GLOBAL EDITION

Computer Organization and Architecture

Designing for Performance

TENTH EDITION

William Stallings

ALWAYS LEARNING

PEARSON

digital resources for students

Your new textbook provides 12-month access to digital resources that may include VideoNotes (step-by-step video tutorials on programming concepts), source code, web chapters, quizzes, and more. Refer to the preface in the textbook for a detailed list of resources.

Follow the instructions below to register for the Companion Website for Stallings' Computer Organization and Architecture, Tenth Edition.

1. Go to www.pearsonglobaleditions.com/stallings
2. Click Companion Website
3. Click Register and follow the on-screen instructions to create a login name and password.

Use a coin to scratch off the coating and reveal your access code.
Do not use a sharp knife or other sharp object as it may damage the code.

Use the login name and password you created during registration to start using the digital resources that accompany your textbook.

IMPORTANT:

This access code can only be used once. This subscription is valid for 12 months upon activation and is not transferable. If the access code has already been revealed it may no longer be valid. If this is the case you can purchase a subscription on the login page for the Companion Website.

For technical support go to http://247pearsoned.custhelp.com

This page intentionally left blank.

Computer Organization and Architecture

Designing for Performance

Tenth Edition

Global Edition

This page intentionally left blank.

Computer Organization and Architecture
Designing for Performance
Tenth Edition
Global Edition

William Stallings

With contribution by
Peter Zeno
University of Bridgeport

With Foreword by
Chris Jesshope
Professor (emeritus) University of Amsterdam

PEARSON

Boston • Columbus • Hoboken • Indianapolis • New York • San Francisco
Amsterdam • Cape Town • Dubai • London • Madrid • Milan • Munich • Paris • Montreal
Toronto • Delhi • Mexico City • São Paulo • Sydney • Hong Kong • Seoul • Singapore • Taipei • Tokyo

Vice President and Editorial Director, ECS: *Marcia J. Horton*
Executive Editor: *Tracy Johnson (Dunkelberger)*
Editorial Assistant: *Kelsey Loanes*
Acquisitions Editor, Global Editions: *Karthik Subramanian*
Program Manager: *Carole Snyder*
Director of Product Management: *Erin Gregg*
Team Lead Product Management: *Scott Disanno*
Project Manager: *Robert Engelhardt*
Project Editor, Global Editions: *K.K. Neelakantan*
Senior Production Manufacturing Controller, Global Editions: *Trudy Kimber*
Media Team Lead: *Steve Wright*
R&P Manager: *Rachel Youdelman*
R&P Senior Project Manager: *Timothy Nicholls*
Procurement Manager: *Mary Fischer*
Senior Specialist, Program Planning and Support: *Maura Zaldivar-Garcia*
Inventory Manager: *Bruce Boundy*
VP of Marketing: *Christy Lesko*
Director of Field Marketing: *Demetrius Hall*
Product Marketing Manager: *Bram van Kempen*
Media Production Manager, Global Editions: *Vikram Kumar*
Marketing Assistant: *Jon Bryant*
Cover Designer: *Lumina Datamatics*
Cover Art: *phipatbig / Shutterstock*
Full-Service Project Management: *Mahalatchoumy Saravanan, Jouve India*

Pearson Education Limited
Edinburgh Gate
Harlow
Essex CM20 2JE
England

and Associated Companies throughout the world

Visit us on the World Wide Web at:
www.pearsonglobaleditions.com

Authorized adaptation from the United States edition, entitled Computer Organization and Architecture: Designing for Performance, 10th edition, ISBN 978-0-13-410161-3, by William Stallings published by Pearson Education © 2016.

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library

10 9 8 7 6 5 4 3 2 1

ISBN 10: 1-292-09685-3

ISBN 13: 978-1-292-09685-8

Typeset in Times Ten LT Std 10/12 by Jouve India

Printed and bound by Courier Westford in The United States of America.

To Tricia
my loving wife, the kindest
and gentlest person

This page intentionally left blank.

CONTENTS

[1]Online chapters, appendices, and other documents are Premium Content, available via the access card at the front of this book.

This page intentionally left blank.

