HZ BOOKS
华章教育

ELSEVIER

经 典 原 版 书 库

计算机组成与设计

硬件/软件接口

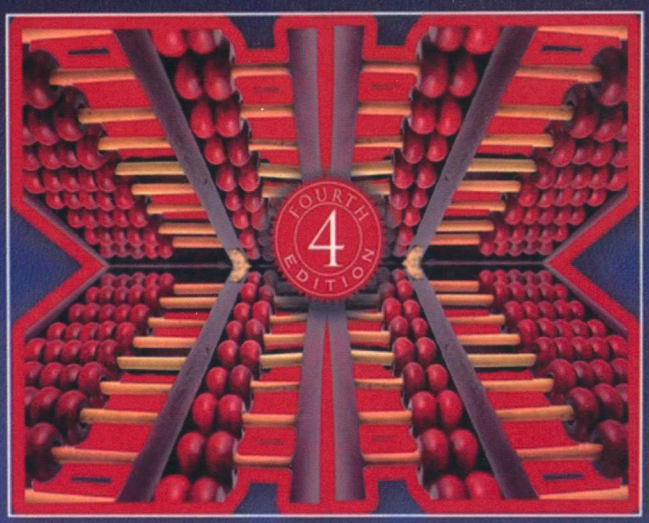
（美） David A. Patterson 加州大学伯克利分校 John L. Hennessy 斯坦福大学 著

（英文版·第4版）

Computer Organization and Design The Hardware/Software Interface (Fourth Edition)

COMPUTER ORGANIZATION AND DESIGN

THE HARDWARE / SOFTWARE INTERFACE



DAVID A. PATTERSON
JOHN L. HENNESSY

附光盘

机械工业出版社
China Machine Press

MK
MORGAN KAUFMANN

典 原 版 书 库

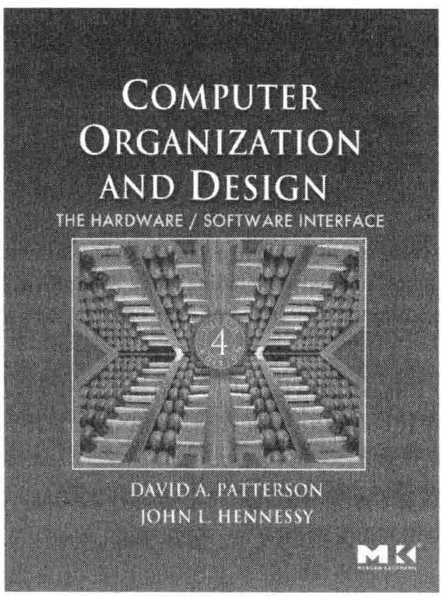
计算机组成与设计

硬件/软件接口

（英文版·第4版）

Computer Organization and Design

The Hardware/Software Interface (Fourth Edition)



（美） David A. Patterson 加州大学伯克利分校 John L. Hennessy 斯坦福大学 著

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

计算机组成与设计：硬件/软件接口（英文版·第4版）/（美）帕特森（Patterson, D. A.），（美）亨尼斯（Hennessy, J. L.）著. —北京：机械工业出版社，2013.1
（经典原版书库）
书名原文：Computer Organization and Design, The Hardware/Software Interface, Fourth Edition

ISBN 978-7-111-41237-3

I. 计… II. ① 帕… ② 亨… III. ① 计算机体系结构－英文 ② 微型计算机－接口设备－英文
IV. ① TP303 ② TP364

中国版本图书馆CIP数据核字（2013）第013509号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）
责任编辑：迟振春
北京诚信伟业印刷有限公司印刷
2013年2月第1版第1次印刷
186mm×240mm·56.75印张
标准书号：ISBN 978-7-111-41237-3
ISBN 978-7-89433-783-2（光盘）
定价：139.00元（附光盘）

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991 88361066 投稿热线：（010）88379604
购书热线：（010）68326294 88379649 68995259 读者信箱：hzjsj@hzbook.com

