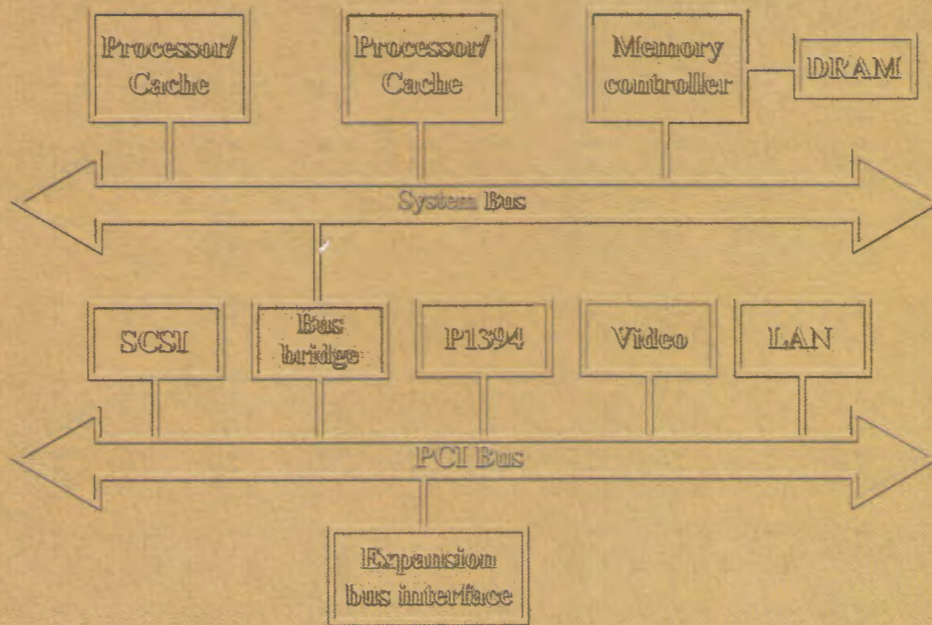


Winner of the 1996 TEXTY
award for the best computer
science & engineering textbook

William Stallings

COMPUTER ORGANIZATION AND ARCHITECTURE

Designing for Performance



Fourth Edition

COMPUTER ORGANIZATION AND ARCHITECTURE

Designing for Performance

FOURTH EDITION

William Stallings



PRENTICE HALL

Upper Saddle River, New Jersey 07458

Library of Congress Cataloging-in-Publication Data

Stallings, William

Computer organization and architecture : designing for performance /

William Stallings. — 4th ed.

p. cm.

Includes bibliographical references and index.

ISBN 0-13-359985-X

1. Computer organization. 2. Computer architecture. I. Title.

QA76.9.C643S73 1996

004.2'2—dc20

96-9115

CIP

Acquisitions editor: Alan Apt

Editorial-production service: Electronic Publishing Services Inc.

Manufacturing buyer: Donna Sullivan

Cover design: Amy Rosen

Cover photograph: Jon Churchman

Production coordinator: Rick DeLorenzo

Editorial assistant: Shirley McGuire



© 1996 by Prentice-Hall, Inc.

Simon & Schuster/A Viacom Company

Upper Saddle River, New Jersey 07458

The author and publisher of this book have used their best efforts in preparing this book. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The author and publisher shall not be liable in any event for incidental or consequential damages in connection with, and arising out of, the furnishing, performance, or use of these programs.

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

Previous edition's copyright © 1993

Printed in the United States of America

10 9 8 7 6

ISBN 0-13-359985-X

Prentice-Hall International (UK) Limited, *London*

Prentice-Hall of Austria Pty. Limited, *Sydney*

Prentice-Hall Canada Inc., *Toronto*

Prentice-Hall Hispanoamericana, S.A., *Mexico*

Prentice-Hall of India Private Limited, *New Delhi*

Prentice-Hall of Japan, Inc., *Tokyo*

Simon & Schuster Asia Pte. Ltd., *Singapore*

Editora Prentice-Hall do Brasil Ltda., *Rio de Janeiro*

system designer of the need to design the basic processor as well as the overall organization.

2. *Advances in processor-to-processor communication schemes.* With hundreds or thousands of processors working cooperatively, efficient and fail-safe methods of coordination and data exchange are vital.
3. *Performance demand.* Even with the stupendous gains made in performance with each successive generation of processor, application performance demands have grown even more rapidly. Parallel processing schemes provide a means of boosting performance using existing processor technology.

This last point is particularly important. Despite the rapidity with which new processor generations are produced, the software demands will always be one step ahead. This is not true just in the supercomputer realm, but on the desktop. Every time computer companies deliver more power in their latest product, invariably new applications come along that utilize the system to the maximum and then some. An Intel 80486 personal computer is clearly overkill for running a word processor under DOS, but only just adequate for running Windows. Pentium systems can handle Windows displays with ease, but they begin to buckle when trying to service multimedia functions (e.g., live-action video or video-phone applications) or advanced technologies such as real-time voice recognition.

The subject of parallel processing is complex, involving issues relating to organization, interconnection structures, inter-processor communication, operating-system design, and application software techniques. The purpose of this section is to provide a brief overview of some of the important organizational concepts.

Types of Parallel Processors

A parallel processor system is one in which multiple processors are interconnected and work cooperatively to execute application programs. The term is meant to refer to systems that support what might be referred to as high-level parallelism. These are systems that offer an explicit framework for developing system software and applications that exploit parallelism. Excluded are systems that employ only low-level parallelism but that are generally still viewed as having a single processor. Examples of low-level parallelism include

- *Instruction Pipelining:* Each instruction execution is divided into a sequence of stages, so that multiple instructions can be executed in parallel, each one in a different stage of execution.
- *Multiple Processor Functional Units:* Replication of ALU units enables a superscalar approach, with multiple instructions executing in parallel, all at the same stage of execution.
- *Separate Specialized Processors:* The most common example is the use of I/O processors, which free the CPU from detailed I/O control responsibilities.

