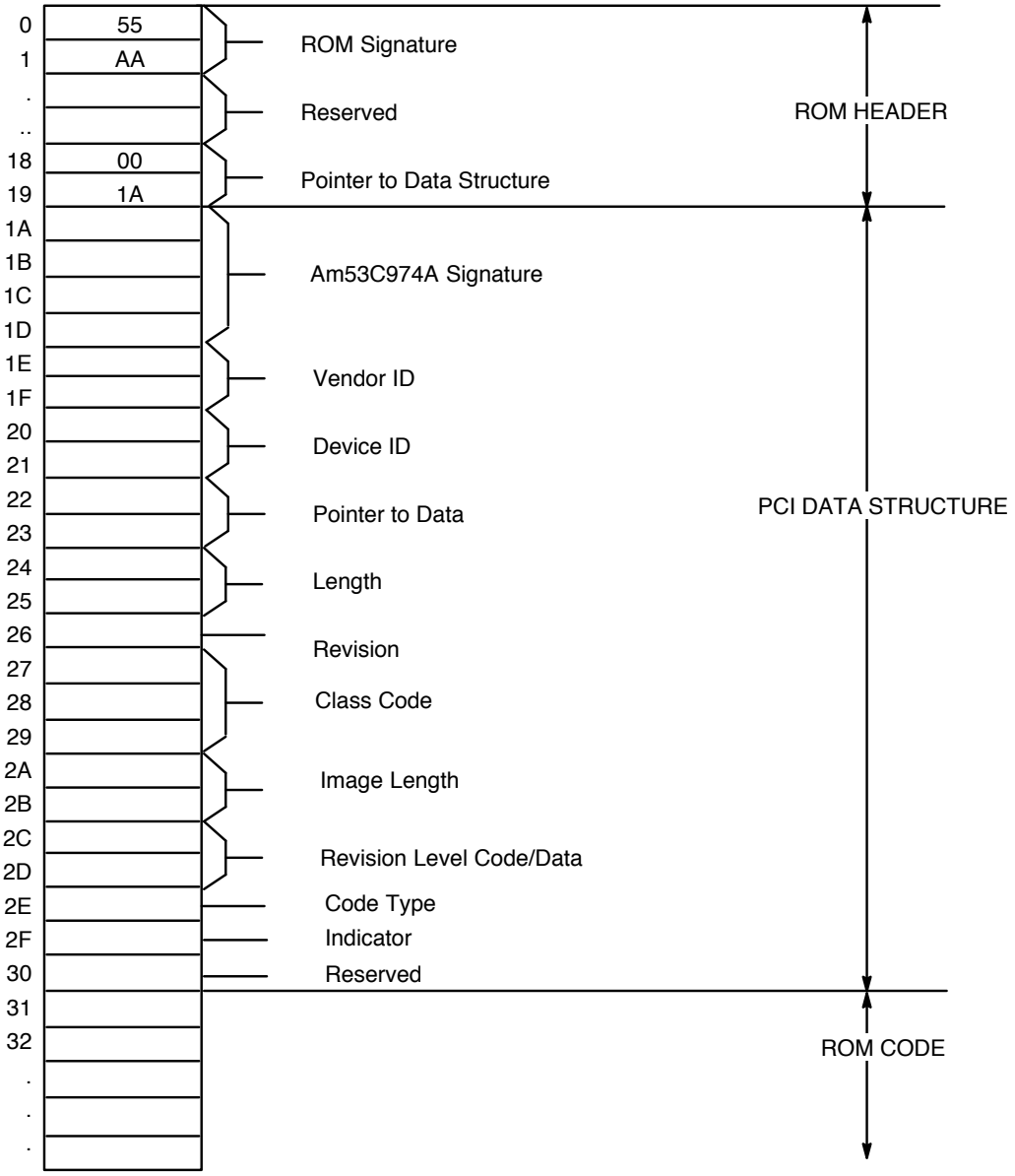


7.5 EXPANSION ROM MAPPING

The Boot ROM attached to the Am53C974A allows the host to access the ROM code at power-up. This ROM code can be accessed using the standard PCI memory read instruction. During these reads, the Am53C974A will insert wait states by keeping $\overline{\text{TRDY}}$ deasserted until the data is ready from the ROM. This will compensate for the difference in access time due to the slower ROM.

Figure 7-5 shows an example of the mapping of the PCI Expansion ROM on the Am53C974A. In this Figure, the first 19h bytes of the expansion ROM area is defined as the ROM Header. Offset 1Ah to 30h is the PCI Data Structure. The ROM code starts at offset 31h. The expansion ROM area starts at offset 00h to its maximum memory location. Users may follow the mapping shown in Figure 7-4 or they may allocate the PCI Data Structure or ROM code in some other offset address. However, the ROM Header offset must start at offset 00h.

Figure 7-5 Am53C974A Expansion ROM Mapping



19113A-33

CHAPTER

8 SCAM TUTORIAL

8.1 INTRODUCTION

The Am53C974A is designed to support Level 1 SCAM as defined in the X3T9.2/93-109r5 (Sept 30, 1993) specifications. As this specification is not yet finalized, users should treat the material in this chapter only as a tutorial and should refer to the original SCAM specification for a more detailed description of the SCAM protocol. Also note that this tutorial assumes familiarity with the SCSI protocol.

SCAM stands for SCSI Configured AutoMagically. It is a proposed protocol to assign IDs to SCSI devices after power-on or after a SCSI bus reset. Various types of devices (SCAM master devices, SCAM slave devices, SCAM tolerant devices, and SCAM intolerant devices) can be connected to a SCAM capable SCSI bus. This chapter provides a quick tutorial of the SCAM protocol.

8.2 REQUIREMENTS

In order to implement the SCAM protocol, the Am53C974A must:

(1) Perform SCAM selection

(2) Have a hard ID

(3) Recognize SCSI bus reset conditions

(4) Use selection time-out

(5) Shall not assert SCSI RST upon a selection time-out

Other requirements:

(1) The Am53C974A will be the only master on the SCSI bus

(2) All slaves will power-up concurrently or before the master

(3) The Am53C974A must work with SCAM tolerant devices, SCAM devices with default IDs, and SCAM intolerant devices

8.3 SCAM TERMINOLOGY

SCAM devices

Any SCSI device that implements the SCAM protocol.

SCAM master devices

Any SCAM device that controls the assignments of soft IDs.

SCAM slave devices

Any SCAM device that receives soft ID assignments through the SCAM protocol. If the slave device is on a SCAM capable bus it uses a soft ID. If it is on a non-SCAM bus it uses its default ID. Note that for Level 1 slave devices, once a slave device has received its soft ID or has its default ID assigned, it will act like a SCAM tolerant device.

SCAM tolerant devices

Any non-SCAM device that satisfies certain requirements on responding to SCAM selection, allowing SCAM tolerant devices to be freely intermixed with SCAM devices.

SCAM intolerant devices

Any non-SCAM device that does not respond to SCAM selection.

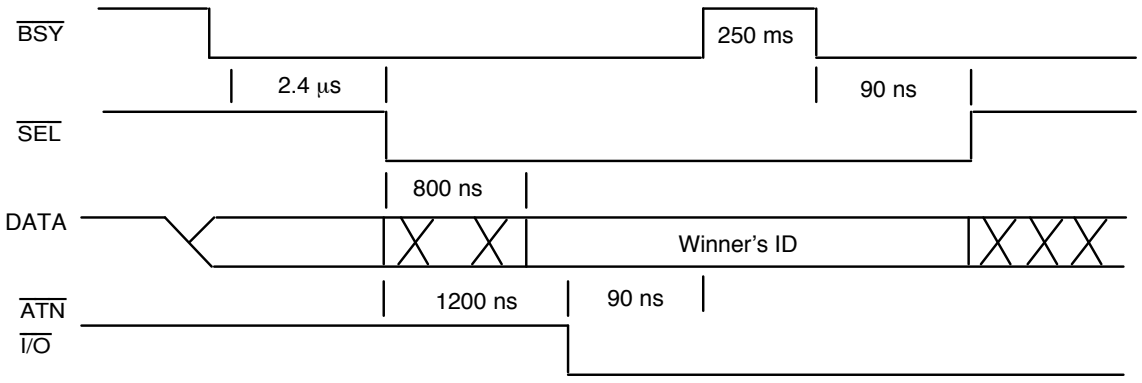
8.4 NORMAL SCSI AND SCAM SELECTIONS

Level 1 SCAM selection sequence is different from normal non-SCAM or SCSI selection sequence. The following describes normal SCSI selection and Level 1 SCAM selection.

8.4.1 Normal SCSI Selection

Figure 8-1 shows the normal SCSI arbitration and selection timing diagram. In Figure 8-1, any SCSI device may assert $\overline{BSY}$ with its own ID on the data bus when there is a bus free phase. After a minimum of 2.4 μs arbitration delay, all participating devices will examine the bus to determine if they have won ownership (the winner is the device with the highest ID). At this point, all losers must release all SCSI bus signals within a bus clear delay of 800 ns. Only the winning device will then assert $\overline{SEL}$ and the target ID with whom it wants to communicate along with other signals during this selection phase. Two de-skew delays after other signals such as $\overline{ATN}$ or $\overline{I/O}$ have been asserted, the selecting device will release $\overline{BSY}$. One selection time-out period (400 ns min) later, the selected target will assert $\overline{BSY}$. Two de-skew delays later, the selecting device will release $\overline{SEL}$, completing the selection process.

Figure 8-1 Normal SCSI Arbitration and Selection Sequence



Arbitrate and win **----** put ID on bus.
wait minimum of 2.4 μs.
assert SEL if ID is highest.

Arbitrate and lose **--** put ID on bus.
any time before 2.4 μs, if ID not the highest or external SEL is on, then terminate by releasing all SCSI signals.

19113A-34

8.4.2 Level 1 SCAM Selection

Figure 8-2 shows the timing sequence of SCAM devices in the SCAM selection process. The SCAM power-on to SCAM selection time (1 s) is the maximum time a SCAM device may wait after power-on before enabling its response to selection. The SCAM reset to

SCAM selection time (250 ms) is the minimum time a SCAM device may wait after a reset condition has occurred before initiating the SCAM protocol.

Once the SCAM device is ready to respond to SCAM selection, it may respond with a long or short selection time. When slow response is anticipated, a SCAM device will maintain a long SCAM selection time (minimum 250 ms). When rapid response is anticipated, a SCAM device will maintain a short SCAM selection time (minimum 1 ms). The SCAM default ID selection response time (4 ms) is the minimum time in which a SCAM slave device may respond to selection of its unconfirmed default ID. This is also the maximum selection time-out delay a SCAM master will use when examining the bus for SCAM tolerant devices.

