PCI System Architecture

Third Edition

MINDSHARE, INC.

TOM SHANLEY
AND
DON ANDERSON

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

Library of Congress Cataloging-in-Publication Data

ISBN: 0-201-40993-3

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

Part II: Revision 2.1 Essentials

CHAPTER 3: Intro to PCI Bus Operation

CHAPTER 4: Intro to Reflected-Wave Switching

CHAPTER 5: The Functional Signal Groups

CHAPTER 6: PCI Bus Arbitration

CHAPTER 7: The Commands

CHAPTER 8: The Read and Write Transfers

CHAPTER 9: Premature Transaction Termination

CHAPTER 10: Error Detection and Handling

CHAPTER 11: Interrupt-Related Issues

CHAPTER 12: Shared Resource Acquisition

CHAPTER 13: The 64-bit PCI Extension

CHAPTER 14: Add-in Cards and Connectors

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

CHAPTER 15: Intro to Configuration Address Space

CHAPTER 16: Configuration Transactions

CHAPTER 17: Configuration Registers

CHAPTER 18: Expansion ROMs

Part IV: PCI-to-PCI Bridge

CHAPTER 19: PCI-to-PCI Bridge

Part V: The PCI BIOS

CHAPTER 20: The PCI BIOS

Part VI: PCI Cache Support

CHAPTER 21: PCI Cache Support

Part VII: 66MHz PCI Implementation

CHAPTER 22: 66MHz PCI Implementation

Part VIII: Overview of VLSI Technology VL82C59x SuperCore PCI Chipset

CHAPTER 23: Overview of VLSI Technology VL82C59x SuperCore PCI Chipset

Appendices

Figures

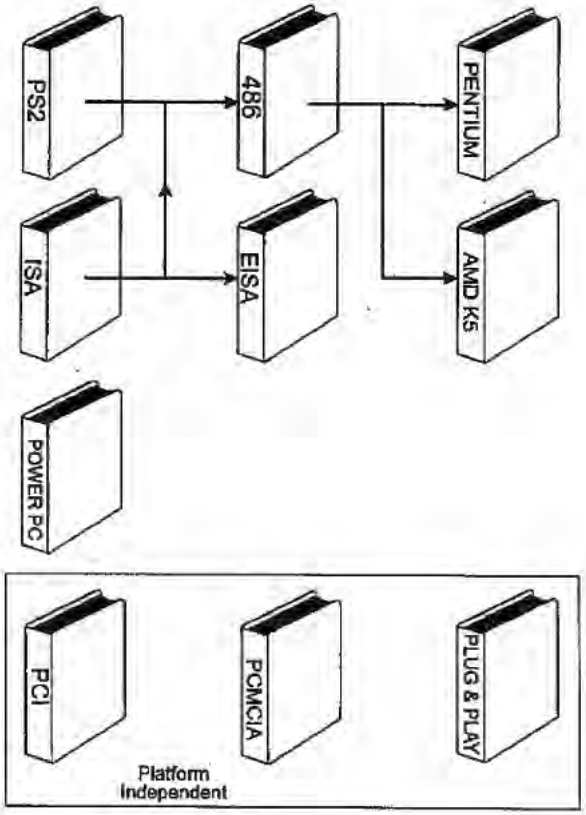
Acknowledgments

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

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

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.

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

To request information on public or private seminars, email your request to: 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

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.

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.