FOREWORD

by Chris Jesshope
Professor (emeritus) University of Amsterdam
Author of Parallel Computers (with R W Hockney), 1981 & 1988

Having been active in computer organization and architecture for many years, it is a pleasure to write this foreword for the new edition of William Stallings' comprehensive book on this subject. In doing this, I found myself reflecting on the trends and changes in this subject over the time that I have been involved in it. I myself became interested in computer architecture at a time of significant innovation and disruption. That disruption was brought about not only through advances in technology but perhaps more significantly through access to that technology. VLSI was here and VLSI design was available to students in the classroom. These were exciting times. The ability to integrate a mainframe style computer on a single silicon chip was a milestone, but that this was accomplished by an academic research team made the achievement quite unique. This period was characterized by innovation and diversity in computer architecture with one of the main trends being in the area of parallelism. In the 1970s, I had hands-on experience of the Illiac IV, which was an early example of explicit parallelism in computer architecture and which incidentally pioneered all semiconductor memory. This interaction, and it certainly was that, kick-started my own interest in computer architecture and organization, with particular emphasis on explicit parallelism in computer architecture.

Throughout the 1980s and early 1990s research flourished in this field and there was a great deal of innovation, much of which came to market through university start-ups. Ironically however, it was the same technology that reversed this trend. Diversity was gradually replaced with a near monoculture in computer systems with advances in just a few instruction set architectures. Moore's law, a self-fulfilling prediction that became an industry guideline, meant that basic device speeds and integration densities both grew exponentially, with the latter doubling every 18 months of so. The speed increase was the proverbial free lunch for computer architects and the integration levels allowed more complexity and innovation at the micro-architecture level. The free lunch of course did have a cost, that being the exponential growth of capital investment required to fulfill Moore's law, which once again limited the access to state-of-the-art technologies. Moreover, most users found it easier to wait for the next generation of mainstream processor than to invest in the innovations in parallel computers, with their pitfalls and difficulties. The exceptions to this were the few large institutions requiring ultimate performance; two topical examples being large-scale scientific simulation such as climate modeling and also in our security services for code breaking. For

everyone else, the name of the game was compatibility and two instruction set architectures that benefited from this were x86 and ARM, the latter in embedded systems and the former in just about everything else. Parallelism was still there in the implementation of these ISAs, it was just that it was implicit, harnessed by the architecture not in the instruction stream that drives it.

Throughout the late 1990s and early 2000s, this approach to implicitly exploiting concurrency in single-core computer systems flourished. However, in spite of the exponential growth of logic density, it was the cost of the techniques exploited which brought this era to a close. In superscalar processors, the logic costs do not grow linearly with issue width (parallelism), while some components grow as the square or even the cube of the issue width. Although the exponential growth in logic could sustain this continued development, there were two major pitfalls: it was increasingly difficult to expose concurrency implicitly from imperative programs and hence efficiencies in the use of instruction issue slots decreased. Perhaps more importantly, technology was experiencing a new barrier to performance gains, namely that of power dissipation, and several superscalar developments were halted because the silicon in them would have been too hot. These constraints have mandated the exploitation of explicit parallelism, despite the compatibility challenges. So it seems that again innovation and diversity are opening up this area to new research.

Perhaps not since the 1980s has it been so interesting to study in this field. That diversity is an economic reality can be seen by the decrease in issue width (implicit parallelism) and increase in the number of cores (explicit parallelism) in mainstream processors. However, the question is how to exploit this, both at the application and the system level. There are significant challenges here still to be solved. Superscalar processors rely on the processor to extract parallelism from a single instruction stream. What if we shifted the emphasis and provided an instruction stream with maximum parallelism, how can we exploit this in different configurations and/or generations of processors that require different levels of explicit parallelism? Is it possible therefore to have a micro-architecture that sequentializes and schedules this maximum concurrency captured in the ISA to match the current configuration of cores so that we gain the same compatibility in a world of explicit parallelism? Does this require operating systems in silicon for efficiency?

These are just some of the questions facing us today. To answer these questions and more requires a sound foundation in computer organization and architecture, and this book by William Stallings provides a very timely and comprehensive foundation. It gives a complete introduction to the basics required, tackling what can be quite complex topics with apparent simplicity. Moreover, it deals with the more recent developments in this field, where innovation has in the past, and is, currently taking place. Examples are in superscalar issue and in explicitly parallel multicores. What is more, this latest edition includes two very recent topics in the design and use of GPUs for general-purpose use and the latest trends in cloud computing, both of which have become mainstream only recently. The book makes good use of examples throughout to highlight the theoretical issues covered, and most of these examples are drawn from developments in the two most widely used ISAs, namely the x86 and ARM. To reiterate, this book is complete and is a pleasure to read and hopefully will kick-start more young researchers down the same path that I have enjoyed over the last 40 years!