Figure 8-2 Timing Diagram Showing the SCAM Device Selection Sequence

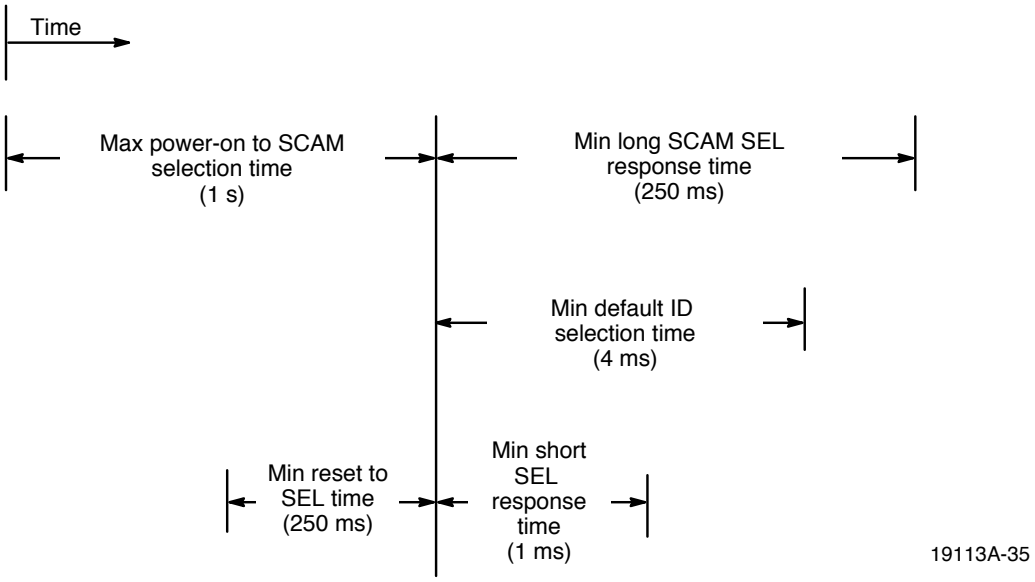


Figure 8-3 shows the timing sequence of SCAM tolerant devices in a SCAM selection process. The SCAM tolerant power-on to SCAM selection time (5 s) is the maximum time a SCAM tolerant device may wait after power-on before enabling its response to selection. The SCAM tolerant reset to SCAM selection time (250 ms) is the maximum time a SCAM tolerant device may wait after a reset condition has occurred before enabling its response to selection.

The SCAM tolerant selection response time (1 ms) is the maximum time in which a SCAM tolerant device may respond to selection of its ID. This is also the minimum selection time-out delay a SCAM master can use when examining the bus for SCAM tolerant devices.

Figure 8-3 Timing Diagram Showing the SCAM Tolerant Devices

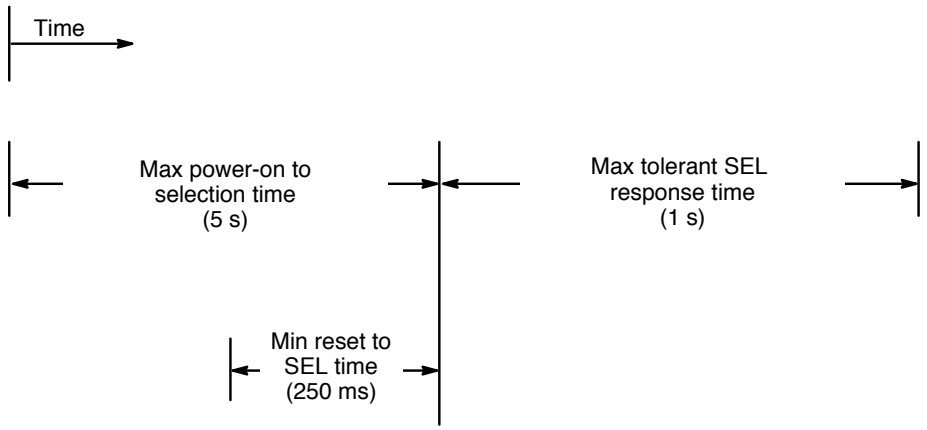
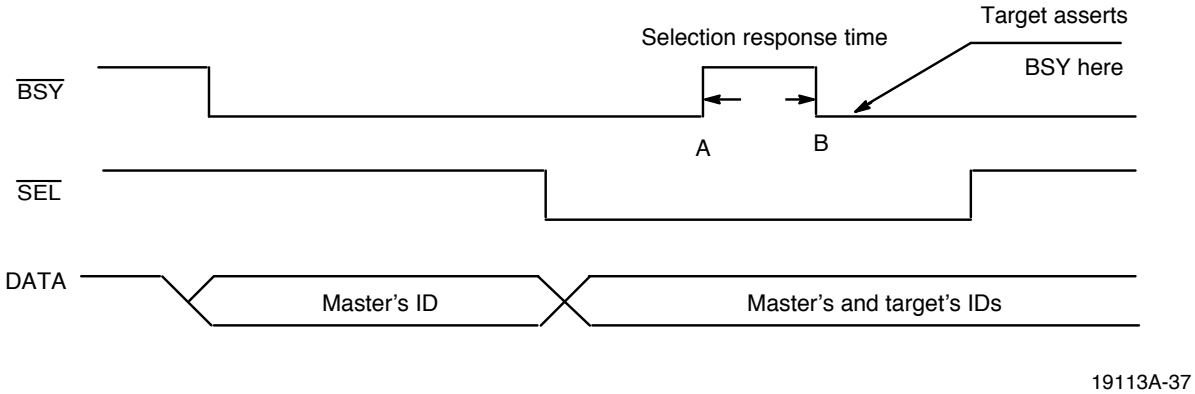


Figure 8-4 shows the Level 1 SCAM selection timing diagram to demonstrate the relation of the parameters specified in Figure 8-2 and 8-3.

The maximum power-on to SCAM selection time of 1 s (SCAM devices) or 5 s (SCAM tolerant devices) refers to the time it takes from power-on to bring up the system to point A in Figure 8-4. For example, when a SCAM master selects a SCAM tolerant device, the SCAM master may release $\overline{BSY}$ (point A) five seconds after the SCAM tolerant device has powered-on. The SCAM tolerant device will recognize the selection and respond within 1 ms by asserting $\overline{BSY}$ (Point B).

Similarly, when a SCAM master device selects a SCAM slave device, the SCAM master may release $\overline{BSY}$ 1 s after the SCAM slave device has powered-on. The SCAM slave device will respond with its default ID by asserting $\overline{BSY}$ at least 4 ms after the master has released $\overline{BSY}$.

Figure 8-4 Level 1 SCAM Arbitration and Selection Timing



It is the difference in selection response time which distinguish between SCAM slave and SCAM tolerant devices.

8.5 SCAM PROTOCOL

The following describes the SCAM protocol for SCAM master and SCAM slave devices.

8.5.1 SCAM Master Device Operation

Figure 8-5 shows the flow diagram for Level 1 SCAM master devices. Whenever the SCAM master is powered-on, it should generate a SCSI bus reset. Once this has occurred, the SCAM master will initialize its internal table of IDs to indicate that all IDs are uncategorized (initialize state). The SCAM master will then proceed with categorizing IDs by winning arbitration and selecting an uncategorized ID using a selection time-out delay less than the SCAM default ID selection response time. This process is repeated until all IDs are categorized. ID categorization is done in four ways:

(1) The SCAM master shall categorize it's own ID as 'in use'.

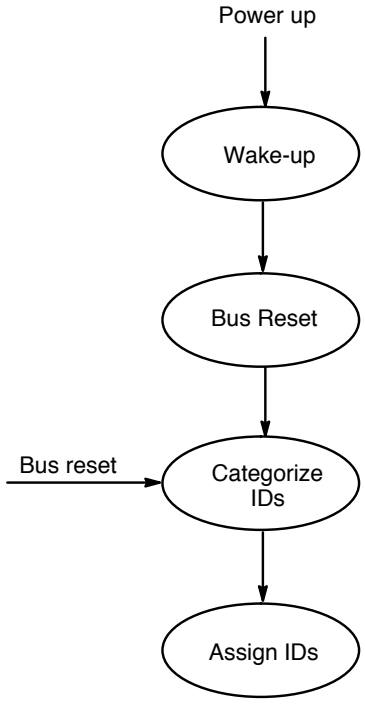
(2) If a target device responds to selection of an uncategorized ID by asserting $\overline{\text{BSY}}$, the master will categorized that ID as 'in use'.

(3) If no device responds to selection of an uncategorized ID within a SCAM tolerant selection response time, and the Selection Phase began later than both a SCAM tolerant power-on to selection time following the master's most recent power-on and a SCAM tolerant reset to selection time following the most recent reset condition, then the master may categorize that ID as 'not in use'.

(4) The target may categorize IDs through non-SCAM means such as configuration parameters.

After categorizing IDs, the SCAM master should initiate the SCAM protocol and assign IDs. A SCAM master may initiate the SCAM protocol as often as it wishes. For example the master should subsequently initiate the SCAM protocol if it determines that a SCAM slave device has just been powered-on or reset.

Figure 8-5 Level 1 SCAM Master Protocol



19113A-38

8.5.2 SCAM Slave Device Operation

Figure 8-6 shows the flow diagram for Level 1 SCAM slave devices. Following a SCSI bus reset or power-on, a SCAM slave device will cease responding to its former ID and perform local initialization (Wake-up state). After completing this, the SCAM slave device will monitor the bus for selection or reselection of its default ID (if any), or for SCAM selection.

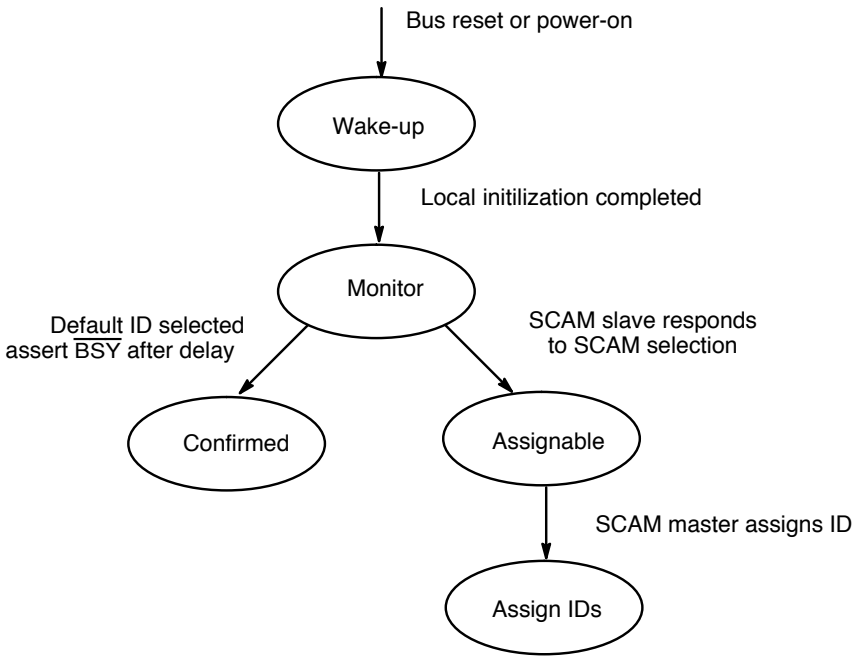
A SCAM slave device will enter the Monitor state no later than a SCAM power-on to SCAM selection time following power-on. Similarly, the device will enter the Monitor state no later than a SCAM reset to SCAM selection time following a bus reset condition.

