To Tricia
my loving wife, the kindest
and gentlest person
CONTENTS

Preface xiii
About the Author xxii

PART ONE  INTRODUCTION 1

Chapter 1   Basic Concepts and Computer Evolution 1
  1.1 Organization and Architecture 2
  1.2 Structure and Function 3
  1.3 The IAS Computer 11
  1.4 Gates, Memory Cells, Chips, and Multichip Modules 17
  1.5 The Evolution of the Intel x86 Architecture 23
  1.6 Embedded Systems 24
  1.7 ARM Architecture 29
  1.8 Key Terms, Review Questions, and Problems 34

Chapter 2   Performance Concepts 37
  2.1 Designing for Performance 38
  2.2 Multicore, MICs, and GPGPUs 44
  2.3 Two Laws that Provide Insight: Amdahl’s Law and Little’s Law 45
  2.4 Basic Measures of Computer Performance 48
  2.5 Calculating the Mean 51
  2.6 Benchmarks and SPEC 59
  2.7 Key Terms, Review Questions, and Problems 66

PART TWO  THE COMPUTER SYSTEM 72

Chapter 3  A Top-Level View of Computer Function and Interconnection 72
  3.1 Computer Components 73
  3.2 Computer Function 75
  3.3 Interconnection Structures 90
  3.4 Bus Interconnection 92
  3.5 Point-to-Point Interconnect 94
  3.6 PCI Express 99
  3.7 Key Terms, Review Questions, and Problems 107

Chapter 4  The Memory Hierarchy: Locality and Performance 112
  4.1 Principle of Locality 113
  4.2 Characteristics of Memory Systems 118
  4.3 The Memory Hierarchy 121
  4.4 Performance Modeling of a Multilevel Memory Hierarchy 128
  4.5 Key Terms, Review Questions, and Problems 135
Chapter 5 Cache Memory 138

5.1 Cache Memory Principles 139
5.2 Elements of Cache Design 143
5.3 Intel x86 Cache Organization 165
5.4 The IBM z13 Cache Organization 168
5.5 Cache Performance Models 169
5.6 Key Terms, Review Questions, and Problems 173

Chapter 6 Internal Memory 177

6.1 Semiconductor Main Memory 178
6.2 Error Correction 187
6.3 DDR DRAM 192
6.4 eDRAM 197
6.5 Flash Memory 199
6.6 Newer Nonvolatile Solid-State Memory Technologies 202
6.7 Key Terms, Review Questions, and Problems 205

Chapter 7 External Memory 210

7.1 Magnetic Disk 211
7.2 RAID 221
7.3 Solid State Drives 231
7.4 Optical Memory 234
7.5 Magnetic Tape 240
7.6 Key Terms, Review Questions, and Problems 242

Chapter 8 Input/Output 245

8.1 External Devices 247
8.2 I/O Modules 249
8.3 Programmed I/O 252
8.4 Interrupt-Driven I/O 256
8.5 Direct Memory Access 265
8.6 Direct Cache Access 271
8.7 I/O Channels and Processors 278
8.8 External Interconnection Standards 280
8.9 IBM z13 I/O Structure 283
8.10 Key Terms, Review Questions, and Problems 287

Chapter 9 Operating System Support 291

9.1 Operating System Overview 292
9.2 Scheduling 303
9.3 Memory Management 309
9.4 Intel x86 Memory Management 320
9.5 ARM Memory Management 325
9.6 Key Terms, Review Questions, and Problems 330

PART THREE ARITHMETIC AND LOGIC 334

Chapter 10 Number Systems 334

10.1 The Decimal System 335
10.2 Positional Number Systems 336
10.3 The Binary System 337
10.4 Converting Between Binary and Decimal 337
10.5 Hexadecimal Notation 340
10.6 Key Terms and Problems 342

Chapter 11 Computer Arithmetic 344
11.1 The Arithmetic and Logic Unit 345
11.2 Integer Representation 346
11.3 Integer Arithmetic 351
11.4 Floating-Point Representation 366
11.5 Floating-Point Arithmetic 374
11.6 Key Terms, Review Questions, and Problems 383

Chapter 12 Digital Logic 388
12.1 Boolean Algebra 389
12.2 Gates 394
12.3 Combinational Circuits 396
12.4 Sequential Circuits 414
12.5 Programmable Logic Devices 423
12.6 Key Terms and Problems 428

