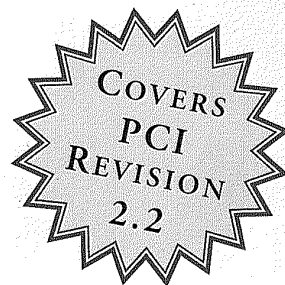


"The 'must-have' PC architecture reference set."

—PC Magazine's "Read Only" column

PCI SYSTEM ARCHITECTURE



FOURTH EDITION



MINDSHARE, INC

Tom Shanley / Don Anderson

**PC SYSTEM
ARCHITECTURE
S E R I E S**

PCI System Architecture, Fourth Edition

MINDSHARE, INC.

*Don Anderson
Tom Shanley*



ADDISON-WESLEY

An imprint of Addison Wesley Longman, Inc.

Reading, Massachusetts • Menlo Park, California • New York

Don Mills, Ontario • Harlow, England • Amsterdam

Bonn • Sydney • Singapore • Tokyo • Madrid • San Juan

Paris • Seoul • Milan • Mexico City • Taipei

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designators appear in this book, and Addison-Wesley was aware of the trademark claim, the designations have been printed in initial capital letters or all capital letters.

The authors and publisher have taken care in preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein. The publisher offers discounts on this book when ordered in quantity for special sales.

For more information, please contact:
Corporate, Government and Special Sales Group
Addison Wesley Longman, Inc.
One Jacob Way
Reading, Massachusetts 01867
(781) 944-3700

Library of Congress Cataloging-in-Publication Data is available.

ISBN: 0-201-30974-2

Copyright ©1999 by MindShare, Inc.
All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher.
Printed in the United States of America. Published simultaneously in Canada.

Sponsoring Editor: Karen Gettman
Production Coordinator: Jacquelyn Young
Cover Designer: Simone R. Payment
Set in 10 point Palatino by MindShare, Inc.

1 2 3 4 5 6 7 8 9-MA-0302010099
First Printing, May 1999

1

Intro To PCI

This Chapter

This chapter provides a brief history of PCI, introduces its major feature set, the concept of a PCI device versus a PCI function, and identifies the specifications that this book is based upon.

The Next Chapter

The next chapter provides an introduction to the PCI transfer mechanism, including a definition of the following basic concepts: burst transfers, the initiator, targets, agents, single and multi-function devices, the PCI bus clock, the address phase, claiming the transaction, the data phase, transaction completion and the return of the bus to the idle state. It defines how a device must respond if the device that it is transferring data with exhibits a protocol violation. Finally, it introduces the "green" nature of PCI—power conservation is stressed in the spec.

PCI Bus History

Intel made the decision not to back the VESA VL standard because the emerging standard did not take a sufficiently long-term approach towards the problems presented at that time and those to be faced in the coming five years. In addition, the VL bus had very limited support for burst transfers, thereby limiting the achievable throughput.

Intel defined the PCI bus to ensure that the marketplace would not become crowded with various permutations of local bus architectures peculiar to a specific processor bus. The first release of the specification, version 1.0, became available on 6/22/92. Revision 2.0 became available in April of 1993. Revision 2.1 was issued in Q1 of 1995. The latest version, 2.2, was completed on December 18, 1998, and became available in February of 1999.

Chapter 1: Intro To PCI

Specifications Book is Based On

This book is based on the documents indicated in Table 1-2 on page 13.

Table 1-2: This Book is Based On The Following Documents

Document Title	Revision
PCI Local Bus Specification	2.2
PCI-to-PCI Bridge Specification	1.1
PCI System Design Guide	1.0
PCI BIOS Specification	2.1
PCI Bus Power Management Interface Specification	1.1
PCI Hot-Plug Specification	1.0

Obtaining PCI Bus Specification(s)

The PCI bus specification, version 1.0, was developed by Intel Corporation. The specification is now managed by a consortium of industry partners known as the PCI Special Interest Group (SIG). MindShare, Inc. is a member of the SIG. The specifications are commercially available for purchase from the SIG. The latest revision of the specification (as of this printing) is 2.2. For information regarding the specifications and/or SIG membership, contact:

PCI Special Interest Group
2575 N.E. Kathryn #17
Hillsboro, Oregon 97124
1-800-433-5177 (USA)
503-693-6232 (International)
503-693-8344 (Fax)
pcisig@pcisig.com
<http://www.pcisig.com>

2 *Intro to PCI Bus Operation*

The Previous Chapter

The previous chapter provided a brief history of PCI, introduced its major feature set, the concept of a PCI device versus a PCI function, and identified the specifications that this book is based upon. It also provided information on contacting the PCI SIG.