If a SCAM slave device detects selection or reselection of its default ID while in the Monitor state, the device must wait a minimum of a SCAM default ID selection response time before asserting $\overline{\text{BSY}}$. If the selection or reselection phase persists for longer than that time, the SCAM slave device will assert $\overline{\text{BSY}}$. The SCAM slave device will then confirm its default ID (Confirmed state) only if it asserted $\overline{\text{BSY}}$ in response to selection or reselection of its default ID. A device that has confirmed its default ID will ignore SCAM selection and will respond to subsequent selection or reselection of its default ID within a SCAM tolerant selection response time. The device's default ID remains confirmed until the next bus reset condition or loss of power.

If a SCAM slave device detects SCAM selection within a SCAM default ID selection response time while in the Monitor state, the slave device will accept an ID assigned by the SCAM master.

Upon receiving a soft ID assignment (Assigned state), a SCAM slave device will ignore SCAM selection and will respond to subsequent selection or reselection of its soft ID within a SCAM tolerant selection response time. The device will continue to use its soft ID until the next bus reset condition or loss of power.

Figure 8-6 Level 1 SCAM Slave Protocol



19113A-39

8.6 SCAM EXAMPLES

The following examples illustrate SCAM operation.

Example 1

A SCSI bus has one SCAM master (The Am53C974A) with ID of 1, one SCAM tolerant device with ID of 4, and one SCAM slave device with default ID of 2. A new soft ID will be assigned to the SCAM slave device as shown.

(1) SCAM tolerant and SCAM slave devices power-on before or concurrently with the Am53C974A.

(2) The Am53C974A arbitrates for the bus and selects device with ID of 2. The Am53C974A will not release $\overline{\text{BSY}}$ until 5 seconds after the SCAM tolerant or SCAM slave device has powered-up.

(3) The Am53C974A deasserts and monitors $\overline{\text{BSY}}$. Within 1 ms, no device on the bus asserts $\overline{\text{BSY}}$. At 5 ms, a device responds by asserting $\overline{\text{BSY}}$. Therefore, device with ID of 2 is a SCAM slave device (default ID confirmed).

(4) The Am53C974A repeats (2) and (3) with target ID of 4. A device asserts $\overline{\text{BSY}}$ within 1 ms. Therefore, device with ID of 4 is a SCAM tolerant device.

(5) The Am53C974A repeats (2) and (3) with other target IDs but no device on the bus asserts $\overline{\text{BSY}}$. Thus the Am53C974A has finished categorizing the bus IDs.

(6) The Am53C974A initiates SCSI bus reset.

(7) The Am53C974A arbitrates for the bus.

(8) At least 250 ms later, the Am53C974A initiates the SCAM protocol and selects the SCAM slave device without ID.

(9) The Am53C974A assigns soft ID to the SCAM slave device.

Example 2

Same as example 1 except there is one more SCAM intolerant device with ID of 6. The selection response time is 3 ms. This can be up to 250 ms maximum according to the SCSI specification.

(1) SCAM master repeat (1) through (3) in Example 1 to categorize IDs for SCAM tolerant devices.

(2) The Am53C974A scans for SCAM slave devices by detecting a response within 4 ms. The Am53C974A detects SCAM slave device with ID of 2, and a SCAM intolerant device with ID of 6.

(3) The Am53C974A initiates a SCSI bus reset.

(4) The Am53C974A assigns a soft ID to the SCAM slave device with a value other than its default ID of 2.

(5) The Am53C974A assigns a soft ID to the SCAM intolerant device with a value other than its default ID of 6. The SCAM intolerant device does not respond to soft ID assignment. As a result, the Am53C974A concludes that this device is a SCAM intolerant device with a hard ID.

(6) The Am53C974A initiates a SCSI bus reset.

(7) The Am53C974A assigns a soft ID to the SCAM slave device only.

Example 3

A SCSI system has two SCAM tolerant devices with IDs of 2 and 3, and one SCAM slave device with default ID of 3. The following algorithm can be applied.

(1) The host detects tolerant devices with IDs of 2 and 3 by using selection time out of 1 ms.

(2) The host initiates the SCAM selection sequences (discussed later) starting with the ID of 0 up to 6.

(3) SCAM slave devices respond to SCAM selection when the host selects device with ID of 3 (the slave device).

(4) The host assigns new ID to SCAM slave device.

8.7 ID ASSIGNMENT

8.7.1 Protocol Initialization

The procedure for assigning soft IDs can be divided into two major sequences: ID categorization and ID assignment. Once IDs have been categorized, the master may initiate the SCAM protocol to perform ID assignments. An example of the ID assignment sequence is shown in Figure 8-7. The SCAM protocol begins with the arbitration and SCAM protocol initialization phases. The following shows the procedure for a Level 1 master initiating the SCAM protocol.

(1) SCAM master arbitrates with its ID and wins.

(2) SCAM master asserts $\overline{\text{MSG}}$ and releases DATA bus after $\overline{\text{SEL}}$ is asserted.

(3) SCAM master releases $\overline{\text{BSY}}$ after at least 2 deskew delays.

(4) SCAM master releases $\overline{\text{MSG}}$ after at least a SCAM selection time.

(5) SCAM slave device asserts $\overline{\text{BSY}}$.

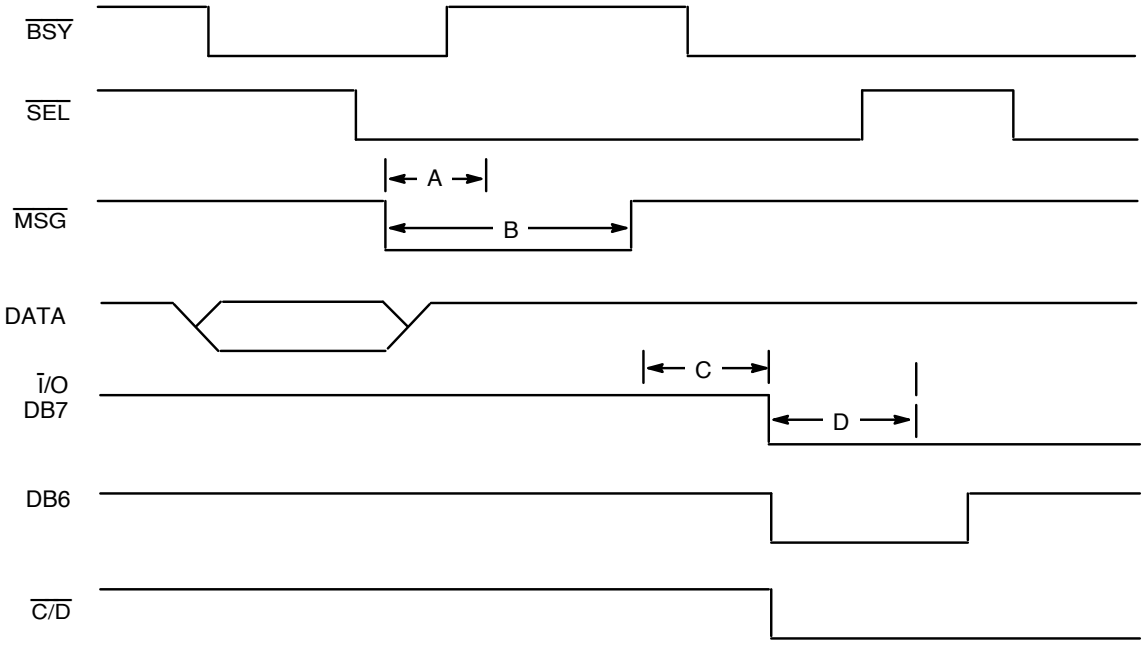
(6) SCAM slave device asserts $\overline{\text{I}}$/O, DB6, DB7 and SCAM master asserts $\overline{\text{I}}$/O $\overline{\text{C}}$/D, and DB7 after at least 2 deskew delays.

(7) SCAM master releases $\overline{\text{SEL}}$ after at least 2 deskew delays.

(8) All SCAM devices release DB6. If $\overline{\text{C}}$/D is not asserted, all SCAM devices will release all signals.

(9) Participating SCAM devices assert $\overline{\text{SEL}}$.

Figure 8-7 SCAM Protocol Initialization Sequence

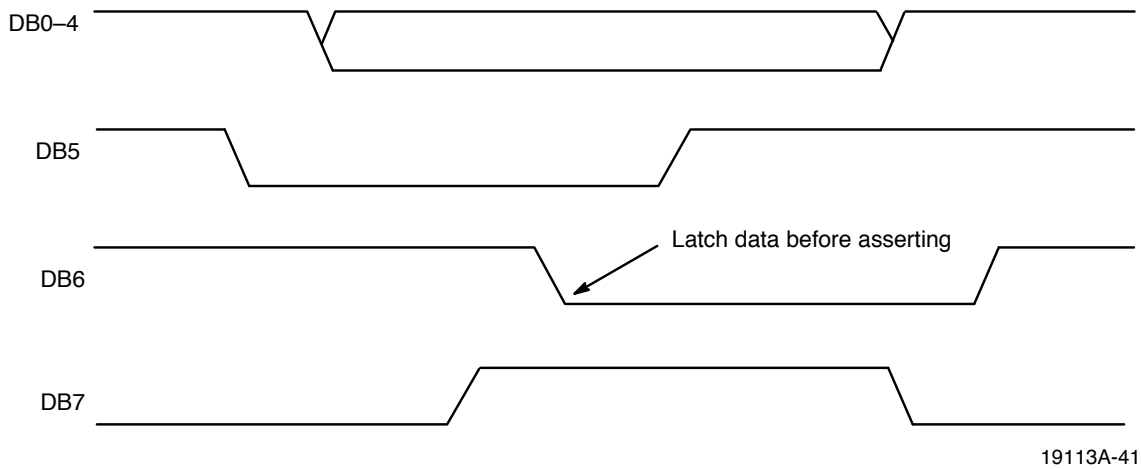


A = minimum 2 deskew delays
B = minimum a SCAM selection time
C = minimum 2 deskew delays
D = minimum 2 deskew delays

19113A-40

The SCAM protocol functions through a sequence of transfer cycles. During each cycle, certain devices (there may be more than two devices participating) send data to all participating SCAM devices. The actual data received is the logical OR of the data sent by all the sending devices. Figure 8-8 shows the transfer cycle operation.