PART FOUR INSTRUCTION SETS AND ASSEMBLY LANGUAGE 432

Chapter 13 Instruction Sets: Characteristics and Functions 432
13.1 Machine Instruction Characteristics 433
13.2 Types of Operands 440
13.3 Intel x86 and ARM Data Types 442
13.4 Types of Operations 445
13.5 Intel x86 and ARM Operation Types 458
13.6 Key Terms, Review Questions, and Problems 466
Appendix 13A Little-, Big-, and Bi-Endian 472

Chapter 14 Instruction Sets: Addressing Modes and Formats 476
14.1 Addressing Modes 477
14.2 x86 and ARM Addressing Modes 483
14.3 Instruction Formats 489
14.4 x86 and ARM Instruction Formats 497
14.5 Key Terms, Review Questions, and Problems 502

Chapter 15 Assembly Language and Related Topics 506
15.1 Assembly Language Concepts 507
15.2 Motivation for Assembly Language Programming 510
15.3 Assembly Language Elements 512
15.4 Examples 518
15.5 Types of Assemblers 523
15.6 Assemblers 523
15.7 Loading and Linking 526
15.8 Key Terms, Review Questions, and Problems 533
PART FIVE  THE CENTRAL PROCESSING UNIT 537

Chapter 16  Processor Structure and Function 537

16.1  Processor Organization 538
16.2  Register Organization 539
16.3  Instruction Cycle 545
16.4  Instruction Pipelining 548
16.5  Processor Organization for Pipelining 566
16.6  The x86 Processor Family 568
16.7  The ARM Processor 575
16.8  Key Terms, Review Questions, and Problems 581

Chapter 17  Reduced Instruction Set Computers 586

17.1  Instruction Execution Characteristics 588
17.2  The Use of a Large Register File 593
17.3  Compiler-Based Register Optimization 598
17.4  Reduced Instruction Set Architecture 600
17.5  RISC Pipelining 606
17.6  MIPS R4000 610
17.7  SPARC 616
17.8  Processor Organization for Pipelining 621
17.9  CISC, RISC, and Contemporary Systems 623
17.10  Key Terms, Review Questions, and Problems 625

Chapter 18  Instruction-Level Parallelism and Superscalar Processors 629

18.1  Overview 630
18.2  Design Issues 637
18.3  Intel Core Microarchitecture 646
18.4  ARM Cortex-A8 652
18.5  ARM Cortex-M3 658
18.6  Key Terms, Review Questions, and Problems 663

Chapter 19  Control Unit Operation and Microprogrammed Control 669

19.1  Micro-operations 670
19.2  Control of the Processor 676
19.3  Hardwired Implementation 686
19.4  Microprogrammed Control 689
19.5  Key Terms, Review Questions, and Problems 698

PART SIX  PARALLEL ORGANIZATION 701

Chapter 20  Parallel Processing 701

20.1  Multiple Processors Organization 703
20.2  Symmetric Multiprocessors 705
20.3  Cache Coherence and the MESI Protocol 709
20.4  Multithreading and Chip Multiprocessors 718
20.5  Clusters 723
20.6  Nonuniform Memory Access 726
20.7  Key Terms, Review Questions, and Problems 730
Chapter 21 Multicore Computers 736

21.1 Hardware Performance Issues 737
21.2 Software Performance Issues 740
21.3 Multicore Organization 745
21.4 Heterogeneous Multicore Organization 747
21.5 Intel Core i7-5960X 756
21.6 ARM Cortex-A15 MPCore 757
21.7 IBM z13 Mainframe 762
21.8 Key Terms, Review Questions, and Problems 765

Appendix A System Buses 768

A.1 Bus Structure 769
A.2 Multiple-Bus Hierarchies 770
A.3 Elements of Bus Design 772

Appendix B Victim Cache Strategies 777

B.1 Victim Cache 778
B.2 Selective Victim Cache 780

Appendix C Interleaved Memory 782

Appendix D The International Reference Alphabet 785

Appendix E Stacks 788

E.1 Stacks 789
E.2 Stack Implementation 790
E.3 Expression Evaluation 791

Appendix F Recursive Procedures 795

F.1 Recursion 796
F.2 Activation Tree Representation 797
F.3 Stack Implementation 803
F.4 Recursion and Iteration 804

Appendix G Additional Instruction Pipeline Topics 807

G.1 Pipeline Reservation Tables 808
G.2 Reorder Buffers 815
G.3 Tomasulo’s Algorithm 818
G.4 Scoreboarding 822

Glossary 826

References 835

Index 844
WHAT’S NEW IN THE ELEVENTH EDITION

Since the tenth 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 tenth 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:

- **Multichip Modules:** A new discussion of MCMs, which are now widely used, has been added to Chapter 1.