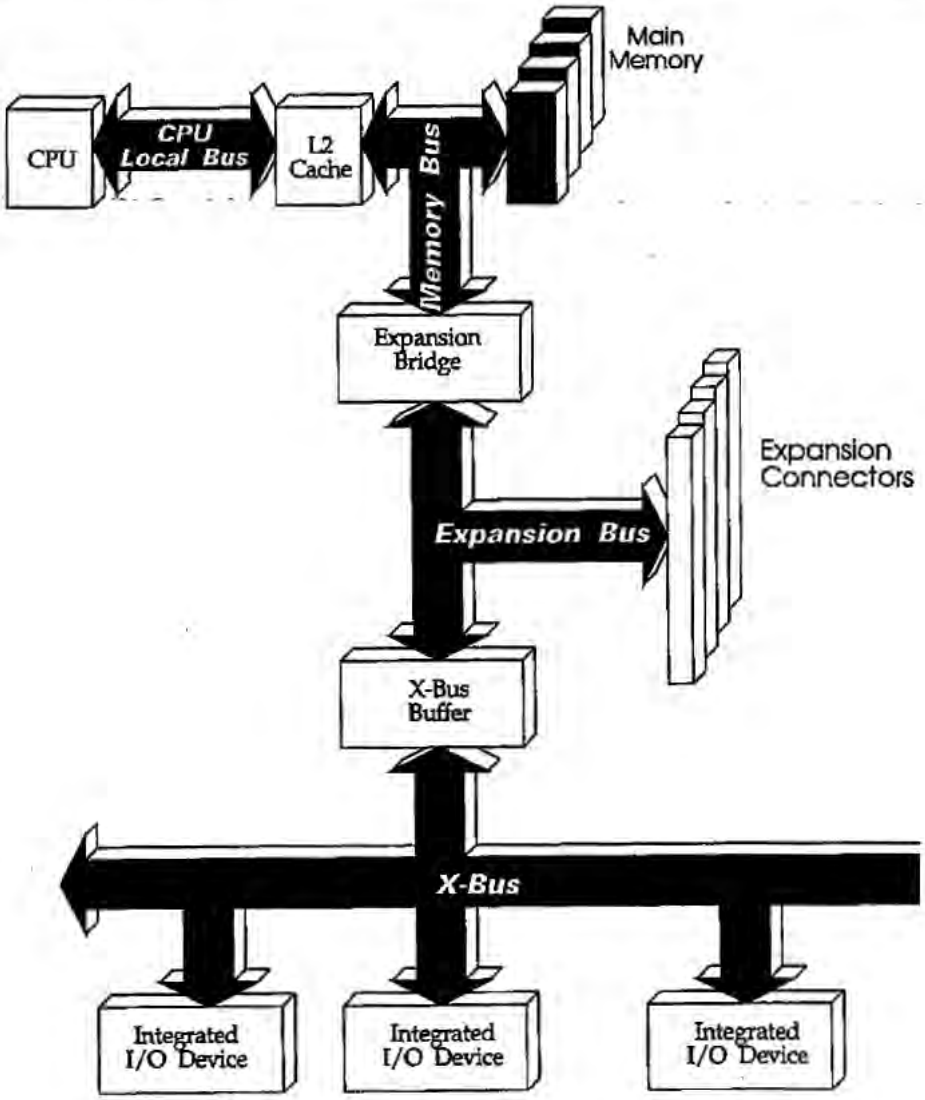


Figure 1-1. The X-Bus

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.

