

SCIENCE & ENGINEERING LIBRARY

# *PCI System Architecture*

*Third Edition*

MINDSHARE, INC.

TOM SHANLEY  
AND  
DON ANDERSON

RECEIVED

JAN 29 1996

SEAVER SCIENCE



**Addison-Wesley Publishing Company**

Reading, Massachusetts • Menlo Park, California • New York  
Don Mills, Ontario • Wokingham, 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 designations appear in this book, and Addison-Wesley was aware of a trademark claim, the designations have been printed in initial capital letters or all capital letters.

The authors and publishers 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.

Library of Congress Cataloging-in-Publication Data

ISBN: 0-201-40993-3

Copyright © 1995 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: Keith Wollman  
Project Manager: Eleanor McCarthy  
Production Coordinator: Lora L. Ryan  
Cover design: Barbara T. Atkinson  
Set in 10 point Palatino by MindShare, Inc.

1 2 3 4 5 6 7 8 9 -MA- 9998979695

First printing, February 1995

Addison-Wesley books are available for bulk purchases by corporations, institutions, and other organizations. For more information please contact the Corporate, Government, and Special Sales Department at (800) 238-9682.

To Nancy and Sheryl, two very understanding ladies.

# Contents

Acknowledgments .....xxx

## About This Book

The MindShare Architecture Series ..... 1  
 Organization of This Book ..... 2  
 Who this Book is For ..... 2  
 Prerequisite Knowledge..... 3  
 Object Size Designations..... 3  
 Documentation Conventions..... 3  
     Hex Notation ..... 3  
     Binary Notation ..... 3  
     Decimal Notation ..... 4  
     Signal Name Representation ..... 4  
     Identification of Bit Fields (logical groups of bits or signals) ..... 4  
 We Want Your Feedback..... 4  
     Bulletin Board..... 5  
     Mailing Address..... 5

## Part I: Introduction to the Local Bus Concept

### CHAPTER 1: The Problem

Block-Oriented Devices ..... 9  
     Graphics Interface Performance Requirements ..... 9  
     SCSI Performance Requirements ..... 10  
     Network Adapter Performance Requirements..... 10  
     X-Bus Device Performance Constraints ..... 10  
 Expansion Bus Transfer Rate Limitations ..... 13  
     ISA Expansion Bus ..... 13  
     EISA Expansion Bus..... 13  
     Micro Channel Architecture Expansion Bus..... 13  
 Teleconferencing Performance Requirements ..... 14

### CHAPTER 2: Solutions, VESA and PCI

Graphics Accelerators: Before Local Bus ..... 19  
 Local Bus Concept..... 20  
     Direct-Connect Approach..... 20  
     Buffered Approach..... 22  
     Workstation Approach ..... 24



---

<b>VESA VL Bus Solution .....</b>	
Logic Cost .....	
Performance.....	
Longevity .....	
Teleconferencing Support.....	
Electrical Integrity .....	
Add-in Connectors.....	
Auto-Configuration.....	
Revision 2.0 VL Specification .....	
<b>PCI Bus Solution.....</b>	
<b>Market Niche for PCI and VESA VL.....</b>	
PCI Device .....	
Specifications Book is Based on .....	
Obtaining PCI Bus Specification(s) .....	

---

## Part II: Revision 2.1 Essentials

---

### CHAPTER 3: Intro to PCI Bus Operation

Burst Transfer.....	
Initiator, Target and Agents.....	
Single vs. Multi-Function PCI Devices.....	
PCI Bus Clock.....	
Address Phase .....	
Claiming the Transaction.....	
Data Phase(s) .....	
Transaction Duration.....	
Transaction Completion and Return of Bus to Idle State.....	
"Green" Machine .....	

---

### CHAPTER 4: Intro to Reflected-Wave Switching

Each Trace Is a Transmission Line .....	
Old Method: Incident-Wave Switching.....	
PCI Method: Reflected-Wave Switching .....	
PCI Timing Characteristics .....	
Introduction.....	
CLK Signal .....	
Output Timing.....	
Input Timing.....	
RST#/REQ64# Timing .....	
Slower Clock Permits Longer Bus .....	

---

**CHAPTER 5: The Functional Signal Groups**

Introduction ..... 53

System Signals ..... 56

    PCI Clock Signal (CLK)..... 56

    CLKRUN# Signal ..... 57

        General..... 57

    Reset Signal (RST#) ..... 58

Address/Data Bus..... 58

Preventing Excessive Current Drain..... 62

Transaction Control Signals ..... 63

Arbitration Signals ..... 64

Interrupt Request Signals ..... 65

Error Reporting Signals..... 65

    Data Parity Error..... 65

    System Error..... 66

Cache Support (Snoop Result) Signals ..... 67

64-bit Extension Signals ..... 68

Resource Locking..... 69

JTAG/Boundary Scan Signals ..... 70

Interrupt Request Lines ..... 71

Sideband Signals ..... 71

Signal Types ..... 71

Central Resource Functions ..... 72

Subtractive Decode..... 73

    Background..... 73

    Tuning Subtractive Decoder..... 74

Reading Timing Diagrams..... 75

**CHAPTER 6: PCI Bus Arbitration**

Arbiter ..... 77

Arbitration Algorithm ..... 79

Example Arbiter with Fairness..... 80

Master Wishes To Perform More Than One Transaction..... 82

Hidden Bus Arbitration ..... 82

Bus Parking..... 82

Request/Grant Timing..... 84

Example of Arbitration Between Two Masters ..... 85

Bus Access Latency ..... 89

    Master Latency Timer: Prevents Master From Monopolizing Bus ..... 91

        Location and Purpose of Master Latency Timer..... 91

# PCI System Architecture

---

How LT Works .....	91
Is Implementation of LT Register Mandatory? .....	92
Can LT Value Be Hardwired (read-only)? .....	92
How Does Configuration Software Determine Timeslice To Be Allocated To Master? .....	92
Treatment of Memory Write and Invalidate Command .....	92
Limit on Master's Latency .....	93
Preventing Target From Monopolizing Bus .....	93
General .....	93
Target Latency on First Data Phase .....	95
Options for Achieving Maximum 16 Clock Latency .....	95
Different Master Attempts Access To Device With Previously-Latched Request .....	97
Special Cycle Monitoring While Processing Request .....	97
Delayed Request and Delayed Completion .....	97
Handling Multiple Data Phases .....	97
Master or Target Abort Handling .....	97
Commands That Can Use Delayed Transactions .....	98
Delayed Read Prefetch .....	98
Request Queuing and Ordering Rules .....	98
Locking, Delayed Transactions and Posted Writes .....	103
<b>Fast Back-to-Back Transactions .....</b>	<b>103</b>
Decision to Implement Fast Back-to-Back Capability .....	106
Scenario One: Master Guarantees Lack of Contention .....	106
How Collision Avoided On Signals Driven By Master .....	106
How Collision Avoided On Signals Driven By Target .....	107
How Targets Recognize New Transaction Has Begun .....	108
Fast Back-to-Back and Master Abort .....	108
Scenario Two: Targets Guarantee Lack of Contention .....	110
State of REQ# and GNT# During RST# .....	111
Pullups On REQ# From Add-In Connectors .....	112
Broken Master .....	112

---

## CHAPTER 7: The Commands

Introduction .....	113
Interrupt Acknowledge Command .....	114
Introduction .....	114
Background .....	114
Host/PCI Bridge Handling of Interrupt Acknowledge Sequence .....	115
PCI Interrupt Acknowledge Transaction .....	116
Special Cycle Command .....	119

General.....	119
Special Cycle Generation .....	121
Special Cycle Transaction.....	121
Single-Data Phase Special Cycle Transaction.....	121
Multiple Data Phase Special Cycle Transaction.....	122
<b>I/O Read and Write Commands .....</b>	<b>124</b>
<b>Accessing Memory.....</b>	<b>124</b>
Reading Memory.....	125
Memory Read Command .....	125
Memory Read Line Command.....	125
Memory Read Multiple Command.....	125
Writing Memory.....	126
Memory Write Command .....	126
Memory Write and Invalidate Command.....	126
Problem .....	126
Description of Memory Write and Invalidate Command .....	127
More Information On Memory Transfers .....	127
<b>Configuration Read and Write Commands .....</b>	<b>128</b>
<b>Dual-Address Cycle.....</b>	<b>128</b>
<b>Reserved Bus Commands .....</b>	<b>128</b>

---

## **CHAPTER 8: The Read and Write Transfers**

<b>Some Basic Rules .....</b>	<b>129</b>
<b>Parity.....</b>	<b>130</b>
<b>Read Transaction.....</b>	<b>130</b>
Description.....	130
Treatment of Byte Enables During Read or Write.....	134
Byte Enable Settings May Vary from Data Phase to Data Phase.....	134
Data Phase with No Byte Enables Asserted .....	135
Target with Limited Byte Enable Support.....	136
Rule for Sampling of Byte Enables .....	136
Ignore Byte Enables During Line Read.....	136
Prefetching .....	137
Performance During Read Transactions .....	137
<b>Write Transaction.....</b>	<b>139</b>
Description.....	139
Performance During Write Transactions .....	144
Posted-Write Buffer .....	146
General .....	146
Combining.....	146
Byte Merging .....	147

# PCI System Architecture

---

Collapsing .....	147
Cache Line Merging .....	147
<b>Addressing Sequence During Memory Burst .....</b>	<b>148</b>
Linear and Cacheline Wrap Addressing .....	148
Target Response to Reserved Setting on AD[1:0] .....	150
<b>Do Not Merge Processor I/O Writes into Single Burst .....</b>	<b>150</b>
<b>PCI I/O Addressing .....</b>	<b>150</b>
General .....	150
Situation Resulting in Target-Abort .....	151
I/O Address Management .....	153
<b>When I/O Target Doesn't Support Multi-Data Phase Transactions .....</b>	<b>153</b>
<b>Address/Data Stepping .....</b>	<b>154</b>
Advantages: Diminished Current Drain and Crosstalk .....	154
Why Targets Don't Latch Address During Stepping Process .....	155
Data Stepping .....	155
How Device Indicates Ability to Use Stepping .....	155
Designer May Step Address, Data, PAR (and PAR64) and IDSEL .....	156
Continuous and Discrete Stepping .....	156
Disadvantages of Stepping .....	157
Preemption While Stepping in Progress .....	157
Broken Master .....	158
Stepping Example .....	159
When Not to Use Stepping .....	161
Who Must Support Stepping? .....	161
<b>Response to Illegal Behavior .....</b>	<b>161</b>

---

## CHAPTER 9: Premature Transaction Termination

<b>Introduction .....</b>	<b>163</b>
<b>Master-Initiated Termination .....</b>	<b>163</b>
Master Preempted .....	164
Preemption During Timeslice .....	164
Timeslice Expiration Followed by Preemption .....	165
Master Abort: Target Doesn't Claim Transaction .....	167
Introduction .....	167
Master Abort on Single Data Phase Transaction .....	167
Master Abort on Multi-Data Phase Transaction .....	169
Action Taken by Master in Response to Master Abort .....	171
General .....	171
Special Cycle and Configuration Access .....	171
<b>Target-Initiated Termination .....</b>	<b>171</b>
STOP# Signal .....	171