In This Chapter

This chapter provides an introduction to the PCI transfer mechanism, including a definition of the following basic concepts: burst transfers, the initiator, targets, agents, single and multi-function devices, the PCI bus clock, the address phase, claiming the transaction, the data phase, transaction completion and the return of the bus to the idle state. It defines how a device must respond if the device that it is transferring data with exhibits a protocol violation. Finally, it introduces the "green" nature of PCI—power conservation is stressed in the spec.

The Next Chapter

Unlike many buses, the PCI bus does not incorporate termination resistors at the physical end of the bus to absorb voltage changes and prevent the wave-front caused by a voltage change from being reflected back down the bus. Rather, PCI uses reflections to advantage. The next chapter provides an introduction to reflected-wave switching.

Burst Transfer

Refer to Figure 2-1 on page 16. A burst transfer is one consisting of a single address phase followed by two or more data phases. The bus master only has to arbitrate for bus ownership one time. The start address and transaction type are issued during the address phase. All devices on the bus latch the address and

PCI System Architecture

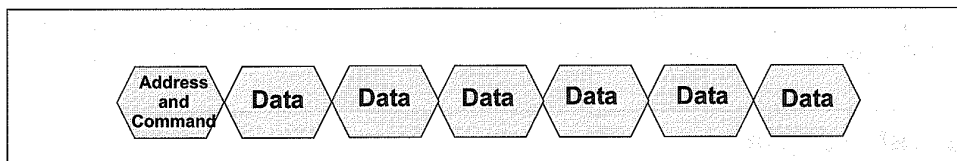
transaction type and decode them to determine which is the target device. The target device latches the start address into an address counter (assuming it supports burst mode—more on this later) and is responsible for incrementing the address from data phase to data phase.

PCI data transfers can be accomplished using burst transfers. Many PCI bus masters and target devices are designed to support burst mode. It should be noted that a PCI target may be designed such that it can only handle single data phase transactions. When a bus master attempts to perform a burst transaction, the target forces the master to terminate the transaction at the completion of the first data phase. The master must re-arbitrate for the bus to attempt resumption of the burst with the next data item. The target terminates each burst transfer attempt when the first data phase completes. This would yield very poor performance, but may be the correct approach for a device that doesn't require high throughput. Each burst transfer consists of the following basic components:

- The address and transfer type are output during the **address phase**.
- A data object (up to 32-bits in a 32-bit implementation or 64-bits in a 64-bit implementation) may then be transferred during each subsequent **data phase**.

Assuming that neither the initiator (i.e., the master) nor the target device inserts wait states in each data phase, a data object (a dword or a quadword) may be transferred on the rising-edge of each PCI clock cycle. At a PCI bus clock frequency of 33MHz, a transfer rate of 132Mbytes/second may be achieved. A transfer rate of 264Mbytes/second may be achieved in a 64-bit implementation when performing 64-bit transfers during each data phase. A 66MHz PCI bus implementation can achieve 264 or 528Mbytes/second transfer rates using 32- or 64-bit transfers. This chapter introduces the burst mechanism used to performing block transfers over the PCI bus.

Figure 2-1: Example Burst Data Transfer



Chapter 2: Intro to PCI Bus Operation

Initiator, Target and Agents

There are two participants in every PCI burst transfer: the initiator and the target. The initiator, or bus master, is the device that initiates a transfer. The terms bus master and initiator can be used interchangeably and frequently are in the PCI specification. The target is the device currently addressed by the initiator for the purpose of performing a data transfer. PCI initiator and target devices are commonly referred to as PCI-compliant agents in the spec.

Single- Vs. Multi-Function PCI Devices

A PCI physical device package may take the form of a component integrated onto the system board or may be implemented on a PCI add-in card. Each PCI package (referred to in the spec as a **device**) may incorporate from one to eight separate functions. A **function** is a logical device. This is analogous to a multi-function card found in any ISA, EISA or Micro Channel machine.

- A package containing one function is referred to as a **single-function PCI device**,
- while a package containing two or more PCI functions is referred to as a **multi-function PCI device**.

Each function contains its own, individually-addressable configuration space, 64 dwords in size. Its configuration registers are implemented in this space. Using these registers, the configuration software can automatically detect the function's presence, determine its resource requirements (memory space, IO space, interrupt line, etc.), and can then assign resources to the function that are guaranteed not to conflict with the resources assigned to other devices.