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

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

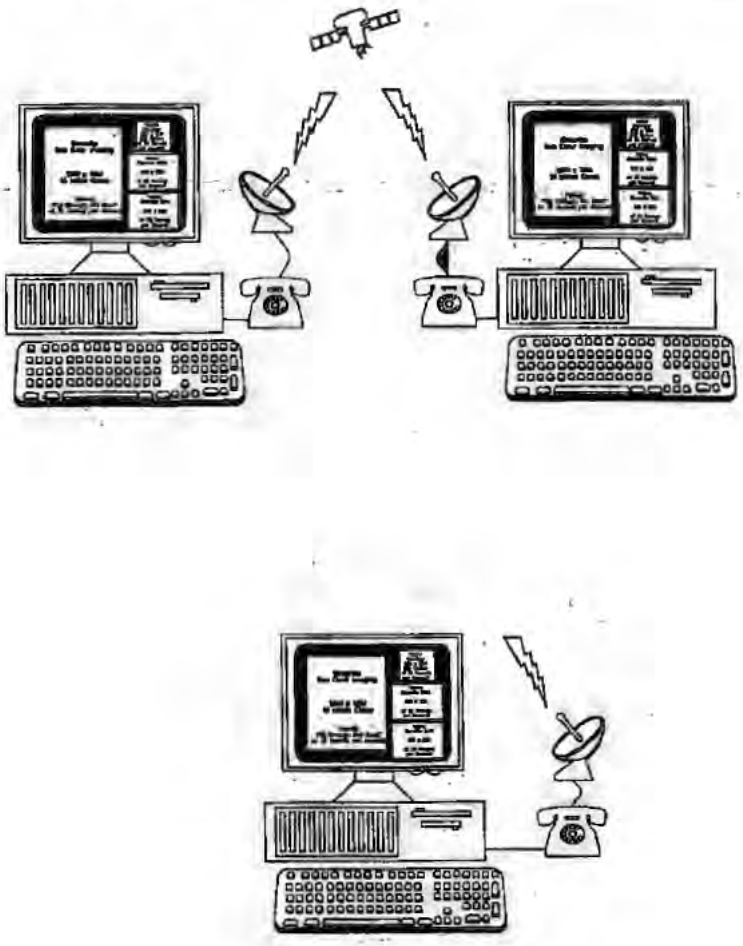


Figure 1-2. The Teleconference

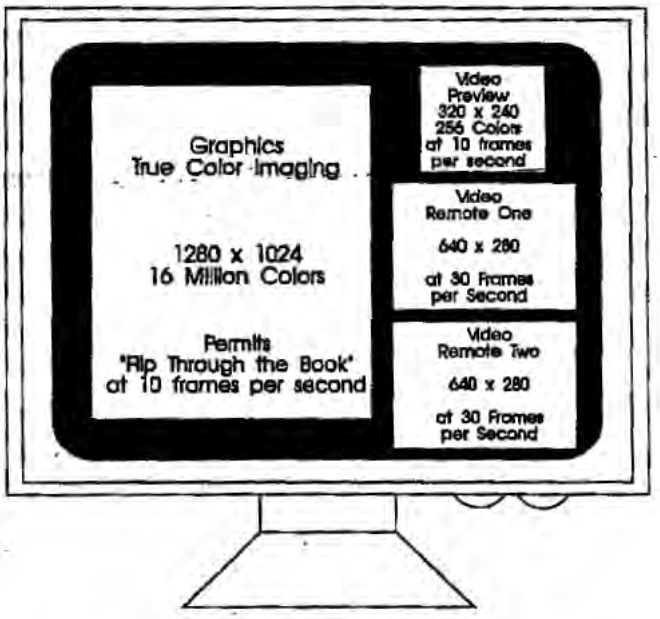


Figure 1-3. The Teleconference Screen Layout

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 offload 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.

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.