---

x



Disconnect.....	172
Description.....	172
Reasons Target Issues Disconnect.....	173
Target Slow to Complete Data Phase.....	173
Memory Target Doesn't Understand Addressing Sequence .....	173
Transfer Crosses Over Target's Address Boundary.....	173
Burst Memory Transfer Crosses Cache Line Boundary .....	174
Type "A" Disconnect: Initiator Not Ready When Target Says STOP .....	174
Type "B" Disconnect: Initiator Ready When Target Says STOP .....	175
Retry (Type C) Disconnect .....	178
Description.....	178
Reasons Target Issues Retry .....	179
Memory Target Doesn't Understand Addressing Sequence .....	179
Target Very Slow to Complete First Data Phase.....	179
Snoop Hit on Modified Cache Line.....	179
Resource Busy.....	180
Memory Target Locked.....	180
Retry Example .....	180
Host Bridge Retry Counter.....	182
Target Abort .....	182
Description.....	182
Reasons Target Issues Target Abort .....	183
Broken Target.....	183
I/O Addressing Error .....	183
Address Phase Parity Error .....	183
Master's Response to Target Abort.....	183
Target Abort Example.....	183
How Soon Does Initiator Attempt to Re-Establish Transfer After	
Retry or Disconnect? .....	185
Target-Initiated Termination Summary .....	185

---

## CHAPTER 10: Error Detection and Handling

Introduction to PCI Parity.....	187
PERR# Signal.....	189
Data Parity .....	189
Data Parity Generation and Checking on Read.....	189
Introduction.....	189
Example Burst Read.....	190
Data Parity Generation and Checking on Write.....	193
Introduction .....	193
Example Burst Write.....	193

# PCI System Architecture

---

Data Parity Reporting .....	196
General .....	196
Parity Error During Read .....	196
Parity Error During Write .....	197
Data Parity Error Recovery .....	198
Special Case: Data Parity Error During Special Cycle .....	199
Devices Excluded from PERR# Requirement .....	199
Chipsets .....	200
Devices That Don't Deal with OS/Application Program or Data .....	200
<b>SERR# Signal.....</b>	<b>201</b>
<b>Address Parity .....</b>	<b>202</b>
Address Parity Generation and Checking.....	202
Address Parity Error Reporting.....	202
<b>System Errors.....</b>	<b>205</b>
General.....	205
Address Phase Parity Error.....	205
Data Parity Error During Special Cycle .....	205
Target Abort Detection.....	205
Other Possible Causes of System Error .....	205
Devices Excluded from SERR# Requirement .....	205

---

## CHAPTER 11: Interrupt-Related Issues

Single-Function PCI Device .....	207
Multi-Function PCI Device.....	209
Connection of INTx# Lines To System Board Traces.....	209
<b>Interrupt Routing.....</b>	<b>210</b>
General .....	210
Platform "Knows" Interrupt Trace Layout .....	216
Well-Designed Platform Has Programmable Interrupt Router.....	216
Interrupt Routing Information.....	216
<b>PCI Interrupts Are Shareable .....</b>	<b>217</b>
<b>"Hooking" the Interrupt .....</b>	<b>217</b>
<b>Interrupt Chaining.....</b>	<b>218</b>
General .....	218
Step One: Initialize All Entries In Table To Null Value .....	219
Step Two: Initialize All Entries For Embedded Devices .....	219
Step Three: Hook Entries For Embedded Device BIOS Routines .....	219
Step Four: Perform PCI Device Scan .....	220
Step Five: Perform Expansion Bus ROM Scan.....	221
Step Six: Load Operating System.....	221
<b>A Linked-List Has Been Built for Each Interrupt Level .....</b>	<b>222</b>



## Contents

---

<b>Servicing Shared Interrupts</b> .....	223
Example Scenario .....	223
Both Devices Simultaneously Generate Requests .....	224
Processor Interrupted and Requests Vector .....	225
First Handler Executed.....	226
Return to Interrupted Program, Followed by Second Interrupt .....	227
First Handler Executed Again, Passes Control to Second .....	227
<b>Implied Priority Scheme</b> .....	227
<b>Interrupts and PCI-to-PCI Bridges</b> .....	228

---

### CHAPTER 12: Shared Resource Acquisition

<b>Using Semaphore to Gain Exclusive Ownership of Resource</b> .....	229
Memory Semaphore Definition.....	229
Synchronization Problem .....	230
<b>PCI Solutions: Bus and Resource Locking</b> .....	231
LOCK# Signal .....	231
Bus Lock: Permissible but Not Preferred .....	231
Resource Lock: Preferred Solution .....	232
Introduction .....	232
Determining Lock Mechanism Availability .....	233
Establishing Lock on Memory Target.....	233
Unlocked Targets May Be Accessed by any Master.....	237
Access to Locked Target by Master Other than Owner: Retry.....	237
Continuation and/or End of Locked Transaction Series .....	239
<b>Potential Deadlock Condition</b> .....	241
Assumptions.....	241
Problem Scenario.....	242
Solution .....	243
<b>Devices that Must Implement Lock Support</b> .....	244
<b>Use of LOCK# with 64-bit Addressing</b> .....	244
<b>Summary of Locking Rules</b> .....	244
Implementation Rules for Masters.....	244
Implementation Rules for Targets .....	245

---

### CHAPTER 13: The 64-bit PCI Extension

<b>64-bit Data Transfers and 64-bit Addressing: Separate Capabilities</b> .....	247
<b>64-bit Cards in 32-bit Add-in Connectors</b> .....	248
<b>Pullups Prevent 64-bit Extension from Floating When Not in Use</b> .....	249
Problem: 64-bit Cards Inserted in 32-bit PCI Connectors .....	250
How 64-bit Card Determines Type of Slot Installed In.....	250
<b>64-bit Data Transfer Capability</b> .....	252

---

## PCI System Architecture

---

Only Memory Commands May Use 64-bit Transfers .....	253
Start Address Quadword-Aligned .....	253
64-bit Target's Interpretation of Address .....	253
32-bit Target's Interpretation of Address .....	254
64-bit Initiator and 64-bit Target .....	254
64-bit Initiator and 32-bit Target .....	257
Null Data Phase Example .....	260
32-bit Initiator and 64-bit Target .....	262
Performing One 64-bit Transfer .....	262
With 64-bit Target .....	263
With 32-bit Target .....	263
Simpler and Just as Fast: Use 32-bit Transfers .....	264
With Known 64-bit Target .....	264
Disconnect on Initial Data Phase .....	267
<b>64-bit Addressing</b> .....	<b>267</b>
Used to Address Memory Above 4GB .....	267
Introduction .....	268
64-bit Addressing Protocol .....	268
64-bit Addressing by 32-bit Initiator .....	268
64-bit Addressing by 64-bit Initiator .....	271
32-bit Initiator Addressing Above 4GB .....	274
Subtractive Decode Timing Affected .....	274
Master Abort Timing Affected .....	275
Address Stepping .....	275
FRAME# Timing in Single Data Phase Transaction .....	276
<b>64-bit Parity</b> .....	<b>276</b>
Address Phase Parity .....	276
PAR64 Not Used for Single Address Phase .....	276
PAR64 Not Used for Dual-Address Phases by 32-bit Master .....	276
PAR64 Used for Dual-Address Phase by 64-bit Master (when requesting 64-bit data transfers) .....	276
Data Phase Parity .....	277

---

## CHAPTER 14: Add-in Cards and Connectors

<b>Expansion Connectors</b> .....	<b>279</b>
32 and 64-bit Connectors .....	279
32-bit Connector .....	283
Card Present Signals .....	283
REQ64# and ACK64# .....	284
64-bit Connector .....	284
3.3V and 5V Connectors .....	285

Shared Slot.....	286
Riser Card.....	288
Snoop Result Signals on Add-in Connector.....	288
<b>Expansion Cards.....</b>	<b>289</b>
3.3V, 5V and Universal Cards.....	289
Long and Short Form Cards.....	289
Component Layout.....	289
Maintain Integrity of Boundary Scan Chain.....	291
Card Power Requirement.....	291
Maximum Card Trace Lengths.....	292
One Load per Shared Signal.....	292

---

## Part III: Device Configuration In System With a Single PCI Bus

---

### CHAPTER 15: Intro to Configuration Address Space

Introduction.....	295
PCI Package vs. PCI Function.....	296
Three Address Spaces: I/O, Memory and Configuration.....	297
System with Single PCI Bus.....	299

### CHAPTER 16: Configuration Transactions

Which "Type" Are We Talking About?.....	301
Who Performs Configuration?.....	302
Bus Hierarchy.....	302
Intro to Configuration Mechanisms.....	304
Configuration Mechanism One.....	305
Background.....	305
Configuration Mechanism One Description.....	307
General.....	307
Configuration Address Port.....	307
Bus Compare and Data Port Usage.....	308
Multiple Host/PCI Bridges.....	309
Single Host/PCI Bridge.....	311
Generation of Special Cycles.....	311
Configuration Mechanism Two.....	312
Basic Configuration Mechanism.....	312
Configuration Space Enable, or CSE, Register.....	314
Forward Register.....	315
Support for Peer Bridges on Host Bus.....	315

---

## PCI System Architecture

---

Generation of Special Cycles .....	315
PowerPC PReP Memory-Mapped Configuration .....	316
Type Zero Configuration Transaction.....	319
Address Phase.....	319
Implementation of IDSEL.....	320
Data Phase .....	321
Type Zero Configuration Transaction Examples.....	321
Target Device Doesn't Exist.....	325
Configuration Burst Transactions.....	325
64-Bit Configuration Transactions Not Permitted .....	325
Resistively-Coupled IDSEL is Slow.....	326

---

### CHAPTER 17: Configuration Registers

Intro to Configuration Header Region .....	327
Mandatory Header Registers .....	329
Introduction.....	329
Vendor ID Register .....	329
Device ID Register.....	329
Command Register .....	329
General .....	329
VGA Color Palette Snooping .....	332
Status Register .....	333
Revision ID Register.....	335
Class Code Register .....	335
Header Type Register .....	340
Optional Header Registers.....	341
Introduction.....	341
Cache Line Size Register.....	341
Latency Timer: "Timeslice" Register.....	342
BIST Register .....	343
Base Address Registers.....	344
Memory-Mapping Recommended.....	345
Memory Base Address Register .....	345
I/O Base Address Register.....	347
Determining Block Size and Assigning Address Range .....	347
Expansion ROM Base Address Register .....	349
CardBus CIS Pointer .....	351
Subsystem Vendor ID and Subsystem ID Registers.....	352
Interrupt Pin Register .....	352
Interrupt Line Register.....	352
Min_Gnt Register: Timeslice Request.....	353

Max_Lat Register: Priority-Level Request .....	353
Add-In Memory.....	354
User-Definable Features .....	354

---

## CHAPTER 18: Expansion ROMs

ROM Purpose .....	357
ROM Detection .....	358
ROM Shadowing Required .....	361
ROM Content .....	361
Multiple Code Images.....	361
Format of a Code Image .....	363
General .....	363
ROM Header Format .....	365
ROM Signature .....	365
Processor/Architecture Unique Data .....	365
Pointer to ROM Data Structure.....	366
ROM Data Structure Format .....	366
ROM Signature .....	367
Vendor ID.....	367
Device ID.....	368
Pointer to Vital Product Data .....	368
PCI Data Structure Length.....	368
PCI Data Structure Revision.....	368
Class Code.....	368
Image Length .....	368
Revision Level of Code/Data .....	369
Code Type.....	369
Indicator Byte.....	369
Execution of Initialization Code .....	369
Introduction to Open Firmware .....	371