Figure 8-8 Transfer Cycle Operation



19113A-41

SCAM device initiates the transfer cycle by putting the data on DB0-4, asserting DB5, and releasing (deasserting) DB7. The following steps shows the sequence of the transfer cycle.

(1) Place data on DB0-4, assert DB5, release (deassert) DB7.

(2) Wait until DB7 is released by all other devices.

(3) Read and latch data, and assert DB6.

(4) Release DB5.

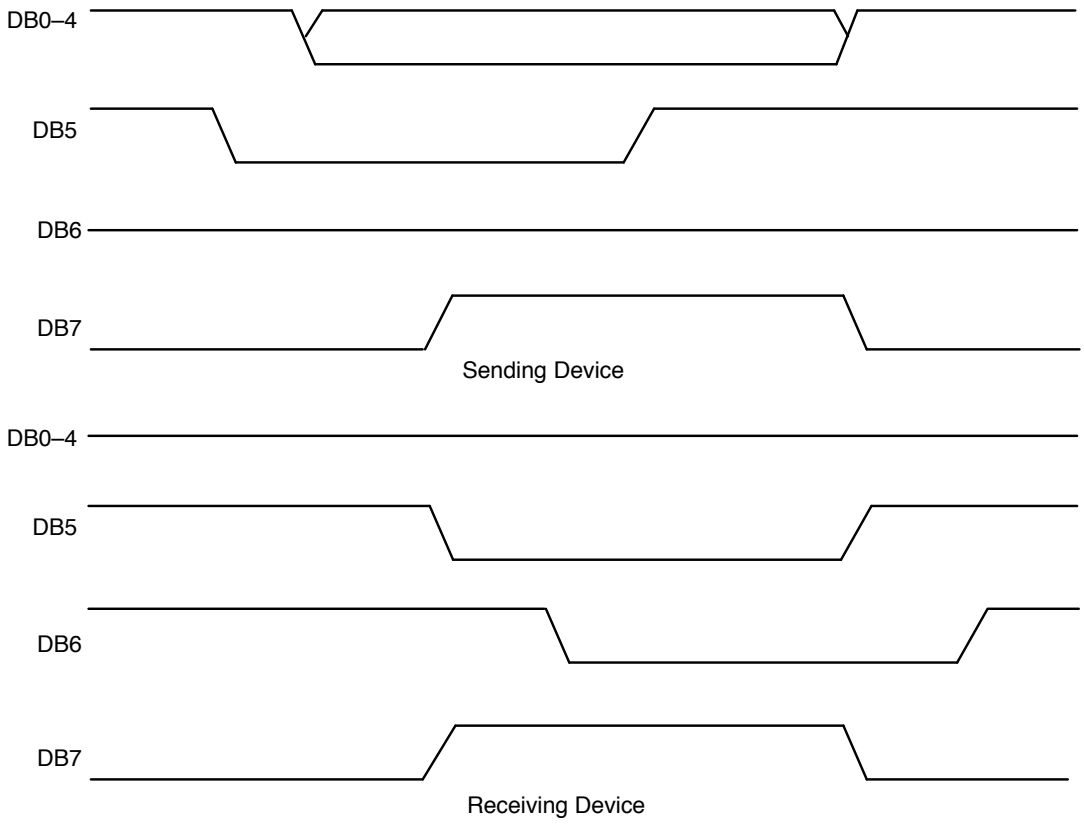
(5) Wait until DB5 is released by all other devices.

(6) Release or change DB0-4, assert DB7, and release DB6.

(7) Wait until DB6 is released by all other devices.

At the beginning and end of each cycle, DB7 is asserted while DB5 and DB6 are released. Figure 8-9 shows an example of a system with two SCAM devices in a transfer cycle. The sending device on a Level 1 SCAM capable bus will be the only SCAM master. After the SCAM master places the data on DB0-4, it asserts DB5 and deasserts DB7. When the SCAM slave device detects the assertion of DB5, it understands that a Transfer cycle is being initiated. At this point it will deassert DB7 and asserts DB5. During the time when DB5 or DB6 is asserted, it indicates that the transfer cycle is in progress. When the SCAM slave device finishes reading the data, it will assert DB6. When the sending device senses the assertion of DB6, it deasserts DB5 and prepares for termination of the cycle. The receiving device deasserts its DB5 to signal the completion of the read operation. The SCAM master will wait until DB5 is released by all devices before releasing DB0-4 and DB7. When the slave device detects the assertion of DB7, it will release DB6.

Figure 8-9 Example Showing Device Output Signal Timing Diagram During Transfer Cycle



8.7.2 Function Codes

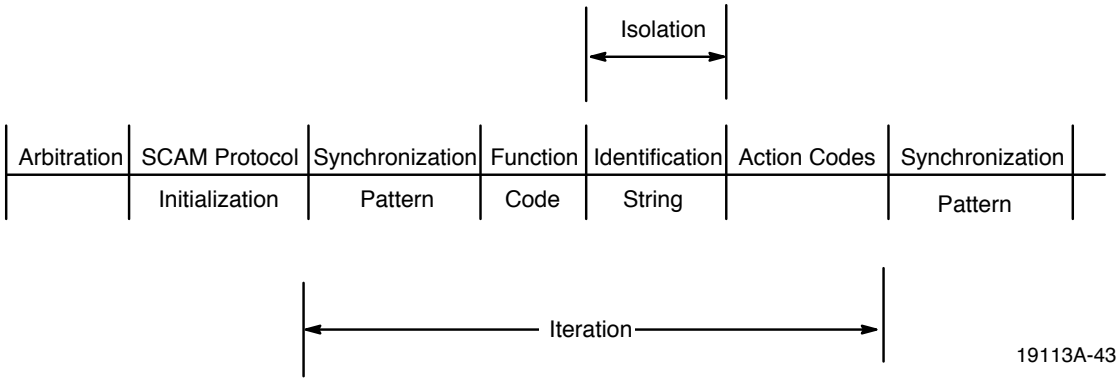
Successive transfer cycles are grouped into iterations. Each iteration performs a distinct functional purpose, such as assigning an ID to a single device. The first transfer cycle in each iteration transfers a synchronization pattern, which consists of all five data bits asserted. The master device asserts the synchronization pattern to begin a new iteration. This synchronization pattern may be asserted at any time to abort an iteration and begin a new one. The second transfer cycle in each iteration contains a function code. Table 8-1 defines the function codes.

Table 8-1 Table of Function Codes

Function Code	Description
00000b	Assign ID
00001b	Set Priority Flag
00010b to 01110b	reserved
01111b	Dominant Master Contention
10000b to 11110b	reserved
11111b	synchronization pattern

Slave devices shall ignore any iteration whose function codes are reserved or are codes they do not recognize. Figure 8-10 shows an example of a SCAM configuration sequence. The sequence begins with a SCAM protocol initialization sequence shown in Figure 8-6. The SCAM master will begin an iteration sequence by first sending a synchronization pattern to all SCAM slave devices. This will then be followed by a function code, identification string, and the tasks required for carrying out the function code.

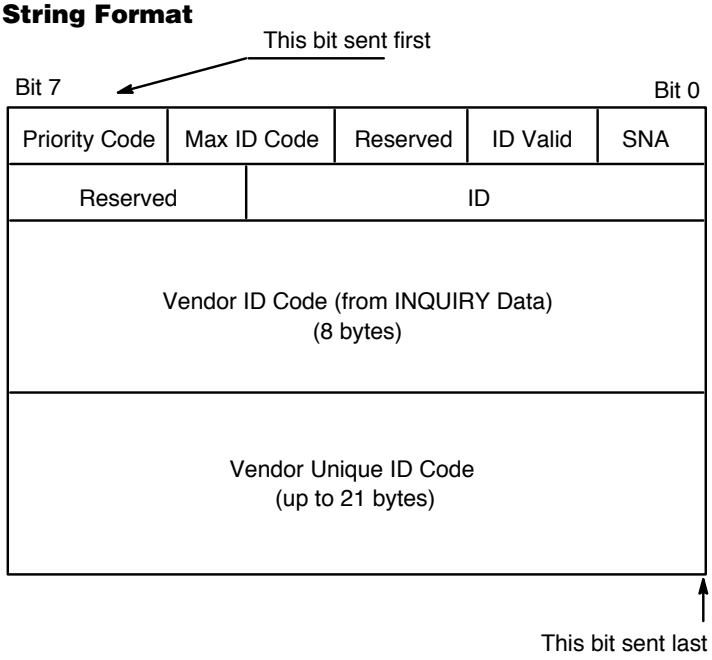
Figure 8-10 Example Showing a SCAM Configuration Sequence



8.7.3 Identification String

In a iteration sequence, each SCAM slave device will send out an identification string serially (will be explained later). This identification string (Figure 8-11) consist of a two byte type code: the 8-byte Vendor ID Code (from the INQUIRY data) followed by a vendor unique code which may consist of up to 21 bytes.

Figure 8-11 Identification String Format



The content of the header is explained below.

Priority Code

The 2 bit code which indicates the priority for a device winning an isolation stage. The values in this field depend upon the function code and are defined in the individual function code description.

Maximum ID Code

The 2 bits encode the maximum SCSI ID that the device can accept.

00b	device can accept up to 1Fh
01b	device can accept up to 0Fh
10b	device can accept up to 07h
11b	reserved

Reserved

This one bit code shall be sent as 0. A device that receives a 1 in this bit shall defer for the iteration.

ID Valid

The 2 bit code indicates the validity and content of the ID field as follows.

00b	ID not valid
01b	ID field contains a default ID or a soft ID
10b	ID field contains a hard ID.
11b	Reserved. A device that receives this code shall defer for the iteration.

SNA

A ‘1’ in this one bit field indicates that the device’s full identification string is available. If the device’s identification string is not yet available, and the device continues to participate in the isolation stage, the device shall stall some subsequent handshake until its identification information is available.

Reserved

The 3 bit code shall be sent as 0.

ID

The device’s default ID, assigned soft ID, or hard ID.

Vendor ID Code

The 8 byte code shows the Vendor ID Code obtained from the INQUIRY data.

Vendor Unique Code

This code may be up to 21 bytes in length. It contains the vendor unique code such as the device’s serial number.

During the isolation stage, each participating device sends an identification string bit serially. The bit values are encoded in bit 0 and 1 of DB0-4. For example, a ‘0’ will be encoded as 00001b while a ‘1’ will be encoded as 00010b on DB0-4. Each participating device will send and read the data in each transfer cycle. The data is interpreted in Table 8-2.

Table 8-2 Bit Pattern Interpretation During Identification String Transfer

Bit Value	Sent on DB4–0	Received on DB4–0	Condition
0	0001b	00001b	Continue
		00011b	Defer
1	00010b	0001Xb	Continue
none	00000b	000X1b	Defer
		0001Xb	Defer
		00000b	Terminate
any	000XXb	100XXb	Terminate
	other combination	other combination	Error

The ‘Continue’ condition means the device shall continue to participate in the isolation stage.