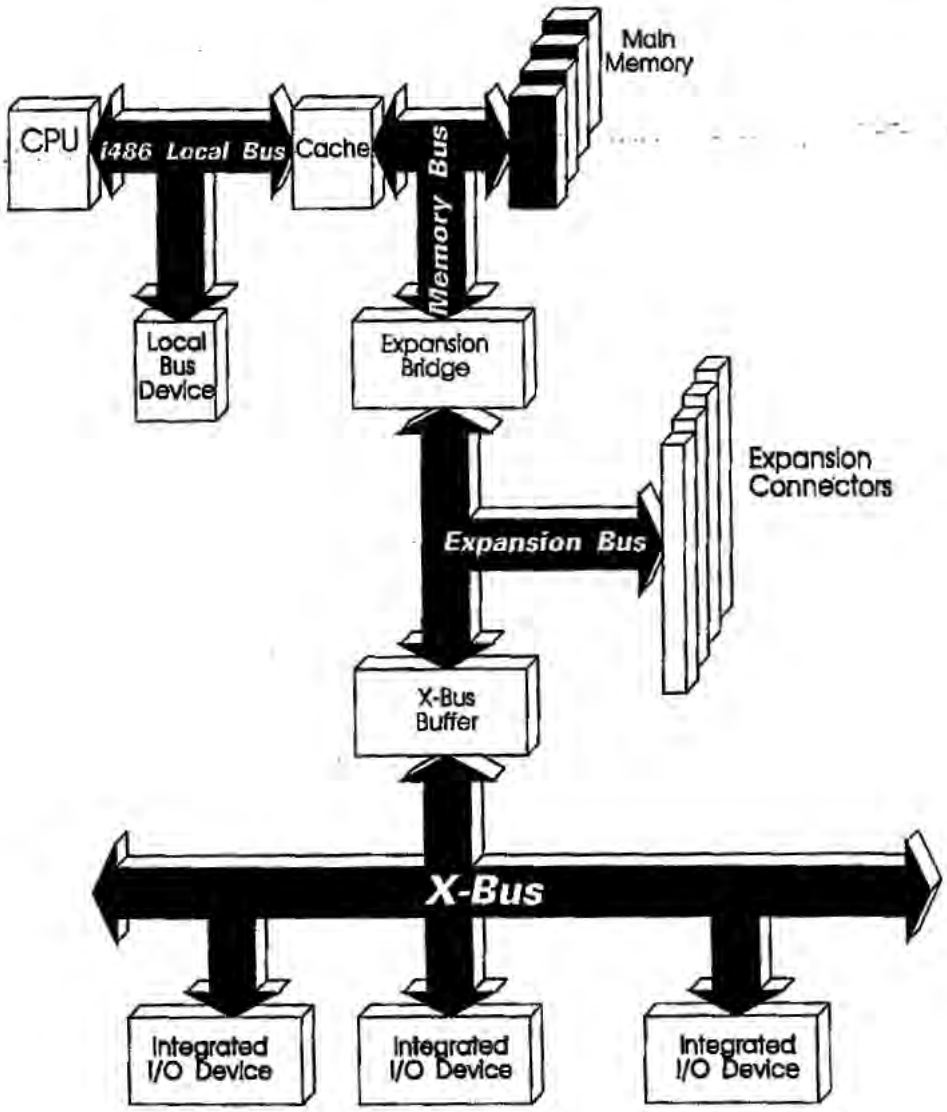


Figure 2-1. The Direct-Connect Local Bus Approach

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.

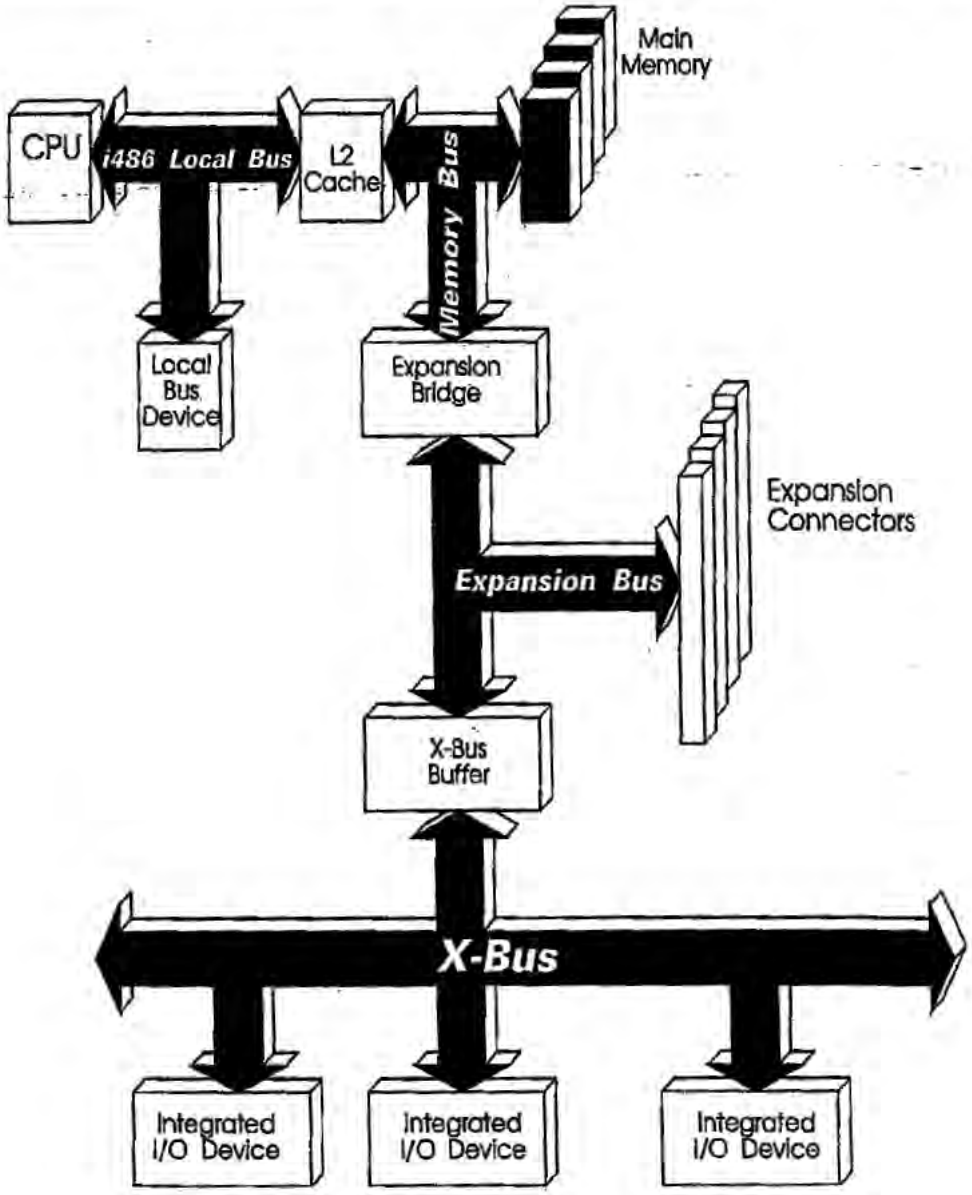


Figure 2-2. The Buffered Local Bus Approach

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).