---

## Part IV: PCI-to-PCI Bridge

---

### CHAPTER 19: PCI-to-PCI Bridge

Scaleable Bus Architecture .....	375
Terminology.....	376
Example Systems.....	377
Example One .....	377
Example Two.....	380
PCI-to-PCI Bridge: Traffic Director .....	382
Configuration Registers .....	387



# PCI System Architecture

---

General.....	387
Header Type Register.....	389
Registers Related to Device ID.....	389
Introduction.....	389
Vendor ID Register.....	389
Device ID Register.....	390
Revision ID Register.....	390
Class Code Register.....	390
Bus Number Registers.....	391
Introduction.....	391
Primary Bus Number Register.....	391
Secondary Bus Number Register.....	391
Subordinate Bus Number Register.....	391
Address Decode-Related Registers.....	392
Basic Transaction Filtering Mechanism.....	392
Bridge Support for Internal Registers and ROM.....	393
Introduction.....	393
Base Address Registers.....	393
Expansion ROM Base Address Register.....	394
Bridge's I/O Filter.....	394
Introduction.....	394
Bridge Doesn't Support Any I/O Space Behind Bridge.....	395
Bridge Supports 64KB I/O Space Behind Bridge.....	395
Effect of ISA Mode Bit.....	400
Bridge Supports 4GB I/O Space Behind Bridge.....	404
Bridge's Memory Filter.....	406
Introduction.....	406
Configuration Software Detection of Prefetchable Memory Target.....	407
Bridge Supports 4GB Prefetchable Memory Space Behind Bridge.....	407
Bridge Supports 2 <sup>64</sup> Prefetchable Memory Space Behind Bridge.....	411
Rules and Options for Prefetchable Memory.....	411
Bridge's Memory-Mapped I/O Filter.....	413
Command Registers.....	414
Introduction.....	414
Command Register.....	415
Bridge Control Register.....	417
Status Registers.....	419
Introduction.....	419
Status Register (Primary Bus).....	419
Secondary Status Register.....	419
Cache Line Size Register.....	419

Latency Timer Registers .....	420
Introduction .....	420
Latency Timer Register (Primary Bus) .....	420
Secondary Latency Timer Register .....	420
BIST Register .....	420
Interrupt-Related Registers .....	420
<b>Configuration and Special Cycle Filter</b> .....	420
Introduction .....	420
Special Cycle Transactions .....	422
Type One Configuration Transactions .....	422
Type Zero Configuration Access .....	422
<b>Interrupt Acknowledge Handling</b> .....	422
<b>Configuration Process</b> .....	422
Introduction .....	422
Bus Number Assignment .....	430
Address Space Allocation .....	430
IRQ Assignment .....	432
Display Configuration .....	432
<b>Reset</b> .....	432
<b>Arbitration</b> .....	432
<b>Interrupt Support</b> .....	432
<b>Buffer Management</b> .....	432
Ensuring Reads Return Correct Data .....	432
Posted Memory Write Buffer .....	432
Handling of Memory Write and Invalidate Command .....	440
Creating Burst Write from Separate Posted Writes .....	440
Merging Separate Memory Writes into Single Data Phase Write .....	440
Collapsing Writes: Don't Do It .....	442
Bridge Support for Cacheable Memory on Secondary Bus .....	442
Multiple-Data Phase Special Cycle Requests .....	442
<b>Potential Deadlock Condition</b> .....	442
<b>Error Detection and Handling</b> .....	442
General .....	442
Handling Address Phase Parity Errors .....	442
Address Phase Parity Error on Primary Side .....	442
Address Phase Parity Error on Secondary Side .....	442
Handling Data Phase Parity Errors .....	442
General .....	442
Read Data Phase Parity Error .....	442
Write Data Phase Parity Error .....	442
General .....	442



---

## PCI System Architecture

---

Write Data Phase Parity Error on Non-Posted Write.....	446
Write Data Phase Parity Error on Posted Write.....	447
Handling Master Abort.....	448
Handling Target Abort.....	448
Handling SERR# on Secondary Side.....	449

---

### Part V: The PCI BIOS

---

#### CHAPTER 20: The PCI BIOS

Purpose of PCI BIOS.....	453
Operating System Environments Supported.....	454
General.....	454
Real-Mode.....	455
286 Protected Mode (16:16).....	456
16:32 Protected Mode.....	457
Flat Mode (0:32).....	457
Determining if System Implements 32-bit BIOS.....	458
Determining Services 32-bit BIOS Supports.....	459
Determining if 32-bit BIOS Supports PCI BIOS Services.....	459
Calling PCI BIOS.....	460

---

### Part VI: PCI Cache Support

---

#### CHAPTER 21: PCI Cache Support

Definition of Cacheable PCI Memory.....	465
Why Specification Supports Cacheable Memory on PCI Bus.....	465
Cache's Task.....	466
Intro to Write-Through vs. Write-Back Caches.....	467
Integrated Cache/Bridge.....	468
PCI Cache Support Protocol.....	472
Basics.....	472
Simple Case: Clean Snoop.....	473
Snoop Hit on Modified Line Followed by Write-Back.....	476
Treatment of Memory Write and Invalidate Command.....	479
Non-Cacheable Access Followed Immediately by Cacheable Access.....	479
Problem.....	479
Solution: Snoop Address Buffer with Two Entries.....	480
Gambling Cacheable Memory Targets.....	481
Snoop Result.....	482
When Host/PCI Bridge Doesn't Incorporate Cache.....	484

When Host/PCI Bridge Incorporates Write-Through Cache .....	484
When Host/PCI Bridge Incorporates Write-Back Cache.....	485
When Burst Transfer Crosses Line Boundary .....	485
Treatment of Snoop Result Signals on Add-in Connectors .....	486
Treatment of Snoop Result Signals After Reset .....	486

---

## Part VII: 66MHz PCI Implementation

---

### CHAPTER 22: 66MHz PCI Implementation

Introduction.....	489
66MHz Uses 3.3V Signaling Environment .....	489
How Components Indicate 66MHz Support.....	490
How Clock Circuit Sets Its Frequency .....	490
Does Clock Have to be 66MHz? .....	490
Clock Signal Source and Routing.....	491
Stopping Clock and Changing Clock Frequency .....	491
How 66MHz Components Determine Bus Speed .....	491
System Board with Separate Buses.....	492
Maximum Achievable Throughput.....	492
Electrical Characteristics .....	492
Addition to Configuration Status Register.....	494
Latency Rule .....	495
66MHz Component Recommended Pinout.....	495
Adding More Loads and/or Lengthening Bus.....	496
Number of Add-In Connectors .....	496

---

## Part VIII: Overview of VLSI Technology VL82C59x SuperCore PCI Chipset

---

### CHAPTER 23: Overview of VLSI Technology VL82C59x SuperCore PCI Chipset

Chipset Features.....	499
Intro to Chipset Members.....	500
VL82C592 Pentium Processor Data Buffer.....	501
'591/'592 Host/PCI Bridge.....	502
General.....	502
System DRAM Controller .....	502
L2 Cache.....	504
Posted-Write Buffer .....	505

## PCI System Architecture

---

General .....	505
Combining Writes Feature.....	506
Read-Around and Merge Features .....	507
Write Buffer Prioritization.....	507
Configuration Mechanism.....	508
PCI Arbitration.....	508
Locking.....	509
Special Cycle Generation .....	509
'591 Configuration Registers .....	510
Vendor ID Register.....	510
Device ID Register.....	510
Command Register.....	510
Status Register .....	512
Revision ID Register.....	512
Class Code Register.....	512
Cache Line Size Configuration Register .....	513
Latency Timer Register.....	513
Header Type Register .....	513
BIST Register.....	513
Base Address Registers.....	513
Expansion ROM Base Address Register.....	513
Interrupt Line Register.....	513
Interrupt Pin Register.....	514
Min_Gnt Register .....	514
Max_Lat Register.....	514
Bus Number Register.....	514
Subordinate Bus Number Register .....	514
PCI Device Selection (IDSEL).....	516
Handling of Host Processor-Initiated Transactions .....	517
Memory Read .....	517
Memory Write .....	518
I/O Read .....	519
I/O Write .....	519
Interrupt Acknowledge.....	519
Special Cycle .....	520
Handling of PCI-Initiated Transactions .....	521
General.....	521
PCI Master Accesses System DRAM.....	521
PCI Master Accesses PCI or ISA Memory.....	522
PCI Master Accesses Non-Existent Memory .....	522
I/O Read or Write Initiated by PCI Master.....	522

## Contents

---

Special Cycle .....	52
Type 0 Configuration Read or Write .....	52
Type 1 Configuration Read or Write .....	52
Dual-Address Command (64-bit Addressing).....	52
Support for Fast Back-to-Back Transactions .....	52
'593 PCI/ISA Bridge .....	52
'593 Handling of Transactions Initiated by PCI Masters .....	52
Subtractive Decode Capability .....	52
'593 Characteristics When Acting as PCI Master .....	52
Interrupt Support .....	52
DMA Support .....	52
'593 Configuration Registers .....	52
Vendor ID Register .....	52
Device ID Register .....	52
Command Register .....	52
Status Register .....	52
Revision ID Register .....	52
Class Code Register .....	52
Cache Line Size Configuration Register .....	53
Latency Timer Register .....	53
Header Type Register .....	53
BIST Register .....	53
Base Address Registers .....	53
Expansion ROM Base Address Register .....	53
Interrupt Line Register .....	53
Interrupt Pin Register .....	53
Min_Gnt Register .....	53
Max_Lat Register .....	53

---

## Appendices

Appendix A: Glossary .....	535
Appendix B: Resources .....	553
Index .....	555

## Figures

Figure 1-1. The X-Bus.....	12
Figure 1-2. The Teleconference .....	16
Figure 1-3. The Teleconference Screen Layout .....	17
Figure 2-1. The Direct-Connect Local Bus Approach.....	21
Figure 2-2. The Buffered Local Bus Approach.....	23
Figure 2-3. The Workstation Approach.....	25
Figure 2-4. The PCI Bus .....	32
Figure 2-5. PCI Devices Attached to the PCI Bus.....	34
Figure 4-1. Device Loads Distributed Along a Trace.....	47
Figure 4-2. CLK Signal Timing Characteristics.....	49
Figure 4-3. Timing Characteristics of Output Drivers .....	50
Figure 4-4. Input Timing Characteristics.....	51
Figure 5-1. PCI-Compliant Master Device Signals .....	54
Figure 5-2. PCI-Compliant Target Device Signals.....	55
Figure 5-3. Typical PCI Timing Diagram .....	76
Figure 6-1. The PCI Bus Arbiter.....	78
Figure 6-2. Example Arbitration Scheme .....	81
Figure 6-3. PCI Bus Arbitration Between Two Masters.....	88
Figure 6-4. Access Latency Components.....	90
Figure 6-5. Back-to-Back Transactions With an Idle State In-Between.....	105
Figure 6-6. Arbitration For Fast Back-To-Back Accesses .....	109
Figure 7-1. The PCI Interrupt Acknowledge Transaction.....	118
Figure 7-2. The Special Cycle Transaction .....	123
Figure 8-1. The Read Transaction .....	134
Figure 8-2. Optimized Read Transaction (no wait states).....	138
Figure 8-3. The PCI Write Transaction.....	143
Figure 8-4. Optimized Write Transaction (no wait states).....	145
Figure 8-5. Example of Address Stepping.....	160
Figure 9-1. Master-Initiated Termination Due to Preemption and Master Latency Timer Expiration .....	166
Figure 9-2. Example of Master-Abort on Single-Data Phase Transaction .....	169
Figure 9-3. Example of Master-Abort on Multiple Data Phase Transaction .....	170
Figure 9-4. Type "A" Disconnect.....	175
Figure 9-5. Type "B" Disconnect.....	177
Figure 9-6. Target-Initiated Retry .....	181
Figure 9-7. Target-Abort Example .....	184
Figure 10-1. Read Transaction.....	192
Figure 10-2. Write Transaction.....	195
Figure 10-3. PCI Device's Configuration Command Register.....	197
Figure 10-4. PCI Device's Configuration Status Register.....	198
Figure 10-5. Address Parity Generation/Checking.....	204