The ‘Defer’ condition means that the device has lost to a device with a higher identification string. The identification string of the winning device is obtained from DB1. For example, suppose that device A sends a bit string of 1010110... and device B sends 1010100....,

Device A	Device B	Bus Data
00010	00010	00010
00001	00001	00001
00010	00010	00010
00001	00001	00001
00010	00010	00010
00010	00001	00011 ←
00001	00001	00001

Since the sixth data bit of device A has a higher value, device B has to defer the identification string transfer. However, device B shall continue to handshake data without asserting DB0-4 and wait for the next synchronization pattern.

The 'Terminate' condition means that the isolation stage has terminated. The SCAM master may terminate the transfer by asserting DB4 (refer the above Terminate condition).

The 'Error' condition implies that a bus error or reserved pattern was encountered. The 'Error' condition is typically treated the same as the 'Defer' condition.

The isolation stage normally terminates with a single remaining participating device. When this occurs, a master may assign this remaining device an ID or instruct it to perform some other action.

8.7.4 Assigning IDs

Two function codes, the Assign ID and Set Priority Flag functions, can be used to assign soft IDs to SCAM slave devices. To assign IDs, the SCAM master should arbitrate, initiate the SCAM protocol, send the Assign ID or Set Priority Flag function code, enter the isolation stage, and assign IDs by sending Action Codes (will be discuss later in this section).

Each SCAM device maintains a Priority Flag while the SCAM protocol is active. The Priority Flag's value determines the content of the Priority Code field sent during the isolation stage for these functions. The Priority Code field will be sent as p0 (binary), where p is the current value of the device's Priority Flag. As a consequence, low priority devices (p=0) will defer to a high priority device (p=1).

Each SCAM device will set its Priority Flag to '1' (set by device's firmware) during SCAM selection. The Clear Priority Flag action of the Action Code (defined later in this section) sets the Priority Flag to '0'. The Assigned ID function leaves the flag unaltered while the Set Priority Flag function sets the flag to '1'.

After the isolation stage terminates, the master device will send an Action Code to the remaining device(s). When this occurs, two situations may result:

(1) One slave device responds to the Action Code ---- one slave device remains with high identification string.

(2) More than one slave device responds to Action Code --- SCAM master terminates isolation stage by asserting DB4. All slave devices will receive and act on Action Code.

Action Codes are two quintets sent on DB4–0. In each quintet, DB2–0 contains a three bit code value and DB4–3 contains two check bits. The value in DB4–3 is the count of the number of zero bits in DB2–0. Table 8-3 shows the possible Action Codes.

Table 8-3 Possible Action Codes

First Quintet	Second Quintet	Description
11000b	ccnnnb	Assign SCSI ID 00nnnb
10001b	ccnnnb	Assign SCSI ID 01nnnb
10010b	ccnnnb	Assign SCSI ID 10nnnb
01011b	ccnnnb	Assign SCSI ID 11nnnb
10100b	11000b	Clear Priority Bit
	10001b	reserved
	10010b	Locate On
	01011b	Locate Off
	others	reserved
01101b	ccnnnb	reserved
01110b	ccnnnb	reserved
00111b	ccnnnb	reserved

An Action Code is valid if the check bits are correct and both quintets are received. The remaining device(s) performs a valid Action Code as soon as they receive it. Transfer cycles after a valid Action Code and preceding the next synchronization pattern are ignored.

The Assign ID Action Code assigns ID to slave devices. The ‘cc’ of the second quintet is the parity code while the ‘nnn’ is the assigned ID.

The Clear Priority Bit Action Code instructs the remaining device(s) to clear their Priority Flag. This function is typically used when the master wishes to defer assigning an ID to any device(s) until a later iteration.

The Locate On and Off Action Code instructs the remaining device(s) to provide assistance for users or service personnel attempting to physically locate the device. Upon receiving a Locate On Action Code, the recommended action is for the remaining device(s) to flash their fault indicators or some similar indication. The indication should be cleared upon receiving a Locate Off Action Code, a reset condition, after a time delay, or upon other vendor unique actions.

A SCAM slave device that receives a valid ID assignment should release all bus signals and cease participating in the SCAM protocol until the next reset condition or power-on. SCAM slave devices shall continue participating in the SCAM protocol if they receive any other Action Code, receive an invalid or reserved Action Code, or do not receive an Action Code.

8.5.5 Default ID and ID Assignment

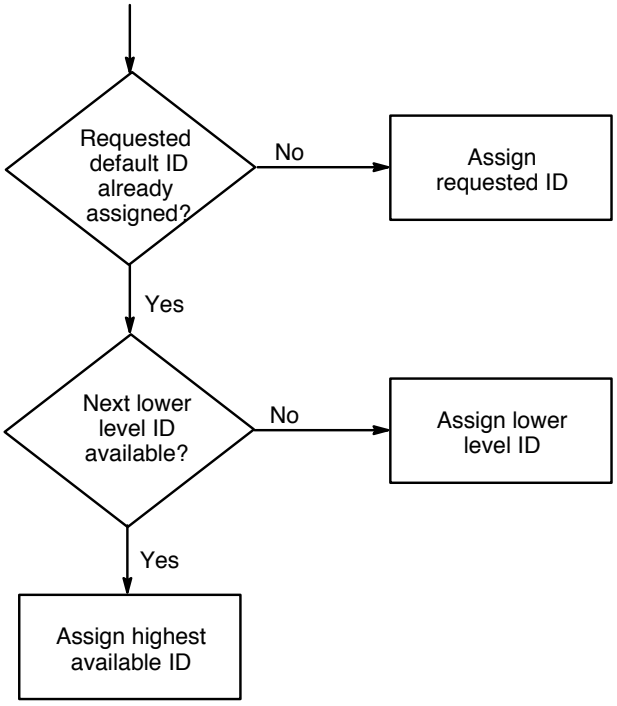
The default IDs of different kinds of SCAM devices are suggested by the SCSI committee and is specified in the *Proposal for Plug and Play SCSI Specification* Version 0.96, December 13, 1993. The default IDs listed in Table 8-4 are required settings for all Plug and Play SCSI devices as shipped from the factory.

Table 8-4 As Shipped SCSI ID Assignment

SCSI ID	SCSI Default ID
7	Host Adapter
6	Disk Drive
5	
4	Tape or R/W Optical
3	CD-ROM
2	Scanner/Printer
1	
0	

It is important to ensure consistent ID assignment to SCSI devices when the system is repeatedly powered-down or reset. Figure 8-12 shows the algorithm for maintaining consistent ID assignments.

Figure 8-13 Algorithm That Ensures Consistence ID Assignments



For example, a SCSI system has two SCAM tolerant hard drives of IDs 2 and 3, one SCAM tolerant CD-ROM with ID of 5, and a SCAM slave device with default ID of 3. During the ID assignment phase, since the default ID of 3 has already been assigned to a hard drive, the algorithm will check if the next lower level ID is available. Since an ID of 2 is also assigned to a hard drive, the algorithm will check for the highest available ID, which is the ID of 4. Since this ID is available, it will be assigned to the SCAM slave device.

CHAPTER

9 DESIGN CONSIDERATIONS FOR MOTHERBOARDS AND ADAPTER CARDS

9.1 INTRODUCTION

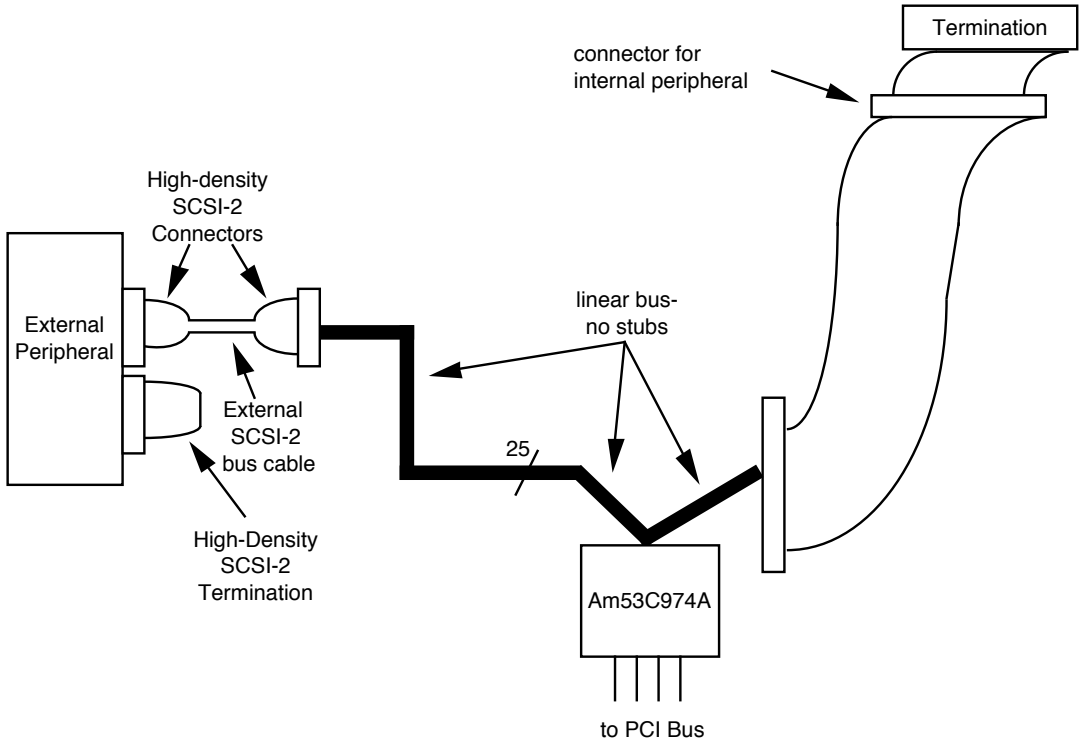
This chapter covers motherboard and add-in card design considerations which use the Am53C974A. It discusses SCSI component placement, signal routing, PCI interface recommendations, noise considerations and termination schemes.

9.2 SIGNAL ROUTING AND SCSI PLACEMENT

The main components for board design include the Am53C974A (whose maximum trace length from the PCI speedway or adapter card edge connector 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 Am53C974A.