PCI Bus Clock

Refer to the CLK signal in Figure 2-2 on page 19. All actions on the PCI bus are synchronized to the PCI CLK signal. The frequency of the CLK signal may be anywhere from 0MHz to 33MHz. The revision 1.0 specification stated that all devices must support operation from 16 to 33MHz, while recommending support for operation down to 0MHz (in other words, when the clock has been stopped as a power conservation strategy). The revision 2.x (x = 1 or 2) PCI specification indicates that **all PCI devices must support PCI operation within the 0MHz to 33MHz range**. Support for operation down to 0MHz provides

PCI System Architecture

low-power and static debug capability. On a bus with a clock running at 33MHz or slower, the PCI CLK frequency may be changed at any time and may be stopped (but only in the low state). Components integrated onto the system board may be designed to only operate at a single frequency and may require a policy of no frequency change (and the system board designer would ensure that the clock frequency remains unchanged). Devices on add-in cards must support operation from 0 through 33MHz (because the card must operate in any platform that it may be installed in).

The revision 2.1 specification also defined PCI bus operation at speeds of up to 66MHz. The chapter entitled "66MHz PCI Implementation" describes the operational characteristics of the 66MHz PCI bus, embedded devices and add-in cards.

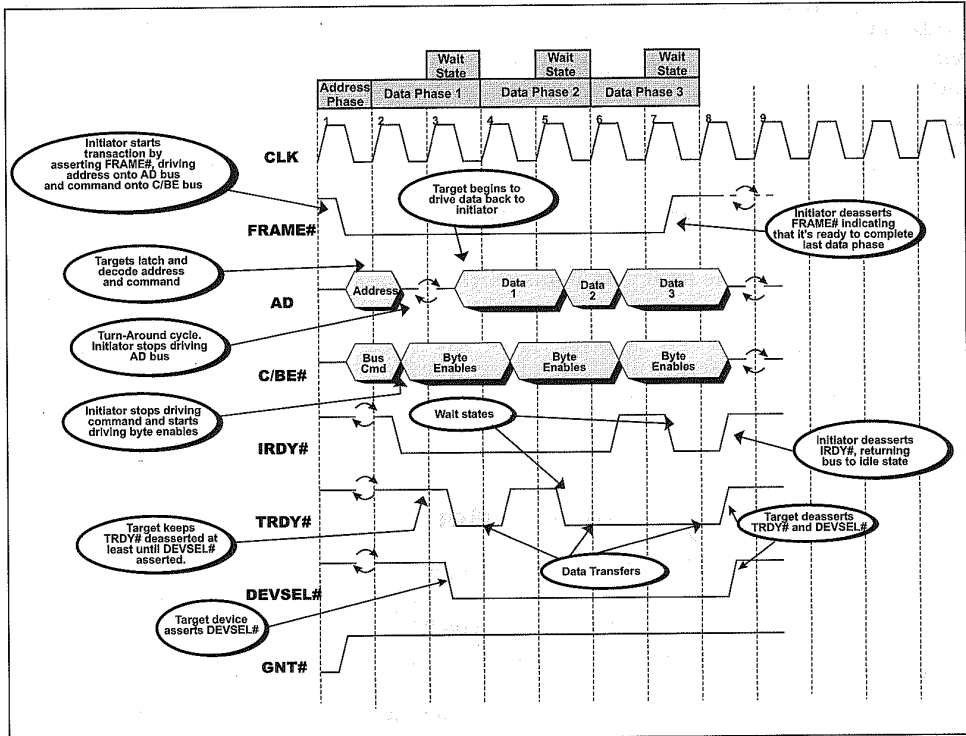
Address Phase

Refer to Figure 2-2 on page 19. Every PCI transaction (with the exception of a transaction using 64-bit addressing) starts off with an address phase one PCI clock period in duration (the only exception is a transaction wherein the initiator uses 64-bit addressing delivered in two address phases and consuming two PCI clock periods—this topic is covered in "64-bit Addressing" on page 287). During the address phase, the initiator identifies the target device (via the address) and the type of transaction (also referred to as the command type). The target device is identified by driving a start address within its assigned range onto the PCI address/data bus. At the same time, the initiator identifies the type of transaction by driving the command type onto the 4-bit wide PCI Command/Byte Enable bus. The initiator also asserts the FRAME# signal to indicate the presence of a valid start address and transaction type on the bus. Since the initiator only presents the start address and command for one PCI clock cycle, it is the responsibility of every PCI target device to latch the address and command on the next rising-edge of the clock so that it may subsequently be decoded.

By decoding the address latched from the address bus and the command type latched from the Command/Byte Enable bus, a target device can determine if it is being addressed and the type of transaction in progress. It's important to note that the initiator only supplies a start address to the target (during the address phase). Upon completion of the address phase, the address/data bus becomes the data bus for the duration of the transaction and is used to transfer data in each of the data phases. It is the responsibility of the target to latch the start address and to auto-increment it (assuming that the target supports burst transfers) to point to the next group of locations (a dword or a quadword) during each subsequent data transfer.