## PCI System Architecture

---

Figure 11-1. PCI Device's Configuration Header Space Format .....	208
Figure 11-2. Preferred Interrupt Design .....	212
Figure 11-3. Alternative Interrupt Layout.....	213
Figure 11-4. Typical Design In Current Machines (1993/1994).....	214
Figure 11-5. Another Typical Design In Current Machines (1993/1994).....	215
Figure 11-6. Shared Interrupt Model.....	224
Figure 12-1. Establishing the Lock to Read the Semaphore.....	236
Figure 12-2. Attempted Access to a Locked Memory Target.....	238
Figure 12-3. The Update of the Memory Semaphore and Release of Lock .....	240
Figure 13-1. REQ64# Signal Routing.....	252
Figure 13-2. Transfer Between a 64-bit Initiator and 64-bit Target .....	256
Figure 13-3. Transfer Between a 64-bit Initiator and a 32-bit Target .....	259
Figure 13-4. Single Data Phase 64-bit Transfer With a 64-bit Target .....	265
Figure 13-5. Dual-Data Phase 64-bit Transfer With a 32-bit Target.....	266
Figure 13-6. 32-bit Initiator Reading From Address Above 4GB .....	270
Figure 13-7. 64-bit Initiator Reading From Address Above 4GB With 64-Bit Data Transfers.....	273
Figure 14-1. 32 and 64-bit Connectors .....	280
Figure 14-2. 3.3V, 5V and Universal Cards.....	286
Figure 14-3. ISA/EISA Unit Expansion Slots.....	287
Figure 14-4. Micro Channel Unit Expansion Slots.....	288
Figure 14-5. Recommended PCI Component Pinout Ordering.....	290
Figure 15-1. PCI Functional Device's Basic Configuration Address Space Format....	298
Figure 15-2. System With a Single PCI Bus.....	300
Figure 16-1. The Configuration Address Register at 0CF8h.....	309
Figure 16-2. Peer Host/PCI Bridges .....	310
Figure 16-3. Configuration Space Enable, or CSE, Register.....	315
Figure 16-4. PowerPC PReP Memory Map.....	318
Figure 16-5. Contents of the AD Bus During Address Phase of a Type Zero Configuration Access.....	322
Figure 16-6. The Type Zero Configuration Read Access .....	323
Figure 16-7. The Type Zero Configuration Write Access .....	324
Figure 17-1. Format of a PCI Device's Configuration Header.....	328
Figure 17-2. The Command Register Bit Assignment.....	331
Figure 17-3. Status Register Bit Assignment.....	335
Figure 17-4. Class Code Register .....	336
Figure 17-5. Header Type Register Format.....	341
Figure 17-6. The BIST Register.....	344
Figure 17-7. Memory Base Address Register Format .....	346
Figure 17-8. I/O Base Address Register Format .....	347
Figure 17-9. Expansion ROM Register Format .....	351

## Figure

Figure 18-1. Format of Expansion ROM Base Address Register.....	35
Figure 18-2. Header Type Zero Configuration Register Format.....	36
Figure 18-3. Multiple Code Images Contained In One Device ROM.....	36
Figure 18-4. Code Image Format.....	36
Figure 19-1. Example System One.....	37
Figure 19-2. Example System Two.....	38
Figure 19-3. PCI-to-PCI Bridge's Configuration Registers.....	38
Figure 19-4. Header Type Register.....	38
Figure 19-5. Class Code Register.....	39
Figure 19-6. I/O Base Register.....	39
Figure 19-7. I/O Limit Register.....	39
Figure 19-8. Example of I/O Filtering Actions.....	39
Figure 19-9. ISA I/O Expansion I/O Ranges.....	40
Figure 19-10. Prefetchable Memory Base Register.....	40
Figure 19-11. Prefetchable Memory Limit Register.....	41
Figure 19-12. Memory (mapped I/O) Base Register.....	41
Figure 19-13. Memory (mapped I/O) Limit Register.....	41
Figure 19-14. Command Register.....	41
Figure 19-15. Bridge Control Register.....	41
Figure 19-16. Contents of the AD Bus During Address Phase of a Type One Configuration Access.....	42
Figure 19-17. The Type One Configuration Read Access.....	42
Figure 19-18. The Type One Configuration Write Access.....	42
Figure 19-19. Example System.....	43
Figure 19-20. Posted-Write Scenario (refer to text).....	43
Figure 19-21. Another Deadlock Scenario (refer to text).....	43
Figure 21-1. The Integrated Cache/Bridge Element.....	46
Figure 21-2. A Simple Clean Snoop.....	47
Figure 21-3. Writeback Caused by a Snoop Hit on a Modified Line.....	47
Figure 21-4. The Snoop Protocol Signals.....	48
Figure 22-1. 33 versus 66MHz Timing.....	49
Figure 23-1. System Design Using VLSI VL82C59x SuperCore Chipset.....	50
Figure 23-2. Rotational Priority Scheme.....	50
Figure 23-3. '591 PCI Configuration Registers.....	51
Figure 23-4. VL82C593's Configuration Registers.....	53



## Tables

Table 1-1. Teleconferencing Transfer Rate Requirements.....	15
Table 1-2. Required Subsystem Transfer Rates .....	15
Table 2-1. VL Bus Characteristics .....	27
Table 2-2. Major PCI Revision 2.1 Features.....	31
Table 2-3. This Book is Based on.....	34
Table 5-1. Byte Enable Mapping To Data Paths and Locations Within the Currently-Addressed Doubleword .....	60
Table 5-2. Interpretation of the Byte Enables During a Data Phase .....	60
Table 5-3. PCI Interface Control Signals.....	63
Table 5-4. Cache Snoop Result Signals .....	68
Table 5-5. The 64-Bit Extension.....	69
Table 5-6. Boundary Scan Signals .....	70
Table 5-7. PCI Signal Types .....	72
Table 6-1. Bus State .....	85
Table 6-2. Access Latency Components .....	89
Table 6-3. Ordering Rules.....	100
Table 7-1. PCI Command Types .....	114
Table 7-2. Message Types defined In the Specification.....	120
Table 7-3. Read Policy When Cache Line Size Register Implemented.....	125
Table 7-4. Read Policy When Cache Line Size Register Not Implemented .....	125
Table 8-1. Memory Burst Address Sequence .....	150
Table 8-2. Examples of I/O Addressing.....	151
Table 8-3. Qualification Requirements .....	156
Table 9-1. Target-Initiated Termination Summary.....	185
Table 11-1. Value To Be Hardwired Into Interrupt Pin Register.....	208
Table 11-2. Interrupt Line Register Values.....	217
Table 11-3. ISA Interrupt Vectors .....	226
Table 11-4. Interrupt Priority Scheme .....	228
Table 14-1. PCI Add-In Card Pinouts.....	281
Table 14-2. Card Power Requirement Indication On Card Present Signals.....	284
Table 14-3. Required Power Supply Current Sourcing Capability (per connector).....	291
Table 16-1. EISA PC I/O Space Usage .....	306
Table 16-2. Sub-Ranges Within C000h through CFFFh I/O Range.....	314
Table 16-3. PREP Memory - to - Configuration Mapping.....	317
Table 17-1. The Command Register.....	330
Table 17-2. The Status Register Bits .....	334
Table 17-3. Defined Class Codes.....	336
Table 17-4. Class Code 0 (pre rev 2.0) .....	337
Table 17-5. Class Code 1: Mass Storage Controllers .....	337
Table 17-6. Class Code 2: Network Controllers.....	337
Table 17-7. Class Code 3: Display Controllers.....	337

## PCI System Architecture

---

Table 17-8. Class Code 4: Multimedia Devices.....	338
Table 17-9. Class Code 5: Memory Controllers.....	338
Table 17-10. Class Code 6: Bridge Devices .....	338
Table 17-11 Class Code 7: Simple Communications Controllers .....	338
Table 17-12 Class Code 8: Base System Peripherals.....	339
Table 17-13. Class Code 9: Input Devices .....	339
Table 17-14. Class Code A: Docking Stations .....	339
Table 17-15. Class Code B: Processors.....	339
Table 17-16. Class Code C: Serial Bus Controllers .....	340
Table 17-17. Definition of IDE Programmer's Interface Byte Encoding .....	340
Table 17- 18. The BIST Register Bit Assignment.....	343
Table 17-19. Bit-Assignment of the CardBus CIS Pointer Register .....	351
Table 18-1. PCI Expansion ROM Header Format.....	365
Table 18-2. PC-Compatible Usage of Processor /Architecture Unique Data Area In ROM Header.....	366
Table 18-3. PCI Expansion ROM Data Structure Format.....	367
Table 19-1. Transaction Types That the Bridge Must Detect and Handle.....	383
Table 19-2. IBM PC and XT I/O Address Space Usage .....	400
Table 19-3. Example I/O Address .....	402
Table 19-4. Usable and Unusable I/O Address Ranges Above 03FFh .....	403
Table 19-5. Command Register Bit Assignment .....	416
Table 19-6. Bridge Control Register Bit Assignment.....	418
Table 19-7. Configuration Transactions That May Be Detected On the Two Buses....	421
Table 19-8. Target Device Number to AD Line Mapping (for IDSEL assertion).....	429
Table 19-9. Interrupt Routing on Add-in Card With PCI-to-PCI Bridge.....	437
Table 20-1. 32-Bit BIOS Data Structure.....	458
Table 20-2. PCI BIOS Function Request Codes.....	461
Table 21-1. Write-Through Cache Action Table .....	470
Table 21-2. Write-Back Cache Action Table .....	471
Table 21-3. Memory Target Interpretation of Snoop Result Signals from Bridge.....	483
Table 22-1. 66MHz Timing Parameters .....	494
Table 22-2. Combinations of 66MHz-Capable Bit Settings .....	495
Table 23-1. '591 Command Register Bit Assignment .....	511
Table 23-2. '591's Status Register Bit Assignment .....	512
Table 23-3. Device Number To AD Line Mapping.....	516
Table 23-4. '593's Command Register Bit Assignment .....	528
Table 23-5. '593's Status Register Bit Assignment .....	529