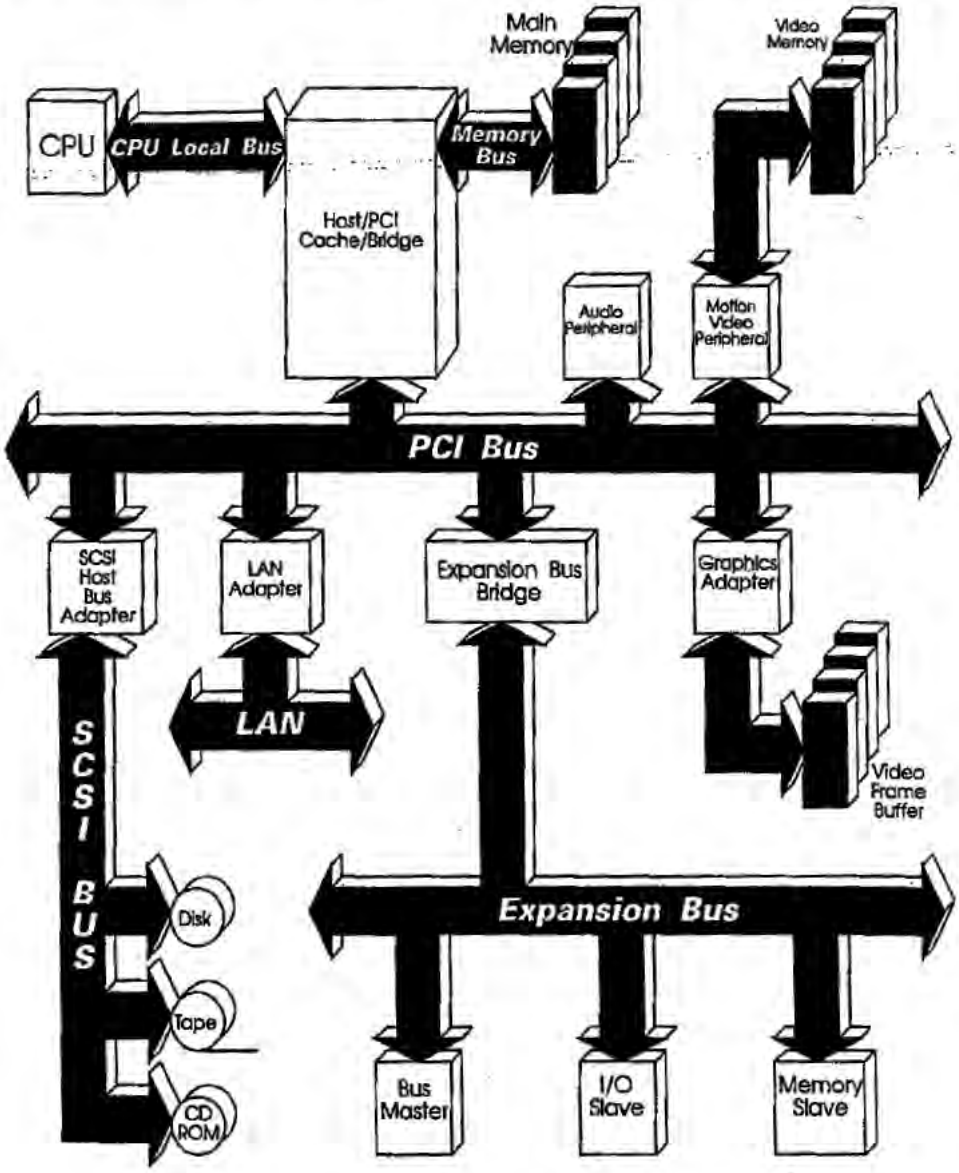


Figure 2-3. The Workstation Approach

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.

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

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

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

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.