- **SPEC benchmarks:** The treatment of SPEC in Chapter 2 has been updated to cover the new SPEC CPU2017 benchmark suite.

- **Memory hierarchy:** A new chapter on memory hierarchy expands on material that was in the cache memory chapter, plus adds new material. The new Chapter 4 includes:
  - Updated and expanded coverage of the principle of locality
  - Updated and expanded coverage of the memory hierarchy
  - A new treatment of performance modeling of data access in a memory hierarchy

- **Cache memory:** The cache memory chapter has been updated and revised. Chapter 5 now includes:
  - Revised and expanded treatment of logical cache organization, including new figures, to improve clarity
  - New coverage of content-addressable memory
  - New coverage of write allocate and no write allocate policies
  - A new section on cache performance modeling.

- **Embedded DRAM:** Chapter 6 on internal memory now includes a section on the increasingly popular eDRAM.
Advanced Format 4k sector hard drives: Chapter 7 on external memory now includes discussion of the now widely used 4k sector hard drive format.

Boolean algebra: The discussion on Boolean algebra in Chapter 12 has been expanded with new text, figures, and tables, to enhance understanding.

Assembly language: The treatment of assembly language has been expanded to a full chapter, with more detail and more examples.

Pipeline organization: The discussion on pipeline organization has been substantially expanded with new text and figures. The material is in new sections in Chapters 16 (Processor Structure and Function), 17 (RISC), and 18 (Superscalar).

Cache coherence: The discussion of the MESI cache coherence protocol in Chapter 20 has been expanded with new text and figures.

SUPPORT OF ACM/IEEE COMPUTER SCIENCE AND COMPUTER ENGINEERING CURRICULA

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 supports 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. This book also supports the ACM/IEEE Computer Engineering Curricula 2016 (CE2016). CE2016 defines a necessary body of knowledge for undergraduate computer engineering, divided into twelve knowledge areas. One of these areas is Computer Architecture and Organization (CE-CAO), consisting of ten core knowledge areas. This text covers all of the CE-CAO knowledge areas listed in CE2016. Table P.2 shows the coverage.