---

xxx

---

## **Acknowledgments**

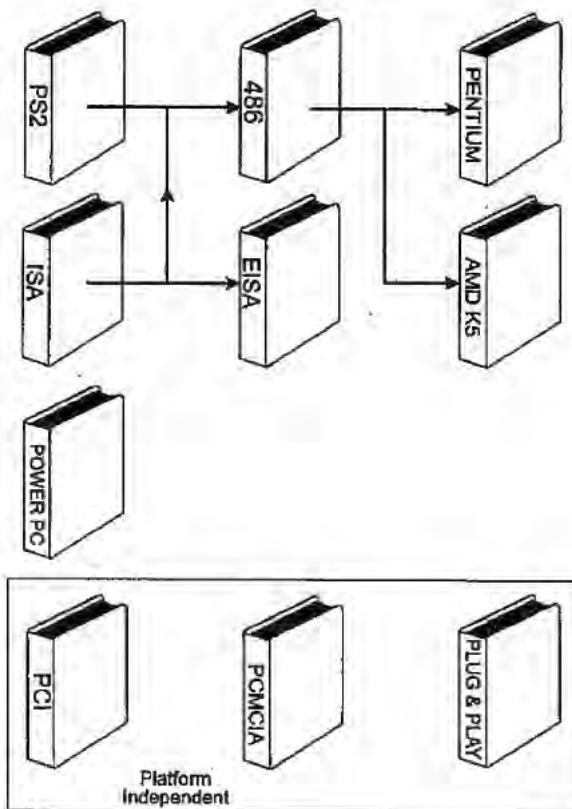
---

To John Swindle for his tireless attention to detail and his marvelous-teaching ability. To the editorial staff at Addison-Wesley for their patience. To the folk at Computer Literacy Bookshops for their collective support in the initial launch of this book series. And finally, to the many hundreds of engineers at Intel, IBM, Compaq, Dell, Hewlett-Packard, Motorola, and other clients, who subject themselves to our teaching on a regular basis.

The MindShare Architecture Series

The MindShare Architecture book series includes: *ISA System Architecture*, *EISA System Architecture*, *80486 System Architecture*, *PCI System Architecture*, *Pentium System Architecture*, *PCMCIA System Architecture*, *PowerPC System Architecture*, *Plug-and-Play System Architecture*, and *AMD K5 System Architecture*.

Rather than duplicating common information in each book, the series uses the building-block approach. *ISA System Architecture* is the core book upon which the others build. The figure below illustrates the relationship of the books to each other.



Series Organization

---

# PCI System Architecture

---

## Organization of This Book

The third edition of *PCI System Architecture* has been updated to reflect revision 2.1 of the PCI bus specification. In addition, it has been completely reorganized and expanded to include more detailed discussions of virtually every topic found in the first two editions. The book is divided into eight parts:

- **Part I: Intro to the Local Bus Concept.** Defines the performance problems inherent in PC architecture before the introduction of the local bus. Having defined the problem, the possible solutions are explored.
- **Part II: Revision 2.1 PCI Essentials.** This part of the book provides a detailed explanation of the mainstream aspects of PCI bus operation.
- **Part III: Device Configuration In a System With a Single PCI Bus.** Provides an introduction to the PCI configuration address space, a detailed description of the methods for generating configuration bus transactions, the configuration read and write transactions timing, the configuration registers defined by the specification, and the implementation of expansion ROMs associated with PCI devices.
- **Part IV: The PCI-to-PCI Bridge.** This part provides a detailed discussion of the PCI-to-PCI Bridge specification, a discussion of peer and hierarchical PCI buses, and the accessing of configuration registers in devices residing on subordinate PCI buses.
- **Part V: The PCI BIOS.** This part provides a detailed discussion of the PCI BIOS specification.
- **Part VI: Support for Cacheable PCI Memory.**
- **Part VII: 66MHz PCI Implementation.**
- **Part VIII: Overview of VLSI VL82C59x Supercore Chipset.** This part provides an operational overview of the VLSI chip set.

---

## Who this Book is For

This book is intended for use by hardware and software design and support personnel. Due to the clear, concise explanatory methods used to describe each subject, personnel outside of the design field may also find the text useful.

### Prerequisite Knowledge

It is highly recommended that the reader have a good knowledge of PC and processor bus architecture prior to reading this book. The MindShare publications entitled *ISA System Architecture* and *80486 System Architecture* provide all of the background necessary for a complete understanding of the subject matter covered in this book. Alternately, the reader may substitute *Pentium System Architecture* or *PowerPC System Architecture* for *80486 System Architecture*.

### Object Size Designations

The following designations are used throughout this book when referring to the size of data objects:

- A byte is an 8-bit object.
- A word is a 16-bit, or two byte, object.
- A doubleword is a 32-bit or four byte, object.
- A quadword is a 64-bit, or eight byte, object.
- A paragraph is a 128-bit, or 16 byte, object.
- A page is a 4K-aligned 4KB area of address space.

### Documentation Conventions

This section defines the typographical convention used throughout this book.

#### Hex Notation

All hex numbers are followed by an "h." Examples:

9A4Eh  
0100h

#### Binary Notation

All binary numbers are followed by a "b." Examples:

0001 0101b  
01b

---

## PCI System Architecture

---

### Decimal Notation

Numbers without any suffix are decimal. When required for clarity, decimal numbers are followed by a "d." The following examples each represent a decimal number:

16  
255  
256d  
128d

---

### Signal Name Representation

Each signal that assumes the logic low state when asserted is followed by a pound sign (#). As an example, the TRDY# signal is asserted low when the target is ready to complete a data transfer.

Signals that are not followed by a pound sign are asserted when they assume the logic high state. As an example, IDSEL is asserted high to indicate that a PCI device's configuration space is being addressed.

---

### Identification of Bit Fields (logical groups of bits or signals)

All bit fields are designated in little-endian bit ordering as follows:

[X:Y],

where "X" is the most-significant bit and "Y" is the least-significant bit of the field. As an example, the PCI address/data bus consists of AD[31:0], where AD31 is the most-significant and AD0 the least-significant bit of the field.

---

### We Want Your Feedback

MindShare values your comments and suggestions. You can contact us via mail, phone, fax or internet email.



## About This Book

---

Phone: (214) 231-2216  
Fax: (214) 783-4715  
E-mail: [mindshar@interserv.com](mailto:mindshar@interserv.com)

To request information on public or private seminars, email your request to: [mindshar@interserv.com](mailto:mindshar@interserv.com) or call our bulletin board at (214) 705-9604.

---

### Bulletin Board

Because we are constantly on the road teaching, we can be difficult to get hold of. To help alleviate problems associated with our migratory habits, we have initiated a bulletin board to supply the following services:

- Download of course abstracts.
- Download of tables of contents of each book in the series.
- Facility to inquire about public architecture seminars.
- Message area to log technical questions.
- Message area to log suggestions for book improvements.
- Facility to view book errata and clarifications.

The bulletin board may be reached 24-hours a day, seven days a week.

BBS phone number: (214) 705-9604

---

### Mailing Address

MindShare, Inc.  
2202 Buttercup Drive  
Richardson, Texas 75082

---

*Part I*

*Introduction to the  
Local Bus Concept*

# Chapter 1

## In This Chapter

This chapter defines the performance constraints experienced when devices that perform block data transfers are placed on the expansion bus (e.g., the ISA, EISA and Micro Channel buses). It also uses the performance requirements of teleconferencing to highlight the bandwidth requirements of systems requiring fast block transfers between multiple subsystems in order to achieve superior system performance.

## The Next Chapter

The next chapter introduces the concept of the local bus. The VESA VL bus and the PCI bus implementations of the local bus are introduced as solutions to the throughput problem.

---

## Block-Oriented Devices

In today's operating environments, it is imperative that large block data transfers be accomplished expeditiously. This is especially true in relation to the following types of subsystems:

- Graphics video adapter.
- Full-motion video adapter.
- SCSI host bus adapter.
- FDDI network adapter.

---

## Graphics Interface Performance Requirements

The Windows, OS/2 and Unix X-Windows user interfaces require extremely fast updates of the graphics image in order to move, resize and update multiple windows without imposing discernible delays on the end-user. Since the

## **PCI System Architecture**

---

screen image is stored in video RAM, this means that the processor must be able to update and/or move large blocks of data within video memory very fast. The same is true for the updating of full-motion video in video ram.

---

### **SCSI Performance Requirements**

The SCSI interface is used to move large blocks of data between target I/O devices and system memory. Mass storage devices such as hard disk drives, CD-ROM drives and tape backup subsystems typically reside on the SCSI bus. The time required to read or write files on hard drives or tape, or to read files from CD-ROM can impose delays on the end-user. Anything that can be done to speed up these block data transfers has a significant effect on overall system performance.

---

### **Network Adapter Performance Requirements**

When a network adapter is used to transfer entire files of information to or from a server (a print or file server), the rate at which the information can be transferred between system memory and the network adapter detracts from or contributes to overall system performance.

---

### **X-Bus Device Performance Constraints**

The devices just described are just some examples of subsystems that benefit significantly from a fast transfer rate. Unfortunately, the majority of subsystems reside on the PC's expansion bus. Depending on the machine's design, this may be the ISA, EISA or Micro Channel expansion bus. As described later in this chapter, all three of these expansion bus architectures suffer from an inadequate data transfer rate.

In many cases, subsystems such as the graphics video adapter have been integrated onto the system board. This would seem to imply that they do not reside on the expansion bus, but this is not the case. Most of the integrated subsystems reside on a buffered version of the expansion bus known as the X-bus (eXtension to the expansion bus; also referred to as the utility bus). This being the case, these subsystems are bound by the mediocre transfer rates achievable when communicating with devices residing on the expansion bus. Figure 1-1 illustrates the relationship of the X-bus to the expansion bus and the system board microprocessor.

## Chapter 1: The Problem

---

When performing memory reads, the microprocessor can communicate with its internal (level one, or L1) cache at its full native speed if the requested information is found in the cache. If the cache is implemented as a write-back cache, memory writes to currently-cached locations may also be completed at full speed. When an L1 cache miss occurs on a memory read or the cache must write information into memory, the processor must use its local bus to communicate with memory. The memory access request is first submitted to the external, level 2 (L2) cache for fulfillment. In the event of an L2 cache miss, the L2 cache performs an access to system DRAM memory. The linkage between the L2 cache and system DRAM memory is typically optimized to allow information transfers to complete as quickly as possible.

When a memory read or write addresses memory other than system DRAM memory or when the processor is performing an I/O read or write, the expansion bus bridge must pass the bus cycle through to the expansion bus. The completion of the bus cycle is bound by the maximum expansion bus speed and the access time of the expansion bus device being accessed. If a large amount of data is to be transferred to or from the target expansion device, performance is bound by the speed of the bus, the access time of the target, and the expansion bus data bus width.