PREFACE

WHAT'S NEW IN THE TENTH EDITION

Since the ninth edition of this book was published, the field has seen continued innovations and improvements. In this new edition, I try to capture these changes while maintaining a broad and comprehensive coverage of the entire field. To begin this process of revision, the ninth edition of this book was extensively reviewed by a number of professors who teach the subject and by professionals working in the field. The result is that, in many places, the narrative has been clarified and tightened, and illustrations have been improved.

Beyond these refinements to improve pedagogy and user-friendliness, there have been substantive changes throughout the book. Roughly the same chapter organization has been retained, but much of the material has been revised and new material has been added. The most noteworthy changes are as follows:

- **GPGPU [General-Purpose Computing on Graphics Processing Units (GPUs)]:** One of the most important new developments in recent years has been the broad adoption of GPGPUs to work in coordination with traditional CPUs to handle a wide range of applications involving large arrays of data. A new chapter is devoted to the topic of GPGPUs.
- **Heterogeneous multicore processors:** The latest development in multicore architecture is the heterogeneous multicore processor. A new section in the chapter on multicore processors surveys the various types of heterogeneous multicore processors.
- **Embedded systems:** The overview of embedded systems in Chapter 1 has been substantially revised and expanded to reflect the current state of embedded technology.
- **Microcontrollers:** In terms of numbers, almost all computers now in use are embedded microcontrollers. The treatment of embedded systems in Chapter 1 now includes coverage of microcontrollers. The ARM Cortex-M3 microcontroller is used as an example system throughout the text.
- **Cloud computing:** New to this edition is a discussion of cloud computing, with an overview in Chapter 1 and more detailed treatment in Chapter 17.
- **System performance:** The coverage of system performance issues has been revised, expanded, and reorganized for a clearer and more thorough treatment. Chapter 2 is devoted to this topic, and the issue of system performance arises throughout the book.

- **Flash memory:** The coverage of flash memory has been updated and expanded, and now includes a discussion of the technology and organization of flash memory for internal memory (Chapter 5) and external memory (Chapter 6).
- **Nonvolatile RAM:** New to this edition is treatment of three important new nonvolatile solid-state RAM technologies that occupy different positions in the memory hierarchy: STT-RAM, PCRAM, and ReRAM.
- **Direct cache access (DCA):** To meet the protocol processing demands for very high speed network connections, Intel and other manufacturers have developed DCA technologies that provide much greater throughput than traditional direct memory access (DMA) approaches. New to this edition, Chapter 7 explores DCA in some detail.
- **Intel Core Microarchitecture:** As in the previous edition, the Intel x86 family is used as a major example system throughout. The treatment has been updated to reflect newer Intel systems, especially the Intel Core Microarchitecture, which is used on both PC and server products.
- **Homework problems:** The number of supplemental homework problems, with solutions, available for student practice has been expanded.

SUPPORT OF ACM/IEEE COMPUTER SCIENCE CURRICULA 2013

The book is intended for both an academic and a professional audience. As a textbook, it is intended as a one- or two-semester undergraduate course for computer science, computer engineering, and electrical engineering majors. This edition is designed to support the recommendations of the ACM/IEEE Computer Science Curricula 2013 (CS2013). CS2013 divides all course work into three categories: Core-Tier 1 (all topics should be included in the curriculum); Core-Tier-2 (all or almost all topics should be included); and Elective (desirable to provide breadth and depth). In the Architecture and Organization (AR) area, CS2013 includes five Tier-2 topics and three Elective topics, each of which has a number of subtopics. This text covers all eight topics listed by CS2013. Table P.1 shows the support for the AR Knowledge Area provided in this textbook.

Table P.1 Coverage of CS2013 Architecture and Organization (AR) Knowledge Area

IAS Knowledge Units	Topics	Textbook Coverage
Digital Logic and Digital Systems (Tier 2)	• Overview and history of computer architecture • Combinational vs. sequential logic/Field programmable gate arrays as a fundamental combinational sequential logic building block • Multiple representations/layers of interpretation (hardware is just another layer) • Physical constraints (gate delays, fan-in, fan-out, energy/power)	—Chapter 1 —Chapter 11
Machine Level Representation of Data (Tier 2)	• Bits, bytes, and words • Numeric data representation and number bases • Fixed- and floating-point systems • Signed and twos-complement representations • Representation of non-numeric data (character codes, graphical data)	—Chapter 9 —Chapter 10