<table>
<thead>
<tr>
<th>IAS Knowledge Units</th>
<th>Topics</th>
<th>Textbook Coverage</th>
</tr>
</thead>
</table>
| 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 12 |
| Machine Level Representation of Data (Tier 2) | Bits, bytes, and words  
Numeric data representation and number bases  
Fixed- and floating-point systems  
Signed and two-complement representations  
Representation of non-numeric data (character codes, graphical data) | — Chapter 10  
— Chapter 11 |
<table>
<thead>
<tr>
<th>IAS Knowledge Units</th>
<th>Topics</th>
<th>Textbook Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assembly Level Machine Organization (Tier 2)</td>
<td>Basic organization of the von Neumann machine</td>
<td>– Chapter 1</td>
</tr>
<tr>
<td></td>
<td>Control unit; instruction fetch, decode, and execution</td>
<td>– Chapter 8</td>
</tr>
<tr>
<td></td>
<td>Instruction sets and types (data manipulation, control, I/O)</td>
<td>– Chapter 13</td>
</tr>
<tr>
<td></td>
<td>Assembly/machine language programming</td>
<td>– Chapter 14</td>
</tr>
<tr>
<td></td>
<td>Instruction formats</td>
<td>– Chapter 15</td>
</tr>
<tr>
<td></td>
<td>Addressing modes</td>
<td>– Chapter 19</td>
</tr>
<tr>
<td></td>
<td>Subroutine call and return mechanisms (cross-reference PL/Language Translation and Execution)</td>
<td>– Chapter 20</td>
</tr>
<tr>
<td></td>
<td>I/O and interrupts</td>
<td>– Chapter 21</td>
</tr>
<tr>
<td></td>
<td>Shared memory multiprocessors/multicore organization</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Introduction to SIMD vs. MIMD and the Flynn Taxonomy</td>
<td></td>
</tr>
<tr>
<td>Memory System Organization and Architecture (Tier 2)</td>
<td>Storage systems and their technology</td>
<td>– Chapter 4</td>
</tr>
<tr>
<td></td>
<td>Memory hierarchy: temporal and spatial locality</td>
<td>– Chapter 5</td>
</tr>
<tr>
<td></td>
<td>Main memory organization and operations</td>
<td>– Chapter 6</td>
</tr>
<tr>
<td></td>
<td>Latency, cycle time, bandwidth, and interleaving</td>
<td>– Chapter 7</td>
</tr>
<tr>
<td></td>
<td>Cache memories (address mapping, block size, replacement and store policy)</td>
<td>– Chapter 9</td>
</tr>
<tr>
<td></td>
<td>Multiprocessor cache consistency/Using the memory system for inter-core synchronization/atomic memory operations</td>
<td>– Chapter 20</td>
</tr>
<tr>
<td></td>
<td>Virtual memory (page table, TLB)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fault handling and reliability</td>
<td></td>
</tr>
<tr>
<td>Interfacing and Communication (Tier 2)</td>
<td>I/O fundamentals: handshaking, buffering, programmed I/O</td>
<td>– Chapter 3</td>
</tr>
<tr>
<td></td>
<td>Interrupt structures: vectored and prioritized, interrupt acknowledgment</td>
<td>– Chapter 7</td>
</tr>
<tr>
<td></td>
<td>External storage, physical organization, and drives</td>
<td>– Chapter 8</td>
</tr>
<tr>
<td></td>
<td>Buses: bus protocols, arbitration, direct-memory access (DMA)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RAID architectures</td>
<td></td>
</tr>
<tr>
<td>Functional Organization (Elective)</td>
<td>Implementation of simple datapaths, including instruction pipelining, hazard detection, and resolution</td>
<td>– Chapter 16</td>
</tr>
<tr>
<td></td>
<td>Control unit: hardwired realization vs. microprogrammed realization</td>
<td>– Chapter 17</td>
</tr>
<tr>
<td></td>
<td>Instruction pipelining</td>
<td>– Chapter 18</td>
</tr>
<tr>
<td></td>
<td>Introduction to instruction-level parallelism (ILP)</td>
<td>– Chapter 19</td>
</tr>
<tr>
<td>Multiprocessing and Alternative Architectures (Elective)</td>
<td>Example SIMD and MIMD instruction sets and architectures</td>
<td>– Chapter 20</td>
</tr>
<tr>
<td></td>
<td>Interconnection networks</td>
<td>– Chapter 21</td>
</tr>
<tr>
<td></td>
<td>Shared multiprocessor memory systems and memory consistency</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Multiprocessor cache coherence</td>
<td></td>
</tr>
<tr>
<td>Performance Enhancements (Elective)</td>
<td>Superscalar architecture</td>
<td>– Chapter 17</td>
</tr>
<tr>
<td></td>
<td>Branch prediction, Speculative execution, Out-of-order execution</td>
<td>– Chapter 18</td>
</tr>
<tr>
<td></td>
<td>Prefetching</td>
<td>– Chapter 20</td>
</tr>
<tr>
<td></td>
<td>Vector processors and GPUs</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Hardware support for multithreading</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Scalability</td>
<td></td>
</tr>
</tbody>
</table>
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:

- Introduction
- The computer system
- Arithmetic and logic
- Instruction sets and assembly language
- The central processing unit
- Parallel organization, including multicore

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, and homework problems. 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.pearson.com/stallings. To gain access to the IRC, please contact your local Pearson sales representative via www.pearson.com/replocator. 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.

STUDENT RESOURCES

For this new edition, a tremendous amount of original supporting material for students has been made available online. The Companion Web Site, at www.pearson.com/stallings, includes a list of relevant links organized by chapter and an errata sheet for the book. To aid the student in understanding the material, a separate set of homework problems with solutions are available at this site. Students can enhance their understanding of the material by working out the solutions to these problems and then checking their answers. The site also includes a number of documents and papers referenced throughout the text.

PROJECTS AND OTHER STUDENT EXERCISES

For many instructors, an important component of a computer organization and architecture course is a project or set of projects by which the student gets hands-on experience to reinforce concepts from the text. This book provides an unparalleled degree of support for including a projects component in the course. The instructor’s support materials available through the IRC not only includes guidance on how to assign and structure the projects but also includes a set of user’s manuals for various project types plus specific assignments, all written especially for this book. Instructors can assign work in the following areas:

Interactive simulation assignments: Described subsequently.

Research projects: A series of research assignments that instruct the student to research a particular topic on the Internet and write a report.

Simulation projects: The IRC provides support for the use of the two simulation packages: SimpleScalar can be used to explore computer organization and architecture design issues. SMPCache provides a powerful educational tool for examining cache design issues for symmetric multiprocessors.

Assembly language projects: A simplified assembly language, CodeBlue, is used and assignments based on the popular Core Wars concept are provided.