# PCI System Architecture

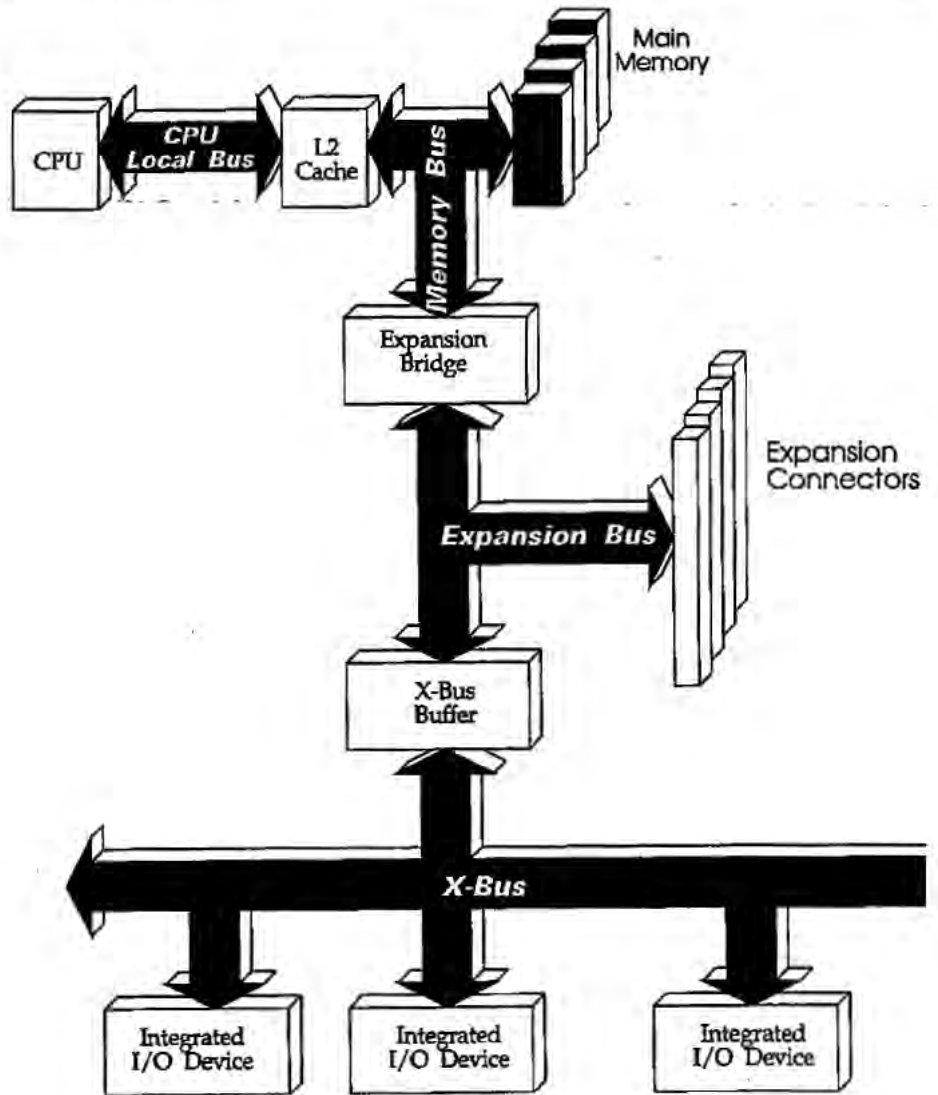


Figure 1-1. The X-Bus



### Expansion Bus Transfer Rate Limitations

---

#### ISA Expansion Bus

All transfers performed over the ISA bus are synchronized to an 8MHz (more typically, 8.33MHz) bus clock signal (BCLK). It takes a minimum of two cycles of the bus clock (if the target device is a zero wait state device) to perform a data transfer. This equates to 4.165 million transfers per second. Since the data path on the ISA bus is only 16-bits wide, a maximum of two bytes may be transferred during each transaction. This equates to a theoretical maximum transfer rate of 8.33 MBytes per second.

For more information on the ISA expansion bus, refer to the Addison-Wesley publication entitled *ISA System Architecture*.

#### EISA Expansion Bus

Like the ISA bus, all transfers performed over the EISA bus are synchronized to an 8MHz (more typically, 8.33MHz) bus clock signal (BCLK). It takes a minimum of one cycle of the bus clock (if the target device supports EISA burst mode transfers) to perform a data transfer. This equates to 8.33 million transfers per second. Since the data path on the EISA bus is 32-bits wide, a maximum of four bytes may be transferred during each transaction. This equates to a theoretical maximum transfer rate of 33 Mbytes per second.

For more information on the EISA expansion bus, refer to the Addison-Wesley publication entitled *EISA System Architecture*.

#### Micro Channel Architecture Expansion Bus

At the current time, the maximum achievable transfer rate on the Micro Channel (as implemented in the PS/2 product line) is 40Mbytes per second (using the 32-bit Streaming Data Procedure). This is based on a 10MHz bus speed with one data transfer taking place during each cycle of the 10MHz clock (10 million transfers per second \* four bytes per transfer). Faster transfer rates of 80 and 160Mbytes per second are possible when the 64-bit and enhanced 64-bit Streaming Data Procedures are implemented.

## PCI System Architecture

---

### Teleconferencing Performance Requirements

Figure 1-2 illustrates three PCs linked via a telecommunications network. Each of the three units has the capability to simultaneously merge multiple graphics and video sources onto the screen in real-time. Figure 1-3 illustrates the contents of each screen.

The large portion of the screen (devoted to a graphics image) is utilized to display the document under discussion. In order to successfully emulate an actual face-to-face conferencing situation, the system must be capable of updating this image fast enough to simulate flipping through the pages of a document at the rate of ten pages (or frames) per second. With an image resolution of 1280 x 1024 pixels and color resolution of 16 million colors (three bytes per pixel), the amount of video memory required to store one image is 3.93216Mbytes. To alter the graphics display at the rate of ten frames per second would require a video memory update rate of 39.3216Mbytes per second.

The video preview portion of the screen is used to display a real-time video image of a video source local to the unit. This image has a resolution of 320 x 240 pixels and a color resolution of 256 colors (one byte per pixel). In order to provide full-motion video, the image must be updated at the rate of thirty frames per second. The amount of video memory required to store one image would be 76.8Kbytes. To alter the graphics display at the rate of thirty frames per second would require a video memory update rate of 2.3Mbytes per second.

Each of the two video remote screen areas is used to display a full-motion video image from one of the other two participants. These images each have a resolution of 640 x 480 pixels and a color resolution of 256 colors (one byte per pixel). In order to provide full-motion video, each image must be updated at the rate of thirty frames per second. The amount of video memory required to store one image would be 307.2Kbytes. To alter each of the video remote windows at the rate of thirty frames per second would require a video memory update rate of 9.2Mbytes per second.

Each of the three video images would be transmitted in compressed video image format at the rate of 200Kbytes per second per video stream.

In summary, each host system must supply sufficient bus bandwidth to support the combined transfer rates required to update the images presented in the graphics, preview, remote one and remote two windows, as well as the

## Chapter 1: The Problem

three 200Kbyte per second compressed video streams. The bus structure must then support the simultaneous transfer rates listed in table 1-1. ISA (8.33Mbytes per second) and the current version of EISA (33Mbytes per second) will not support the combined bandwidth requirement of 60.516Mbytes per second. The Micro Channel (40Mbytes per second) does not currently support the required rate, but the 64-bit Streaming Data Procedures (not supported on the PS/2 product line) are able to achieve transfer rates of 80 to 160Mbytes per second. As described later in this document, the PCI bus currently supports a transfer rate of 132Mbytes per second. If the 64-bit PCI extension is implemented, a transfer rate of 264Mbytes per second can be achieved. Table 1-2 lists the transfer rates for the video and other subsystems.

*Table 1-1. Teleconferencing Transfer Rate Requirements*

Screen Element	Transfer Rate (Mbytes/second)
Graphics Window	39.216
Preview Window	2.3
Video Remote One	9.2
Video Remote Two	9.2
Preview Compressed Video Stream	.2
Video Remote One Compressed Video Stream	.2
Video Remote Two Compressed Video Stream	.2
Total transfer rate required to support teleconferencing	60.516

*Table 1-2. Required Subsystem Transfer Rates*

Subsystem	(Mbytes per second)
Graphics	30 to 40
Full-Motion Video	2 to 9 per window
LAN	15 for FDDI (Fiber Distributed Data Interface)
	3 for Token Ring
	2 for Ethernet
Hard Disk	5 to 20 using SCSI
CD-ROM	2 using SCSI
Audio	1 for CD quality output

# PCI System Architecture

---

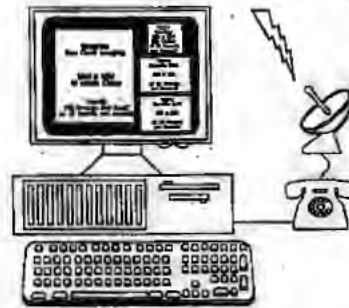
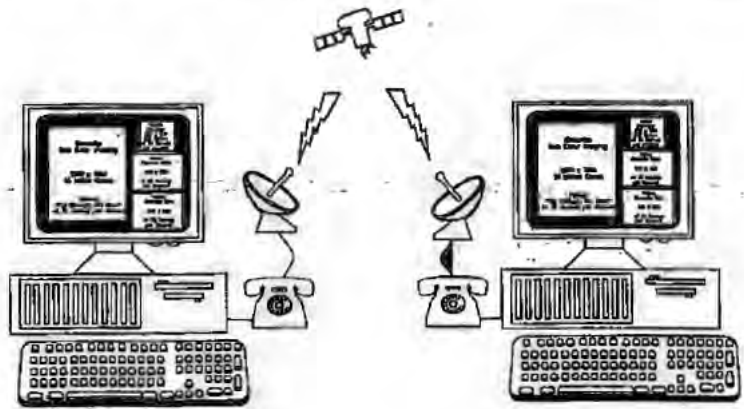


Figure 1-2. The Teleconference

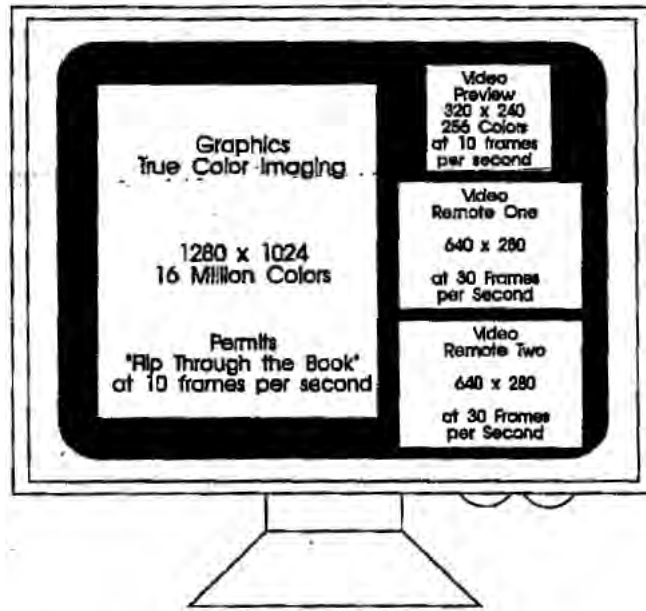


Figure 1-3. The Teleconference Screen Layout

# Chapter 2

## The Previous Chapter

The previous chapter discussed the performance constraints placed upon subsystems installed on the expansion bus or integrated onto the system board's X-bus.

## This Chapter

This chapter introduces the concept of the local bus and provides an overview of the two major local bus standards:

- the VESA VL bus
- the PCI bus

## The Next Chapter

The next chapter provides an introduction to the PCI transfer mechanism.