A taxonomy that highlights parallel processor systems first introduced by Flynn [FLYN72] is still the most common way of categorizing such systems. Flynn proposed the following categories of computer systems:

- *Single Instruction, Single Data (SISD):* This is the simplest form of parallel processing, where a single instruction stream controls a single data stream.
- *Single Instruction, Multiple Data (SIMD):* This is a form of parallel processing where a single instruction stream controls multiple data streams.
- *Multiple Instruction, Single Data (MISD):* This is a form of parallel processing where multiple instruction streams control a single data stream.
- *Multiple Instruction, Multiple Data (MIMD):* This is the most general form of parallel processing, where multiple instruction streams control multiple data streams.

With the must be at data transfer processors then each processors as a multi If the pr is a self-co fixed paths as a multi Practical computers Figure 16.1 control uni processing un SIMD, then

Single Inst
Multiple Dat
(SIMD)

FIGURE 16.1

- *Single Instruction Single Data (SISD) stream:* A single processor interprets a single instruction stream to operate on data stored in a single memory. Some parallelism can be achieved using any of the techniques listed above.
- *Single Instruction Multiple Data (SIMD) stream:* A single machine instruction controls the simultaneous execution of a number of processing elements on a lock-step basis. Each processing element has an associated data memory, so that each instruction is executed on a different set of data by the different processors. Vector and array processors fall into this category.
- *Multiple Instruction Single Data (MISD) stream:* A sequence of data is transmitted to a set of processors, each of which executes a different instruction sequence. This structure has never been implemented.
- *Multiple Instruction Multiple Data (MIMD) stream:* A set of processors simultaneously execute different instruction sequences on different data sets.

With the MIMD organization, the processors are general-purpose, because they must be able to process all the instructions necessary to perform the appropriate data transformation. MIMDs can be further subdivided by the means in which the processors communicate (Figure 16.15). If the processors share a common memory, then each processor accesses programs and data stored in the shared memory, and processors communicate with each other via that memory; such a system is known as a **multiprocessor**. Multiprocessors were covered in Sections 16.1 and 16.2.

If the processors each have a dedicated memory, then each processing element is a self-contained computer. Communication among the computers is either via fixed paths or via some message-switching mechanism. Such a system is known as a **multicomputer**.

Practical parallel processing systems include SIMD, multiprocessors, and multicomputers. Figure 16.16 illustrates the general organization. First, as a baseline, Figure 16.16a illustrates the general organization of an SISD. There is some sort of control unit that provides an instruction stream (IS) to a processing unit. The processing unit operates on a single data stream (DS) from a memory unit. With an SIMD, there is still a single control unit, now feeding a single instruction stream to

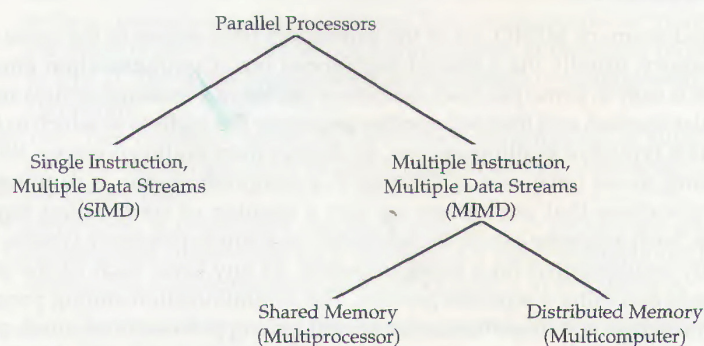


FIGURE 16.15. Practical Parallel Processor Approaches

COMPUTER ORGANIZATION AND ARCHITECTURE: Designing for Performance Fourth Edition WILLIAM STALLINGS

Using the Intel Pentium and PowerPC as running examples, Stallings relates contemporary design issues to the fundamentals of computer organization and architecture. The author's highly accessible presentation and readable writing style make this an eminently usable book.

The challenge of designing a system with balanced, optimized performance grows with the rapid improvement of computer system performance parameters. This book meets the challenge through careful examination of important issues such as the design of innovative bus interconnection structures, cache memory organization and protocols, instruction set architecture, and I/O architecture.

Stallings provides currency through the following **Key Features**:

- Examples of two recent system bus specifications (PCI and Futurebus+).
- Expanded coverage of cache memory material, including separate instruction and data caches as well as the use of two levels of caches.
- Recent developments in DRAM organization.
- Discussion on RAID technology.
- Detailed examination of SCSI parallel and P1394 serial bus specifications.
- Cache coherency in multiple-processor systems and MESI protocol are explored.
- Parallel organizations including many microprocessors are discussed.
- Big-Endian, Little-Endian, and the new Bi-Endian approach are presented.

About the Author

William Stallings has made a unique contribution to understanding the technical developments in computer networking and architecture. He has authored 15 titles on these subjects, including *Data and Computer Communications, Fourth Edition* (Prentice Hall, 1994) which has become the standard in the field.

Dr. Stallings holds a Ph.D. in computer science from MIT and a B.S. in electrical engineering from Notre Dame. Over the last 20 years he has been a technical contributor, technical manager, and an executive with several high-technology firms. Currently Dr. Stallings is an independent consultant with computer and networking manufacturers, software development firms, and government research institutions. In addition, he is a frequent lecturer and author of numerous technical papers.

ISBN 0-13-359985-X



9 780133 599855

PRENTICE HALL Upper Saddle River, NJ 07458