Reading/report assignments: A list of papers in the literature, one or more for each chapter, that can be assigned for the student to read and then write a short report.

Writing assignments: A list of writing assignments to facilitate learning the material.

Test bank: Includes T/F, multiple choice, and fill-in-the-blank questions and answers.

This diverse set of projects and other student exercises enables the instructor to use the book as one component in a rich and varied learning experience and to tailor a course plan to meet the specific needs of the instructor and students.
INTERACTIVE SIMULATIONS

An important feature in this edition is the incorporation of interactive simulations. These simulations provide a powerful tool for understanding the complex design features of a modern computer system. A total of 20 interactive simulations are used to illustrate key functions and algorithms in computer organization and architecture design. At the relevant point in the book, an icon indicates that a relevant interactive simulation is available online for student use. Because the animations enable the user to set initial conditions, they can serve as the basis for student assignments. The instructor’s supplement includes a set of assignments, one for each of the animations. Each assignment includes several specific problems that can be assigned to students.

ACKNOWLEDGMENTS

This new edition has benefited from review by a number of people, who gave generously of their time and expertise. The following professors provided a review of the entire book: Nikhil Bhargava (Indian Institute of Management, Delhi), James Gil de Lamadrid (Bowie State University, Computer Science Department), Debra Calliss (Computer Science and Engineering, Arizona State University), Mohammed Anwaruddin (Wentworth Institute of Technology, Dept. of Computer Science), Roger Kieckhafer (Michigan Technological University, Electrical & Computer Engineering), Paul Fortier (University of Massachusetts Dartmouth, Electrical and Computer Engineering), Yan Zhang (Department of Computer Science and Engineering, University of South Florida), Patricia Roden (University of North Alabama, Computer Science and Information Systems), Sanjeev Baskiyar (Auburn University, Computer Science and Software Engineering), and (Jayson Rock, University of Wisconsin-Milwaukee, Computer Science). I would especially like to thank Professor Roger Kieckhafer for permission to make use of some of the figures and performance models from his course lecture notes.

Thanks also to the many people who provided detailed technical reviews of one or more chapters: Rekai Gonzalez Alberquilla, Allen Baum, Jalil Boukhobza, Dmitry Bufistov, Humberto Calderón, Jesus Carretero, Ashkan Eghbal, Peter Glaskowsky, Ram Huggahalli, Chris Jesshope, Athanasios Kakarountas, Isil Oz, Mitchell Poplingher, Roger Shepherd, Jigar Savla, Karl Stevens, Siri Uppalapati, Dr. Sriram Vajapeyam, Kugan Vivekanandarajah, Pooria M. Yaghini, and Peter Zeno,

Professor Cindy Norris of Appalachian State University, Professor Bin Mu of the University of New Brunswick, and Professor Kenrick Mock of the University of Alaska kindly supplied homework problems.

Aswin Sreedhar of the University of Massachusetts developed the interactive simulation assignments.

Professor Miguel Angel Vega Rodriguez, Professor Dr. Juan Manuel Sánchez Pérez, and Professor Dr. Juan Antonio Gómez Pulido, all of University of Extremadura, Spain, prepared the SMPCache problems in the instructor’s manual and authored the SMPCache User’s Guide.
Todd Bezenek of the University of Wisconsin and James Stine of Lehigh University prepared the SimpleScalar problems in the instructor’s manual, and Todd also authored the SimpleScalar User’s Guide.

Finally, I would like to thank the many people responsible for the publication of the book, all of whom did their usual excellent job. This includes the staff at Pearson, particularly my editor Tracy Johnson, her assistant Meghan Jacoby, and project manager Bob Engelhardt. Thanks also to the marketing and sales staffs at Pearson, without whose efforts this book would not be in front of you.
Dr. William Stallings has authored 18 textbooks, and counting revised editions, over 70 books on computer security, computer networking, and computer architecture. In over 30 years in the field, he has been a technical contributor, technical manager, and an executive with several high-technology firms. Currently, he is an independent consultant whose clients have included computer and networking manufacturers and customers, software development firms, and leading-edge government research institutions. He has 13 times received the award for the best computer science textbook of the year from the Text and Academic Authors Association.

He created and maintains the Computer Science Student Resource Site at ComputerScienceStudent.com. This site provides documents and links on a variety of subjects of general interest to computer science students (and professionals). He is a member of the editorial board of Cryptologia, a scholarly journal devoted to all aspects of cryptology.

Dr. Stallings holds a PhD from MIT in computer science and a BS from Notre Dame in electrical engineering.