---

## Graphics Accelerators: Before Local Bus

An interim attempt to improve the performance of video graphics adapters implemented as expansion bus devices involved the enhancement of the adapter's intelligence. Earlier adapters processed very low-level commands issued by the microprocessor. The processor and therefore the programmer had to be intimately involved in every aspect of screen management. Later adapters are frequently based on processors like the Intel i860XR/XP or the Texas Instruments TMS34010/34020 and can handle high-level commands to off-load screen-intensive operations from the microprocessor. As an example, a BITBLT command can be issued to the adapter, causing it to quickly move a window graphic from one area of video memory to another without any further intervention on the microprocessor's part. The video memory is on the expansion adapter card and can therefore be accessed directly by the adapter's local processor at high speed.



---

## PCI System Architecture

---

### Local Bus Concept

To maximize throughput when performing updates to video graphics memory, many PC vendors have moved the video graphics adapter from the slow expansion bus to the processor's local bus. Figure 2-1 illustrates the processor's local bus. The video adapter is redesigned to connect directly to the processor's local bus and the adapter design is optimized to minimize or eliminate the number of wait states inserted into each bus cycle when the processor accesses video memory and the adapter's I/O registers. In addition, the video graphics adapter typically also incorporates a local processor and can handle high-level commands (as discussed earlier).

---

### Direct-Connect Approach

There are three basic methods for connecting a device to the microprocessor's local bus. The first scenario is pictured in figure 2-1 and is very straightforward: the device is connected directly to the processor's bus structure. This could be any processor type (such as the 486). As an example, when the 486 performs zero wait-state bus cycles at its maximum rated speed of 33MHz (the actual bus speed is processor implementation-dependent), read burst transfers can be performed at the rate of 132Mbytes per second (if the processor is communicating with video memory that supports burst mode and is cacheable). When performing memory writes to update the video frame buffer in memory, the programmer may specify no more than four bytes to be written to memory per bus cycle. If the video memory supports zero wait-state writes, this would permit a data transfer rate of 66Mbytes/second. The direct-connect approach imposes a number of important design constraints:

- Since the device is connected directly to the processor's local bus, it must be redesigned in order to be used with next generation processors (if the bus structure or protocol are altered).
- Due to the extra loading placed on the local bus, no more than one local bus device may be added.
- Because the local bus is running at a high frequency, the design of the local bus device's bus interface is difficult.
- Although the system may work when it's shipped, it may exhibit aberrant behavior when an Intel Overdrive Processor is installed in the upgrade socket (thereby placing another load on the local bus).
- It does not permit the processor to perform transfers with one device while the local bus device is involved in a transfer with another device.

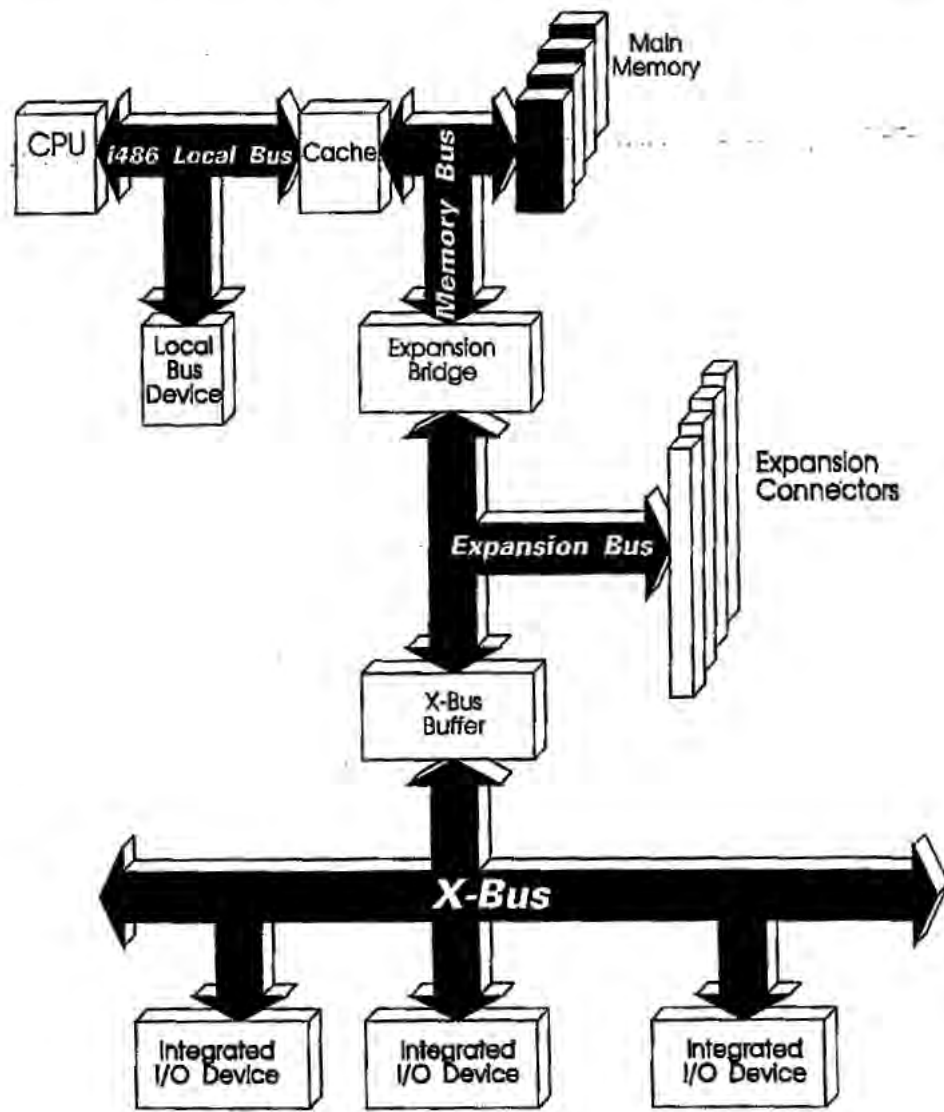


Figure 2-1. The Direct-Connect Local Bus Approach

## PCI System Architecture

---

### Buffered Approach

The second approach that can be utilized in connecting a local bus device to the processor's local bus is the buffered approach. Figure 2-2 illustrates this scenario. The buffer/driver redrives all of the local bus signals, thereby permitting fanout to more than one local bus device. Since the buffered local bus is electrically-isolated from the microprocessor's local bus, it only presents one load to the microprocessor's local bus. Typically, a maximum of three local bus devices can be placed on the buffered local bus. This is the only real advantage of this approach over the direct-connect approach.

A major disadvantage of the buffered approach is that the processor's local bus and the buffered local bus are essentially one bus. Any transaction initiated by the processor appears on the local and buffered local buses. Likewise, any bus transaction initiated by a bus master that resides on the buffered local bus appears on both the buffered local bus and the processor's local bus. In other words, either the processor or a local bus master may use the bus, but not both simultaneously. If a local bus master is using the bus and the processor requires the bus to perform a transaction, the processor is stalled until the bus master surrenders ownership of the bus. The reverse situation is also true.

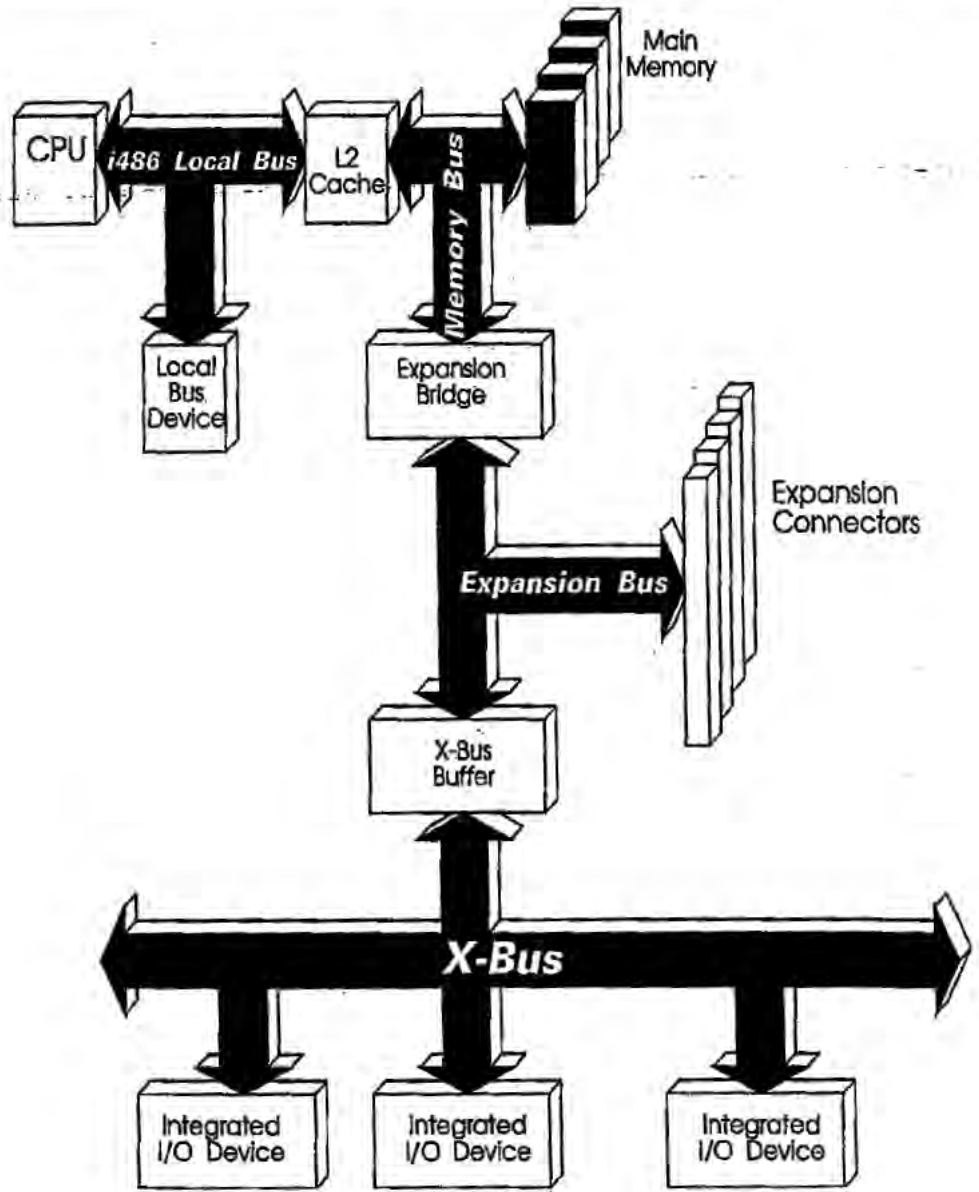


Figure 2-2. The Buffered Local Bus Approach

## PCI System Architecture

---

### Workstation Approach

Figure 2-3 illustrates an approach used in many workstation architectures to achieve high performance. The processor's L2 cache controller is combined with a bridge that provides the interface between the processor, main memory and the high-speed I/O bus (in this case, the PCI bus). The devices that reside on the I/O bus may consist only of target devices or a mixture of targets and intelligent peripheral adapters with bus master capability. Via the specially-designed bridge, either the processor (through its L2 cache) or a bus master on the I/O bus (or the expansion bus) can access main memory. Optimally, the processor can continue to fetch information from its L1 or L2 cache while the cache controller provides a bus master on the I/O bus with access to main memory. Bus masters on the I/O bus can also communicate directly with target devices on the I/O bus while the processor is accessing its L1 or L2 cache or while the L2 cache controller is accessing main memory for the processor.