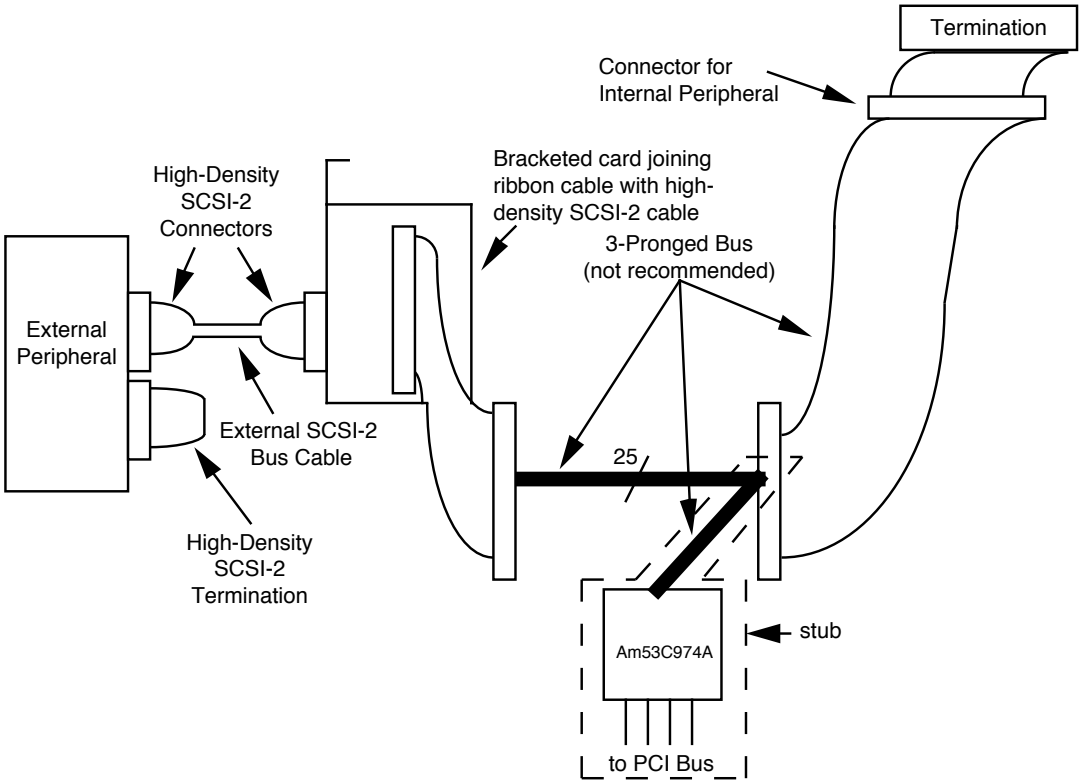
There are many ways to route SCSI bus traces on a host adapter board or motherboard. Ideally, traces from the internal SCSI bus connector, from the Am53C974A and from the external SCSI bus connector should all connect in series. Care should be taken not to have any stubs in the SCSI bus. (Stubs are any 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 Am53C974A. 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 9-1 and 9-2.

Figure 9-1 Ideal Routing Scheme



19113A-46

Figure 9-2 A Poor Routing Scheme



19113A-47

9.2.1 The Motherboard

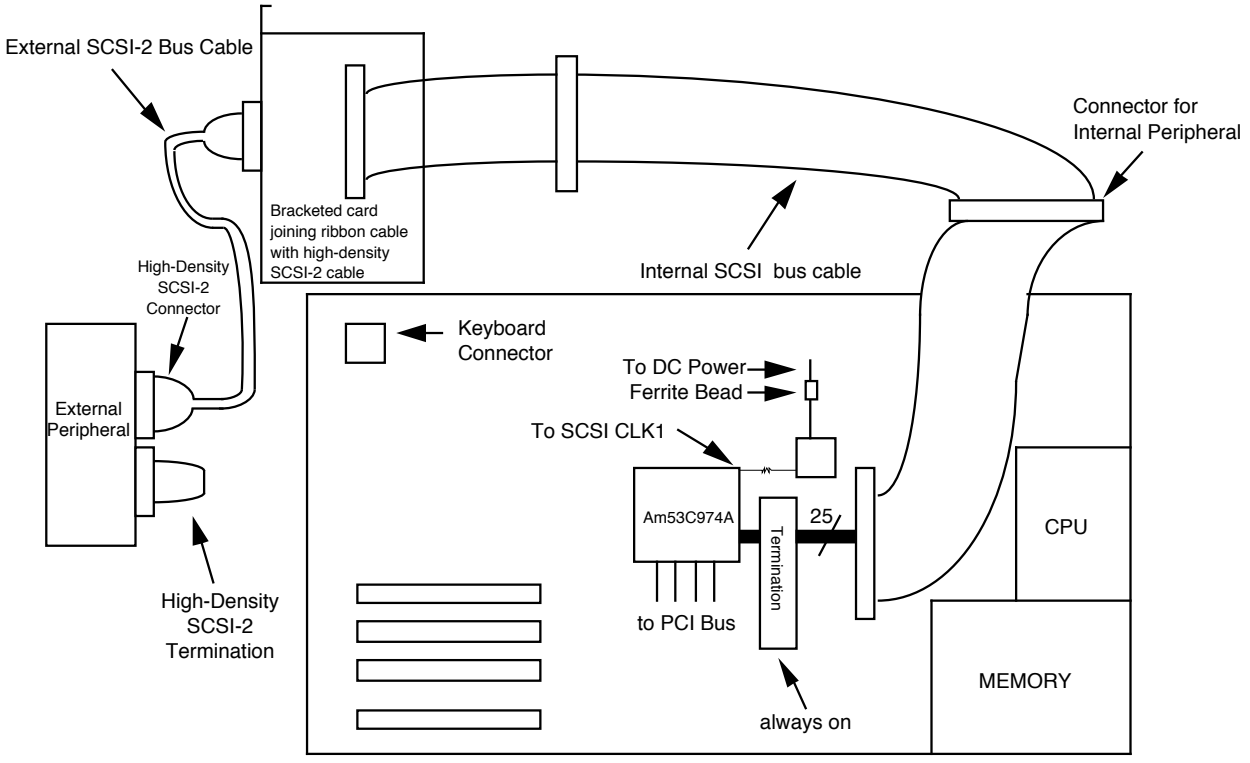
The following two layouts may be used as a guideline for the design of motherboards which incorporate the Am53C974A. For both layouts, the SCSI connectors should be placed as far away from the PCI Speedway as possible. Each layout refers to termination considerations which are described in further detail in section 9.4.

9.2.1.1 Layout #1

This approach avoids the cost of placing an external connector on the motherboard. In this configuration, the Am53C974A 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 9-3 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 Am53C974A.
- External terminators connected to the end of the external bus.

Figure 9-3 Motherboard Layout — Approach #1



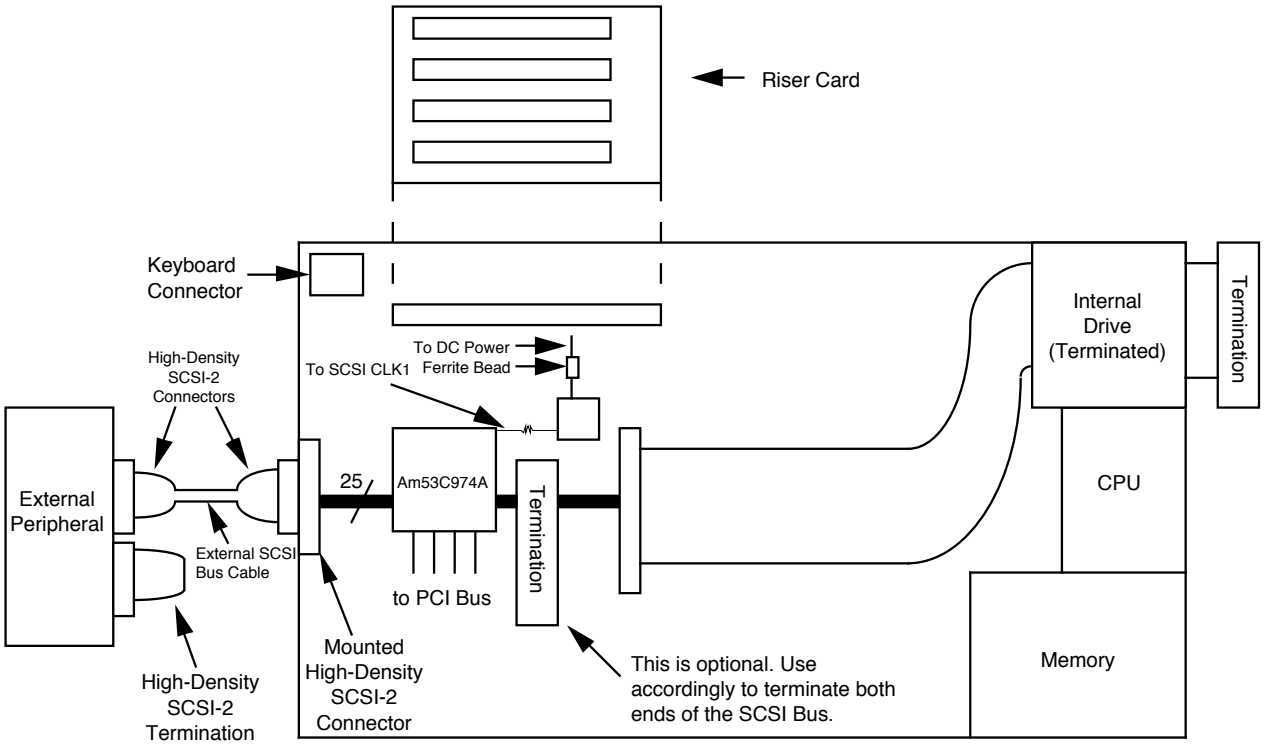
19113A-48

9.2.1.2 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 9-4:

- An external connector mounted on the motherboard with routings that connect to the Am53C974A.
- The Am53C974A 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 Am53C974A.

Figure 9-4 Motherboard Layout #2



19113A-49

Motherboard designs which place an internal and external connector on either side of the Am53C974A are discouraged since:

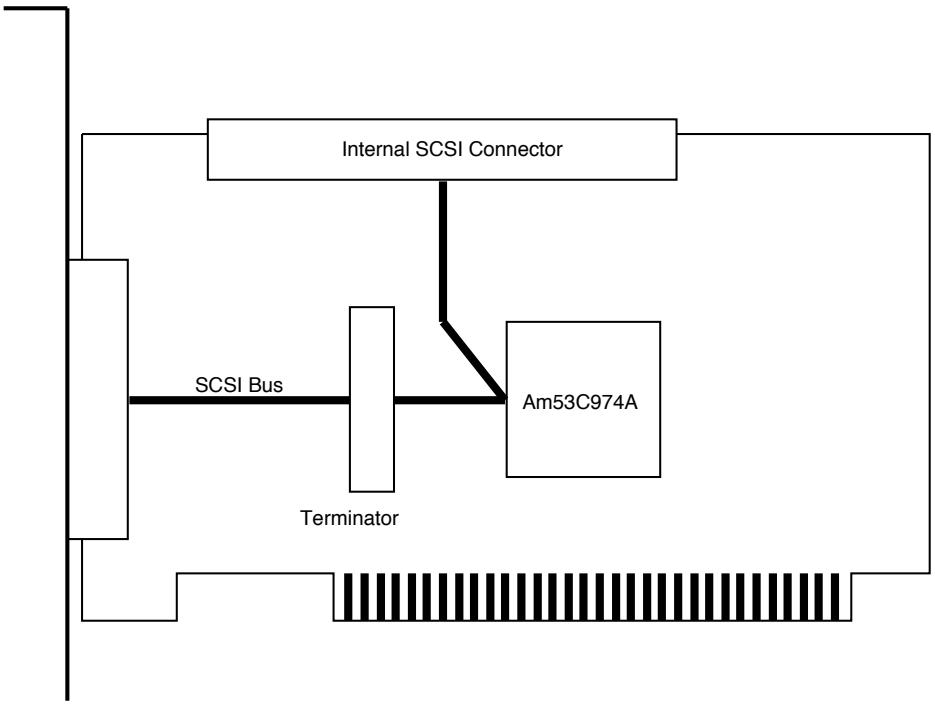
- Two SCSI connectors are required on the motherboard—one more than what the other two approaches call for.