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自 1998 年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与 Pearson，McGraw-Hill，Elsevier，MIT，John Wiley & Sons，Cengage 等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出 Andrew S. Tanenbaum，Bjarne Stroustrup，Brain W. Kernighan，Dennis Ritchie，Jim Gray，Afred V. Aho，John E. Hopcroft，Jeffrey D. Ullman，Abraham Silberschatz，William Stallings，Donald E. Knuth，John L. Hennessy，Larry L. Peterson 等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

华章网站：www.hzbook.com

电子邮件：hzjsj@hzbook.com

联系电话：（010）88379604

联系地址：北京市西城区百万庄南街 1 号

邮政编码：100037

华章教育

华章科技图书出版中心

In Praise of *Computer Organization and Design: The Hardware/Software Interface*, Fourth Edition

"Patterson and Hennessy not only improve the pedagogy of the traditional material on pipelined processors and memory hierarchies, but also greatly expand the multiprocessor coverage to include emerging multicore processors and GPUs. The fourth edition of *Computer Organization and Design* sets a new benchmark against which all other architecture books must be compared."

—David A. Wood, *University of Wisconsin-Madison*

"Patterson and Hennessy have greatly improved what was already the gold-standard of textbooks. In the rapidly evolving field of computer architecture, they have woven an impressive number of recent case studies and contemporary issues into a framework of time-tested fundamentals."

—Fred Chong, *University of California at Santa Barbara*

"Since the publication of the first edition in 1994, *Computer Organization and Design* has introduced a generation of computer science and engineering students to computer architecture. Now, many of those students have become leaders in the field. In academia, the tradition continues as faculty use the latest edition of the book that inspired them to engage the next generation. With the fourth edition, readers are prepared for the next era of computing."

—David I. August, *Princeton University*

"The new coverage of multiprocessors and parallelism lives up to the standards of this well-written classic. It provides well-motivated, gentle introductions to the new topics, as well as many details and examples drawn from current hardware."

—John Greiner, *Rice University*

"As computer hardware architecture moves from uniprocessor to multicores, the parallel programming environments used to take advantage of these cores will be a defining challenge to the success of these new systems. In the multicore systems, the interface between the hardware and software is of particular importance. This new edition of *Computer Organization and Design* is mandatory for any student who wishes to understand multicore architecture including the interface between programming it and its architecture."

—Jesse Fang, *Director of Programming System Lab at Intel*

"The fourth edition of *Computer Organization and Design* continues to improve the high standards set by the previous editions. The new content, on trends that are reshaping computer systems including multicores, Flash memory, GPUs, etc., makes this edition a must read—even for all of those who grew up on previous editions of the book."

—Parthasarathy Ranganathan, *Principal Research Scientist, HP Labs*

Preface

The most beautiful thing we can experience is the mysterious.
It is the source of all true art and science.

Albert Einstein, *What I Believe*, 1930

About This Book

We believe that learning in computer science and engineering should reflect the current state of the field, as well as introduce the principles that are shaping computing. We also feel that readers in every specialty of computing need to appreciate the organizational paradigms that determine the capabilities, performance, and, ultimately, the success of computer systems.

Modern computer technology requires professionals of every computing specialty to understand both hardware and software. The interaction between hardware and software at a variety of levels also offers a framework for understanding the fundamentals of computing. Whether your primary interest is hardware or software, computer science or electrical engineering, the central ideas in computer organization and design are the same. Thus, our emphasis in this book is to show the relationship between hardware and software and to focus on the concepts that are the basis for current computers.

The recent switch from uniprocessor to multicore microprocessors confirmed the soundness of this perspective, given since the first edition. While programmers could ignore the advice and rely on computer architects, compiler writers, and silicon engineers to make their programs run faster without change, that era is over. For programs to run faster, they must become parallel. While the goal of many researchers is to make it possible for programmers to be unaware of the underlying parallel nature of the hardware they are programming, it will take many years to realize this vision. Our view is that for at least the next decade, most programmers are going to have to understand the hardware/software interface if they want programs to run efficiently on parallel computers.