IAS Knowledge Units	Topics	Textbook Coverage
Assembly Level Machine Organization (Tier 2)	• Basic organization of the von Neumann machine • Control unit; instruction fetch, decode, and execution • Instruction sets and types (data manipulation, control, I/O) • Assembly/machine language programming • Instruction formats • Addressing modes • Subroutine call and return mechanisms (cross-reference PL/Language Translation and Execution) • I/O and interrupts • Shared memory multiprocessors/multicore organization • Introduction to SIMD vs. MIMD and the Flynn Taxonomy	—Chapter 1 —Chapter 7 —Chapter 12 —Chapter 13 —Chapter 17 —Chapter 18 —Chapter 20 —Chapter 21 —Appendix A
Memory System Organization and Architecture (Tier 2)	• Storage systems and their technology • Memory hierarchy: temporal and spatial locality • Main memory organization and operations • Latency, cycle time, bandwidth, and interleaving • Cache memories (address mapping, block size, replacement and store policy) • Multiprocessor cache consistency/Using the memory system for inter-core synchronization/atomic memory operations • Virtual memory (page table, TLB) • Fault handling and reliability	—Chapter 4 —Chapter 5 —Chapter 6 —Chapter 8 —Chapter 17
Interfacing and Communication (Tier 2)	• I/O fundamentals: handshaking, buffering, programmed I/O, interrupt-driven I/O • Interrupt structures: vectored and prioritized, interrupt acknowledgment • External storage, physical organization, and drives • Buses: bus protocols, arbitration, direct-memory access (DMA) • RAID architectures	—Chapter 3 —Chapter 6 —Chapter 7
Functional Organization (Elective)	• Implementation of simple datapaths, including instruction pipelining, hazard detection, and resolution • Control unit: hardwired realization vs. microprogrammed realization • Instruction pipelining • Introduction to instruction-level parallelism (ILP)	—Chapter 14 —Chapter 16 —Chapter 20 —Chapter 21
Multiprocessing and Alternative Architectures (Elective)	• Example SIMD and MIMD instruction sets and architectures • Interconnection networks • Shared multiprocessor memory systems and memory consistency • Multiprocessor cache coherence	—Chapter 12 —Chapter 13 —Chapter 17
Performance Enhancements (Elective)	• Superscalar architecture • Branch prediction, Speculative execution, Out-of-order execution • Prefetching • Vector processors and GPUs • Hardware support for multithreading • Scalability	—Chapter 15 —Chapter 16 —Chapter 19

OBJECTIVES

This book is about the structure and function of computers. Its purpose is to present, as clearly and completely as possible, the nature and characteristics of modern-day computer systems.

This task is challenging for several reasons. First, there is a tremendous variety of products that can rightly claim the name of computer, from single-chip microprocessors costing a few dollars to supercomputers costing tens of millions of dollars. Variety is exhibited not only in cost but also in size, performance, and application. Second, the rapid pace of change that has always characterized computer technology continues with no letup. These changes cover all aspects of computer technology, from the underlying integrated circuit technology used to construct computer components to the increasing use of parallel organization concepts in combining those components.

In spite of the variety and pace of change in the computer field, certain fundamental concepts apply consistently throughout. The application of these concepts depends on the current state of the technology and the price/performance objectives of the designer. The intent of this book is to provide a thorough discussion of the fundamentals of computer organization and architecture and to relate these to contemporary design issues.

The subtitle suggests the theme and the approach taken in this book. It has always been important to design computer systems to achieve high performance, but never has this requirement been stronger or more difficult to satisfy than today. All of the basic performance characteristics of computer systems, including processor speed, memory speed, memory capacity, and interconnection data rates, are increasing rapidly. Moreover, they are increasing at different rates. This makes it difficult to design a balanced system that maximizes the performance and utilization of all elements. Thus, computer design increasingly becomes a game of changing the structure or function in one area to compensate for a performance mismatch in another area. We will see this game played out in numerous design decisions throughout the book.

A computer system, like any system, consists of an interrelated set of components. The system is best characterized in terms of structure—the way in which components are interconnected, and function—the operation of the individual components. Furthermore, a computer's organization is hierarchical. Each major component can be further described by decomposing it into its major subcomponents and describing their structure and function. For clarity and ease of understanding, this hierarchical organization is described in this book from the top down:

- **Computer system:** Major components are processor, memory, I/O.
- **Processor:** Major components are control unit, registers, ALU, and instruction execution unit.
- **Control unit:** Provides control signals for the operation and coordination of all processor components. Traditionally, a microprogramming implementation has been used, in which major components are control memory, microinstruction sequencing logic, and registers. More recently, microprogramming has been less prominent but remains an important implementation technique.