- 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 Am53C974A are not routed correctly. See Figure 9-1 for the ideal routing scheme.

9.2.2 The Adapter Card

Generally, add-in SCSI adapter cards provide both an internal and external SCSI connector to allow for the attachment of SCSI peripherals. As a result, care should be taken in the layout of the board so that the ideal SCSI bus routing scheme is followed (Figure 9-1). Figure 9-5 shows a layout which preserves the ideal routing scheme and may be used as a guideline for the design of adapter cards which use the Am53C974A.

Figure 9-5 Layout Scheme for Adapter Card Designs



The following lists the recommendations for add-in SCSI card designs:

- A keyed 50-pin internal SCSI header for connecting an internal SCSI bus ribbon cable.
- A high-density 50-pin external SCSI connector for connecting an external SCSI-2 cable.
- On board regulated terminators a maximum of 0.1 m (3.937 in.) from the Am53C974A. This terminator can be designed to be automatically enabled or disabled depending on whether the adapter card is at one end of the SCSI chain or in the middle of the chain (refer to section 9.4.1.3).
- A 4-pin header for connecting the front panel disk drive activity LED.

9.3 NOISE CONSIDERATIONS

Some areas of a PCI motherboard or host adapter design (which include the Am53C974A) 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 Am53C974A.

9.3.1 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 pin on the Am53C974A, introduces the possibility of unwanted high frequency components, including harmonics, coupling with Am53C974A signals. To help prevent this, a 33 Ohm 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 pin. 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 Am53C974A.
- 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.

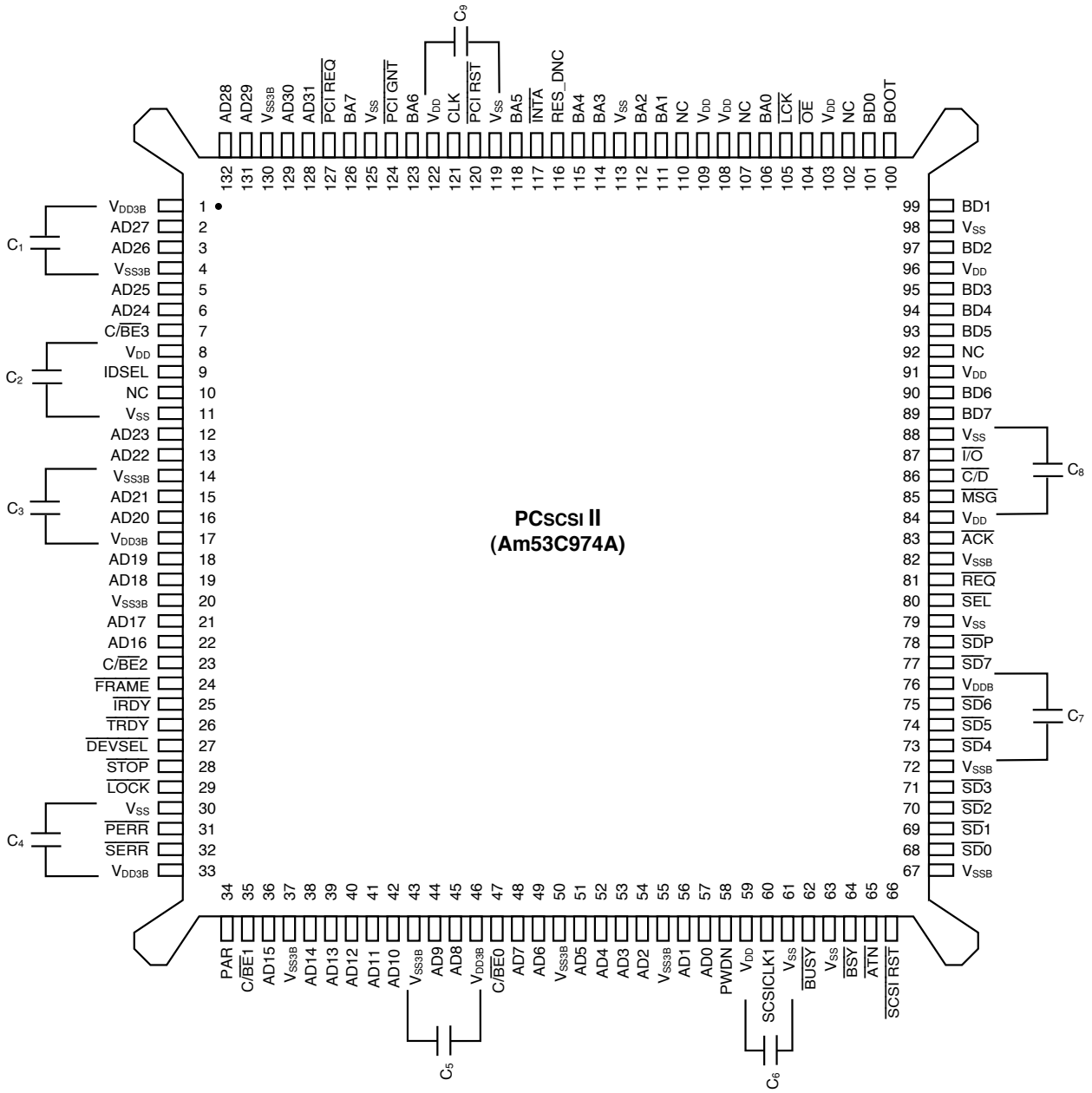
9.3.2 Decoupling Methods

Decoupling capacitors should be used across all V_{DD} and V_{SS} pins on motherboards and adapter cards. There are pairs of V_{DD} and V_{SS} pins on the Am53C974A that should each have their own decoupling capacitor (Figure 9-6). 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 Am53C974A 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 Am53C974A 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 feedthroughs to the V_{DD} and V_{SS} planes to reduce the inductance present.
- The trace length from pin to capacitor should be less than 0.25″, assuming a 20 mil trace.
- The average decoupling should be at least 0.01 μF per V_{DD} pin.

Additionally, all 3.3 V pins must also be decoupled with an average of at least 0.01 μF per gold finger.

Figure 9-6 Decoupling Capacitor Placement



19113A-50

where C1 – C9 are decoupling capacitors.

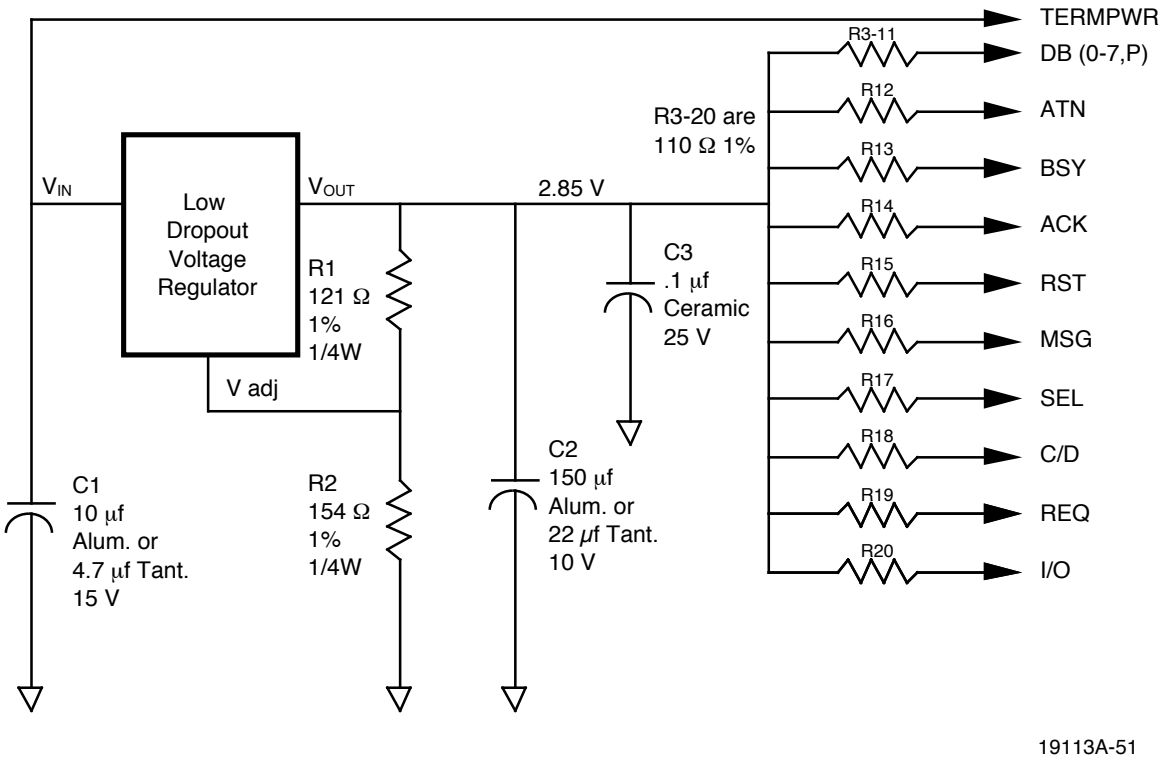
Decoupling methods which are NOT recommended include:

- 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.

9.4 TERMINATION CONSIDERATIONS

The use of active or regulated termination for terminators on motherboards and adapter cards is recommended (see Figure 9-7), 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 9-7 Regulated Termination



19113A-51

9.4.1 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 Am53C974A chip are connected by a linear SCSI bus. See Figure 9-1. In all schemes, it is recommended that active termination be used.

9.4.1.1 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.