Another very distinct advantage of this approach is that it renders the I/O bus device interface independent of the processor bus. Processor upgrades can be easily implemented without impacting the design of the I/O bus and its associated devices. Only the cache/bridge would require a redesign (to match the new host processor interface).

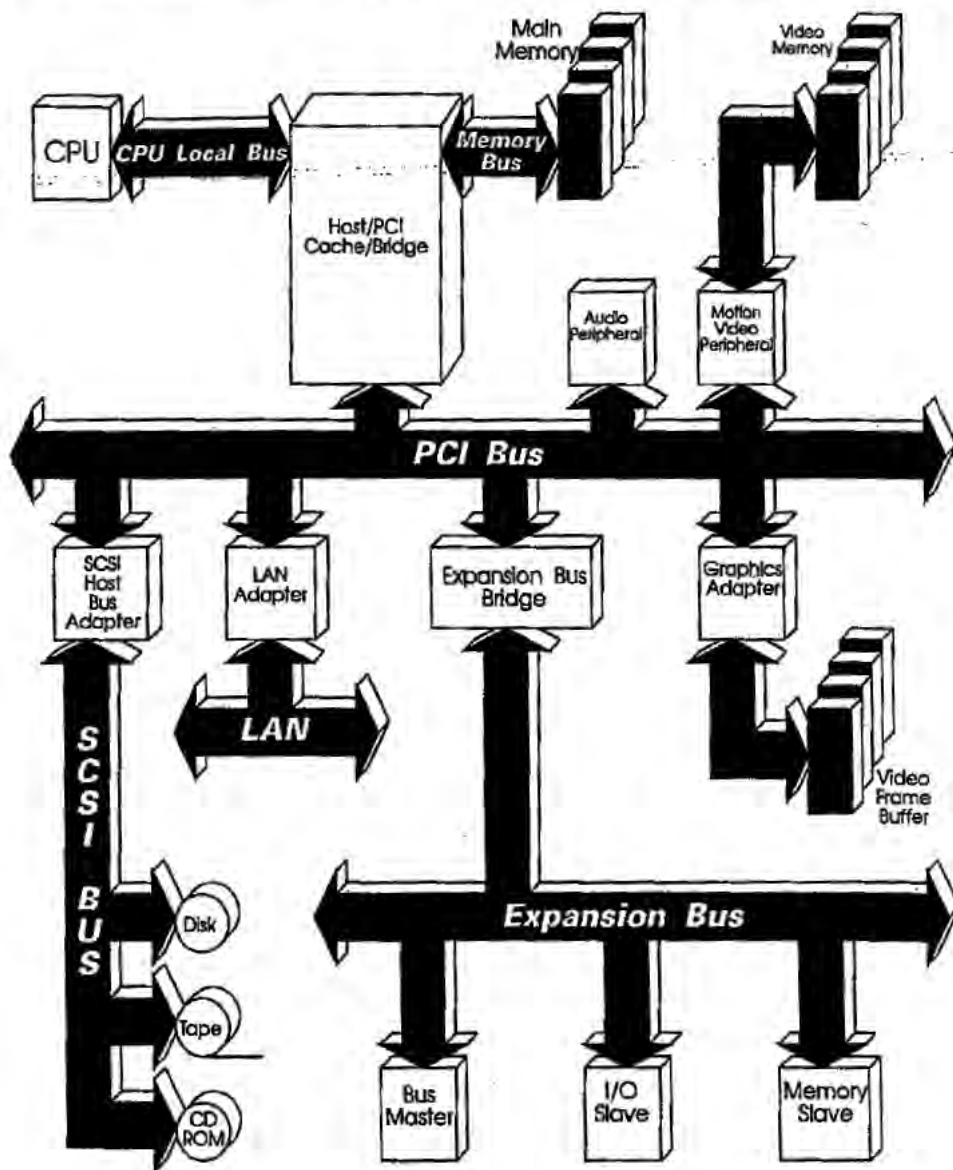


Figure 2-3. The Workstation Approach



## PCI System Architecture

---

A variation on this theme would find the processor's L2 cache implemented as a lookaside cache located on the processor's local bus. In this configuration, main memory may be located either on a dedicated memory bus (as shown in figure 2-3) or it may reside on the processor's local bus along with the lookaside L2 cache. If the main memory is located on the processor's local bus, it should be noted that it can only be accessed by the performance of a memory read or write transaction on the processor's local bus. This is true even if a bus master located on the I/O bus is accessing main memory. This could diminish the processor's performance by diminishing the local bus availability. The reverse would also be true: bus masters other than the host processor cannot access main memory while the processor is utilizing its local bus.

---

## VESA VL Bus Solution

Until several years ago, there existed no standard that defined the interconnection schema used to integrate local bus devices into the PC environment. The Video Electronics Standards Association (VESA), an association of companies involved in the design and manufacturing of video graphics adapters, commissioned the development of a local bus standard. The preliminary specification was completed and refers to the local bus as the VL bus (VESA Local bus). The initial version of the VESA specification, version 1.0, defines two methods of interfacing to the microprocessor's local bus: the direct-connect and the buffered approaches described earlier. The direct-connect approach is referred to as the VL Type "A" bus, while the buffered version is referred to as the VL Type "B" bus. In both cases, the bus is modeled on the 486 bus. Some characteristics of each implementation are listed in table 2-1. A brief description of each of the listed characteristics follows the table.

## Chapter 2: Solutions, VESA and PCI

Table 2-1. VL Bus Characteristics

Characteristics	Type "A"	Type "B"
logic cost	\$0.	Cost of buffering.
Performance	At a bus speed of 33MHz, 132Mbytes/second (peak) on burst reads, and 66Mbytes/second on write transfers.	Same as type "A", but the delay imposed by the buffer almost certainly causes wait states to be inserted in each transfer.
Longevity	Tied to 386/486 bus structure.	Tied to 386/486 bus structure.
Teleconferencing Support	One local bus device.	Three local bus devices.
Electrical Integrity	Not defined.	Not defined.
Modularity	None.	Three Micro Channel connectors.
Auto-Configuration	Supports Auto Configuration (see "Auto-Configuration" section later in this chapter).	Supports Auto-Configuration (see "Auto-Configuration" section later in this chapter).

### Logic Cost

No additional system board logic is necessary to implement a VL Type "A" local bus device. The device is connected directly to the microprocessor's local bus. In a Type "B" design, the cost of the buffering logic must be taken into account.

### Performance

Using the type "A" and "B" approaches, a peak data transfer rate of 132Mbytes per second may be achieved (at a processor bus speed of 33MHz). It should be noted that the longest burst read performed by the 486 processor occurs during a cache line fill operation. Sixteen bytes (four doublewords) are transferred to the processor during the cache line fill. The first doubleword takes two processor clocks, while the subsequent three doublewords may be transferred back to the microprocessor at the rate of one per processor clock cycle (if the access time of the target device supports this speed).

The 486 processor is only capable of performing burst writes under the following circumstances:

- When it attempts to write two to four bytes (in one bus cycle) to an 8-bit device (BS8# is sampled asserted). An 8-bit device that supports burst mode operation can achieve a transfer rate of 33Mbytes/second (one byte transferred during each processor clock cycle), but it should be noted that

---

## PCI System Architecture

---

this rate can only be sustained for the transfer of up to three successive bytes.

- When it attempts to write two to four bytes (in one bus cycle) to a 16-bit device (BS16# is sampled active). A 16-bit device that supports burst mode operation can achieve a transfer rate of 66Mbytes/second, but it should be noted that this rate can only be sustained for the transfer of one 16-bit object.

When performing 32-bit non-burst write transfers, the 486 microprocessor can achieve a maximum transfer rate of 66Mbytes/second (two processor clock cycles per 32-bit transfer).

It should be noted that the VL bus specification defines bus speeds up to a maximum frequency of 66MHz. All performance estimates quoted in this publication are based on a maximum bus speed of 33MHz because this is the achievable norm at the current time.

---

### Longevity

Both the type "A" and "B" approaches are short term solutions because they are designed around the and 486 processor bus structure. The interface logic must be redesigned for next generation processors with more advanced bus structures. Bridge logic would be necessary to translate between the new processor bus and the VL bus.

---

### Teleconferencing Support

The type "A" approach does not offer a teleconferencing solution because it provides support for only one local bus device. At a bare minimum, teleconferencing requires high-speed support for at least two peripheral subsystems: the graphics and full-motion video adapters. The type "B" solution provides minimal teleconferencing support by supporting up to three local bus peripherals.

---

### Electrical Integrity

The VESA VL 1.0 bus specification provides no electrical design guidelines to ensure the integrity of local bus design. System board designers must design the PCI system board layout from scratch. While this isn't a problem at low

## Chapter 2: Solutions, VESA and PCI

---

bus speeds, buses running at today's accelerated rates present a formidable design challenge.

---

### Add-In Connectors

Modularity refers to the ability to add new local bus peripherals by installing an option card into a local bus connector. Type "A" solutions are direct-connect and do not provide a connector. Type "B" solutions can support up to three connectors. The VL specification defines a Micro Channel-style connector as the expansion vehicle.

---

### Auto-Configuration

The VESA VL 1.0 specification states that VL local bus devices must support automatic system configuration. However, the specification does not define the standard automatic configuration support that must be provided in each VL bus-compliant local bus device. The specification also states that VL bus-compliant local bus devices must be transparent to device drivers. In other words, they must respond to the same command set and supply the same status as their non-local bus cousins.

The fact that the VL bus specification does not define the location or format of the local bus devices' configuration registers opens the door for a "tower of Babel" scenario regarding the software interface to these devices.

---

### Revision 2.0 VL Specification

The rev 2.0 specification adds support for VESA VL bus masters.

---

### PCI Bus Solution

Intel defined the PCI bus to ensure that the marketplace would not become crowded with various permutations of local bus architectures implemented in a short-sighted fashion. The first release of the specification, version 1.0, became available on 6/22/92. Revision 2.0 became available in April of 1993. The current revision of the specification, 2.1, became available in Q1 of 1995. 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

## PCI System Architecture

---

years. In addition, the VL bus has very limited support for burst transfers thereby limiting the achievable throughput.

*PCI stands for Peripheral Component Interconnect.* The PCI bus can be populated with adapters requiring fast accesses to each other and/or system memory and that can be accessed by the host processor at speeds approaching that of the processor's full native bus speed. It is very important to note that all read and write transfers over the PCI bus are burst transfers. The length of the burst is negotiated between the initiator and target devices and may be of any length. *This is in sharp contrast to the burst capability inherent in the VL bus design.* Table 2-2 identifies some of PCI's major design goals. The chapters that follow provide a detailed description of the PCI bus and related subjects.

The PCI specification allows system design centered around two of the three approaches discussed earlier: the buffered and workstation approaches. Due to its performance and flexibility advantages, the workstation approach is preferred. Figure 2-4 illustrates the basic relationship of the PCI, expansion processor and memory buses.