Chapter 2: Intro to PCI Bus Operation

Figure 2-2: Typical PCI Transaction



Claiming the Transaction

Refer to Figure 2-2 on page 19. When a PCI target determines that it is the target of a transaction, it must claim the transaction by asserting DEVSEL# (Device Select). If the initiator doesn't sample DEVSEL# asserted within a predetermined amount of time, it aborts the transaction.

Data Phase(s)

Refer to Figure 2-2 on page 19. The data phase of a transaction is the period during which a data object is transferred between the initiator and the target. The number of data bytes to be transferred during a data phase is determined by the number of Command/Byte Enable signals that are asserted by the initiator during the data phase.

PCI System Architecture

Each data phase is at least one PCI clock period in duration. Both the initiator and the target must indicate that they are ready to complete a data phase, or the data phase is extended by a wait state one PCI CLK period in duration. The PCI bus defines ready signal lines used by both the initiator (IRDY#) and the target (TRDY#) for this purpose.

Transaction Duration

The initiator doesn't issue a transfer count to the target. Rather, in each data phase it indicates whether it's ready to transfer the current data item and, if it is, whether it's the final data item. FRAME# is asserted at the start of the address phase and remains asserted until the initiator is ready (asserts IRDY#) to complete the final data phase. When the target samples IRDY# asserted and FRAME# deasserted in a data phase, it realizes that this is the final data phase. However, the data phase will not complete until the target has also asserted the TRDY# signal.

Transaction Completion and Return of Bus to Idle State

Refer to Figure 2-2 on page 19. The initiator indicates that the last data transfer (of a burst transfer) is in progress by deasserting FRAME# and asserting IRDY#. When the last data transfer has been completed, the initiator returns the PCI bus to the idle state by deasserting its ready line (IRDY#).

If another bus master had previously been granted ownership of the bus by the PCI bus arbiter and was waiting for the current initiator to surrender the bus, it can detect that the bus has returned to the idle state by detecting FRAME# and IRDY# both deasserted on the same rising-edge of the PCI clock (on the rising-edge of clock nine in the figure).

Response to Illegal Behavior

The PCI specification does not encourage the device designer to actively check for protocol violations. If a device does detect a violation, however, the following policy is advised.

Upon detection of illegal use of bus protocol, all PCI devices should be designed to gracefully return to the idle state (i.e., cease driving all bus signals and return its target or master state machine to the quiescent state) as quickly as possible. The good device (as opposed to the evil one) should attempt to stay operational so it can participate in future transactions.

Chapter 2: Intro to PCI Bus Operation

The specification is understandably vague on this point. It depends on the nature of the protocol violation as to whether the devices can gracefully return to their idle states and still function properly. As an example, the specification cites the case where the initiator simultaneously deasserts FRAME# and IRDY# (an illegal action). In this case, when the target detects this illegal end to the transaction, it is suggested that the target deassert all target-related signals and return its state machine to the idle state. In the event that a protocol violation leaves a target device questioning its ability to function correctly in the future, it can respond to all future access attempts with a Target Abort (explained in "Target Abort" on page 194). If the target thinks that the protocol violation has not impaired its ability to function correctly, it just surrenders all signals, returns to the idle state, and does not indicate any type of error.

"Green" Machine

In keeping with the goal of low power consumption, the specification calls for low-power, CMOS output drivers and receivers to be used by PCI devices.

The next chapter introduces the reflected-wave switching used in the PCI bus environment to permit low-power, CMOS drivers to successfully drive the bus.

If the address/data bus and command/byte enable signals are permitted to float (around the switching region of the CMOS input buffers which exist in every PCI device) for extended periods of time, the receiver inputs would oscillate and draw excessive current. This is a violation of the spec. To prevent this from happening, it is a rule in PCI that the address/data bus, the command/byte enables, and the parity signal must not be permitted to float for extended periods of time. Since the bus is normally driven most of the time, it may be assumed that the pre-charged bus (precharged by the last entity to drive the bus) will retain its state while not being driven for brief periods of time during turnaround cycles (turnaround cycles are described in the chapter entitled "Read Transfers" on page 123).

"Bus Parking" on page 65 describes the mechanism utilized to prevent the bus from floating when the bus is idle. The chapters entitled "Read Transfers" on page 123 and "Write Transfers" on page 135 describe the mechanism utilized during data phases with wait states. The chapter entitled "The 64-bit PCI Extension" describes the mechanism utilized to keep the upper half of the bus from floating when it's not in use (during a 32-bit transfer).