9.4.1.2 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

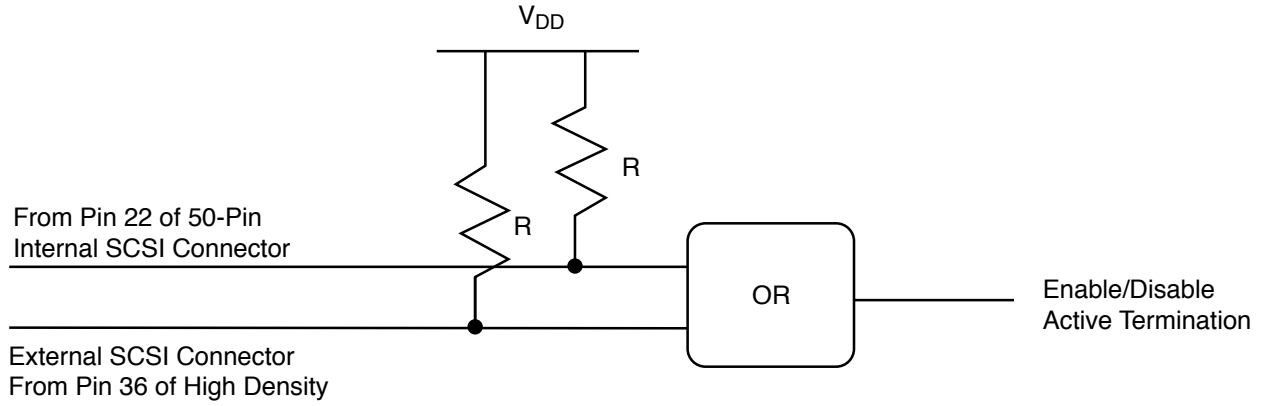
density SCSI-2 cable should connect the external port on the motherboard or host adapter to the first external peripheral. Peripherals may be added with more cables and the last peripheral should be terminated.

9.4.1.3 Scheme #3

In this case, both internal and external SCSI peripherals are used. The regulated terminators should be deactivated (since the Am53C974A will sit in the middle of the SCSI bus). This may be accomplished through hardware. This 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. Care should be taken to ensure that each end of the Bus is terminated.

Figure 9-8 shows the circuit to determine when to turn the on-board terminators on or off. In this figure, the inputs of the circuit are derived from one of the designated ground pins on each SCSI connector (internal and external). In this example, pin 22 of the Internal connector and pin 50 of the external connector are chosen for illustration purposes. When this is done, these ground pins are no longer connected to ground on the adapter card. Instead, they are pulled to V_{DD} through resistors R. As a result, when only internal or external peripherals are attached to the SCSI bus, one of the inputs to the OR gate will remain high so that the OR output will be high to enable the active terminator. However, when both internal and external peripherals are attached to the SCSI bus (the adapter card is in the middle of the SCSI chain), both OR inputs will be pulled low and the OR output will go low to disable the active terminator.

Figure 9-8 Circuit to Automatically Enable/Disable On-Board Active Termination



19113A-52

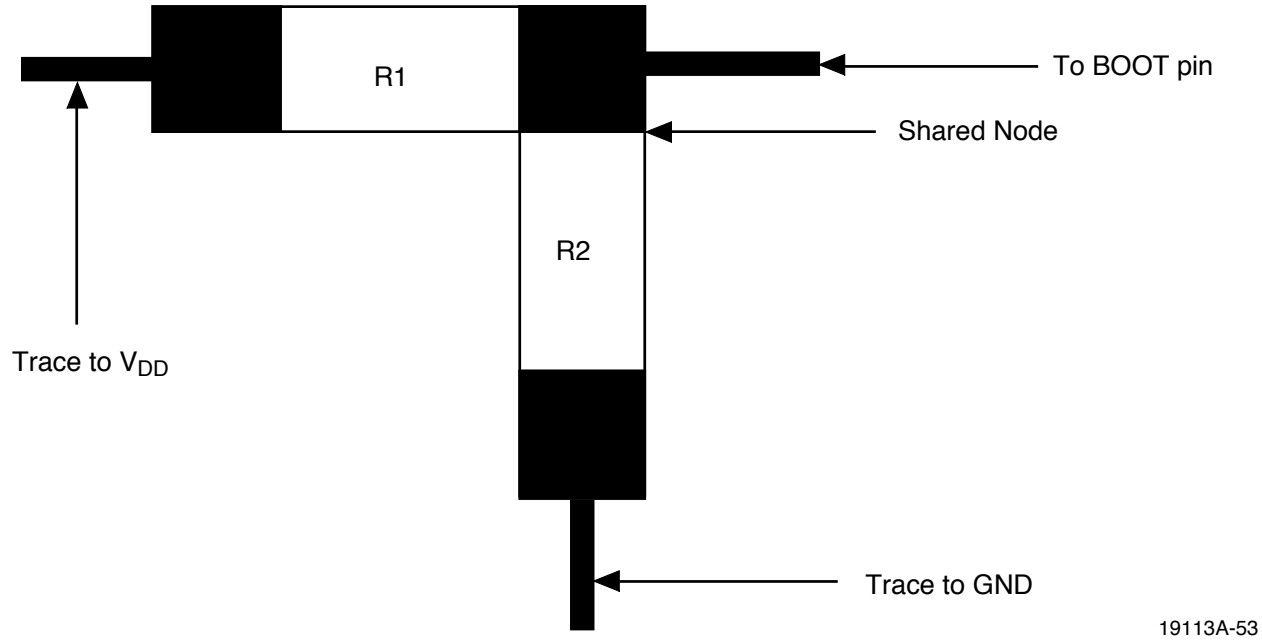
9.5 OTHER CONSIDERATIONS

The following are considerations for routing and layout for both motherboards and adapter cards. They should be taken into account along with SCSI considerations where applicable.

- 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.
- For a PCI CLK of 33 MHz, the maximum round trip time of any shared mother board signal should be less than 10 ns.
- The Am53C974A provides expansion ROM support which can be enabled/disabled via pin 100 (BOOT). For flexibility in building cards with or without Boot ROM capability, adapter cards may provide a pull-up or pull-down resistor stuffing option for this

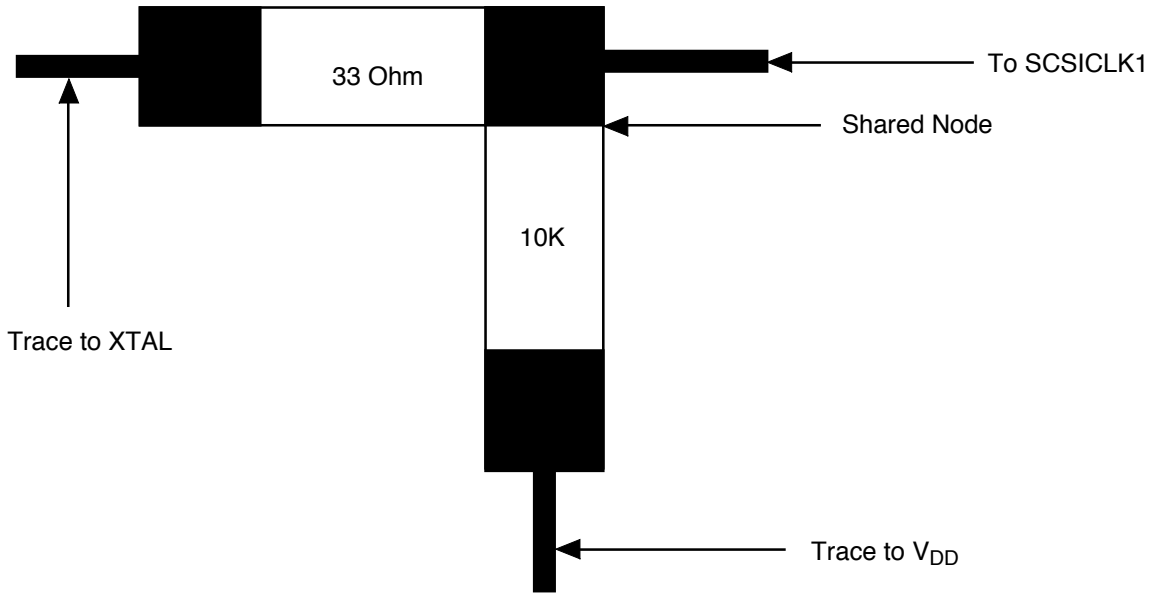
pin so that the Boot ROM can be enabled/disabled during the manufacturing of the board. In this case, the layout stuffing option can be made so that only one resistor can be stuffed. This can be done by laying the resistors perpendicular to each other while sharing one pad (Figure 9-9). Thus depending on which resistor is stuffed, the ROM can be enabled or disabled.

Figure 9-9 Layout of Stuffing Option for Boot Pin



- Similarly, since the Am53C974A can be operated with or without an external oscillator on the SCSICLK1 pin, a stuffing option can also be provided for this feature. In this case, if the oscillator is used, then only the oscillator and a 33 Ohm resistor are stuffed. However if the oscillator is not used, then only a 10K pull-up should be stuffed. As before, a perpendicular layout of the 33 and 10K Ohm resistors can be used to insure that only one resistor is stuffed.

Figure 9-10 Layout of Stuffing Option for SCSICLK1 Pin



19113A-54

- All unused inputs on the Am53C974A should be pulled to the inactive state with a 10K resistor unless specified otherwise in this manual.
- PRSNT[1:2] pins on adapter cards should reflect the maximum power consumption of the card. Typically these pins are grounded (7.5W maximum) for cards which use the Am53C974A. For motherboards, the PRSNT[2:1] pins should each be decoupled to ground with 0.01 μF capacitors.
- TDI and TDO pins on adapter cards should be connected together to provide JTAG continuity.

CHAPTER

10 AMD's PCscsi II SOFTWARE

10.1 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 II 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.

10.2 PCscsi II 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 II Software Architecture features:

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

10.3 OPERATING SYSTEM SUPPORT

AMD PCSCSI II 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.

10.3.1 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

10.3.2 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

10.3.3 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.

10.3.4 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

10.3.5 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.

10.3.6 SCO UNIX

a) Boot-Time Loadable Driver

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

10.3.7 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

10.4 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:

- Archive
- Connor
- 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 Am53C974A LITERATURE/TOOL SUPPORT

To enhance the ease of use for the Am53C974A, Bus Mastering SCSI-2 Controller for PCI systems, AMD has provided the following list of literature/tool support:

- Am53C974A Data Sheet – PID #19084A
- Am53C974A Technical Manual – PID #19113A
- AMD PC$_{SCSI}$ II Software Solutions Brochure – PID #18203A
- Embedding SCSI on PCI Motherboards Brochure – PID #18179A
- AMD's PCI Solutions Brochure – PID #18684A
- Am53C974A PC$_{SCSI}$ II Product Evaluation Kit containing
  - — Hardware reference platform
  - — SCSI cable
  - — SCSI Driver licensing information
  - — Evaluation software drivers for
    - — SCSI BIOS
    - — DOS
    - — Windows
    - — Windows NT
    - — NetWare
    - — SCO UNIX
    - — OS/2
  - — PCI Host Adapter Board User's Manual
  - — PC$_{SCSI}$ II Driver Installation Manual
  - — OS/Peripheral compatibility summary
  - — Am53C974A Data Sheet
  - — Am53C974A Technical Manual
  - — Performance benchmarking report
  - — PC$_{SCSI}$ II software 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 Am53C974A PC$_{SCSI}$ II 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