The objective is to present the material in a fashion that keeps new material in a clear context. This should minimize the chance that the reader will get lost and should provide better motivation than a bottom-up approach.

Throughout the discussion, aspects of the system are viewed from the points of view of both architecture (those attributes of a system visible to a machine language programmer) and organization (the operational units and their interconnections that realize the architecture).

EXAMPLE SYSTEMS

This text is intended to acquaint the reader with the design principles and implementation issues of contemporary operating systems. Accordingly, a purely conceptual or theoretical treatment would be inadequate. To illustrate the concepts and to tie them to real-world design choices that must be made, two processor families have been chosen as running examples:

- **Intel x86 architecture:** The x86 architecture is the most widely used for nonembedded computer systems. The x86 is essentially a complex instruction set computer (CISC) with some RISC features. Recent members of the x86 family make use of superscalar and multicore design principles. The evolution of features in the x86 architecture provides a unique case-study of the evolution of most of the design principles in computer architecture.
- **ARM:** The ARM architecture is arguably the most widely used embedded processor, used in cell phones, iPods, remote sensor equipment, and many other devices. The ARM is essentially a reduced instruction set computer (RISC). Recent members of the ARM family make use of superscalar and multicore design principles.

Many, but by no means all, of the examples in this book are drawn from these two computer families. Numerous other systems, both contemporary and historical, provide examples of important computer architecture design features.

PLAN OF THE TEXT

The book is organized into six parts:

- Overview
- The computer system
- Arithmetic and logic
- The central processing unit
- Parallel organization, including multicore
- The control unit

The book includes a number of pedagogic features, including the use of interactive simulations and numerous figures and tables to clarify the discussion. Each chapter includes a list of key words, review questions, homework problems, and suggestions for further reading. The book also includes an extensive glossary, a list of frequently used acronyms, and a bibliography.

INSTRUCTOR SUPPORT MATERIALS

Support materials for instructors are available at the **Instructor Resource Center (IRC)** for this textbook, which can be reached through the publisher's Web site www.pearsonglobaleditions.com/stallings or by clicking on the link labeled "Pearson Resources for Instructors" at this

book's Companion Web site at www.pearsonglobaleditions.com/stallings. To gain access to the IRC, please contact your local Pearson sales representative. The IRC provides the following materials:

- **Projects manual:** Project resources including documents and portable software, plus suggested project assignments for all of the project categories listed subsequently in this Preface.
- **Solutions manual:** Solutions to end-of-chapter Review Questions and Problems.
- **PowerPoint slides:** A set of slides covering all chapters, suitable for use in lecturing.
- **PDF files:** Copies of all figures and tables from the book.
- **Test bank:** A chapter-by-chapter set of questions.
- **Sample syllabuses:** The text contains more material than can be conveniently covered in one semester. Accordingly, instructors are provided with several sample syllabuses that guide the use of the text within limited time. These samples are based on real-world experience by professors with the first edition.

The **Companion Web site**, at www.pearsonglobaleditions.com/stallings (click on Instructor Resources link) includes the following:

- Links to Web sites for other courses being taught using this book.
- Sign-up information for an Internet mailing list for instructors using this book to exchange information, suggestions, and questions with each other and with the author.

STUDENT RESOURCES

For this new edition, a tremendous amount of original supporting material for students has been made available online, at two Web locations. The **Companion Web Site**, at www.pearsonglobaleditions.com/stallings (click on Student Resources link), includes a list of relevant links organized by chapter and an errata sheet for the book.

Purchasing this textbook new grants the reader six months of access to the **Premium Content Site**, which includes the following materials:

- **Online chapters:** To limit the size and cost of the book, two chapters of the book are provided in PDF format. The chapters are listed in this book's table of contents.
- **Online appendices:** There are numerous interesting topics that support material found in the text but whose inclusion is not warranted in the printed text. A total of 13 appendices cover these topics for the interested student. The appendices are listed in this book's table of contents.
- **Homework problems and solutions:** To aid the student in understanding the material, a separate set of homework problems with solutions are available. Students can enhance their understanding of the material by working out the solutions to these problems and then checking their answers.