The audience for this book includes those with little experience in assembly language or logic design who need to understand basic computer organization as well as readers with backgrounds in assembly language and/or logic design who want to learn how to design a computer or understand how a system works and why it performs as it does.

About the Other Book

Some readers may be familiar with *Computer Architecture: A Quantitative Approach*, popularly known as Hennessy and Patterson. (This book in turn is often called Patterson and Hennessy.) Our motivation in writing the earlier book was to describe the principles of computer architecture using solid engineering fundamentals and quantitative cost/performance tradeoffs. We used an approach that combined examples and measurements, based on commercial systems, to create realistic design experiences. Our goal was to demonstrate that computer architecture could be learned using quantitative methodologies instead of a descriptive approach. It was intended for the serious computing professional who wanted a detailed understanding of computers.

A majority of the readers for this book do not plan to become computer architects. The performance and energy efficiency of future software systems will be dramatically affected, however, by how well software designers understand the basic hardware techniques at work in a system. Thus, compiler writers, operating system designers, database programmers, and most other software engineers need a firm grounding in the principles presented in this book. Similarly, hardware designers must understand clearly the effects of their work on software applications.

Thus, we knew that this book had to be much more than a subset of the material in *Computer Architecture*, and the material was extensively revised to match the different audience. We were so happy with the result that the subsequent editions of *Computer Architecture* were revised to remove most of the introductory material; hence, there is much less overlap today than with the first editions of both books.

Changes for the Fourth Edition

We had five major goals for the fourth edition of *Computer Organization and Design:* given the multicore revolution in microprocessors, highlight parallel hardware and software topics throughout the book; streamline the existing material to make room for topics on parallelism; enhance pedagogy in general; update the technical content to reflect changes in the industry since the publication of the third edition in 2004; and restore the usefulness of exercises in this Internet age.

Before discussing the goals in detail, let's look at the table on the next page. It shows the hardware and software paths through the material. Chapters 1, 4, 5, and 7 are found on both paths, no matter what the experience or the focus. Chapter 1 is a new introduction that includes a discussion on the importance of power and how it motivates the switch from single core to multicore microprocessors. It also includes performance and benchmarking material that was a separate chapter in the third edition. Chapter 2 is likely to be review material for the hardware-oriented, but it is essential reading for the software-oriented, especially for those readers interested in learning more about compilers and object-oriented programming

Chapter or appendix	Sections	Software focus	Hardware focus
1. Computer Abstractions and Technology	1.1 to 1.9		
	1.10 (History)		
2. Instructions: Language of the Computer	2.1 to 2.14		
	2.15 (Compilers & Java)		
	2.16 to 2.19		
	2.20 (History)		
E. RISC Instruction-Set Architectures	E.1 to E.19		
3. Arithmetic for Computers	3.1 to 3.9		
	3.10 (History)		
C. The Basics of Logic Design	C.1 to C.13		
4. The Processor	4.1 (Overview)		
	4.2 (Logic Conventions)		
	4.3 to 4.4 (Simple Implementation)		
	4.5 (Pipelining Overview)		
	4.6 (Pipelined Datapath)		
	4.7 to 4.9 (Hazards, Exceptions)		
	4.10 to 4.11 (Parallel, Real Stuff)		
	4.12 (Verilog Pipeline Control)		
	4.13 to 4.14 (Fallacies)		
	4.15 (History)		
D. Mapping Control to Hardware	D.1 to D.6		
5. Large and Fast: Exploiting Memory Hierarchy	5.1 to 5.8		
	5.9 (Verilog Cache Controller)		
	5.10 to 5.12		
	5.13 (History)		
6. Storage and Other I/O Topics	6.1 to 6.10		
	6.11 (Networks)		
	6.12 to 6.13		
	6.14 (History)		
7. Multicores, Multiprocessors, and Clusters	7.1 to 7.13		
	7.14 (History)		
A. Graphics Processor Units	A.1 to A.12		
B. Assemblers, Linkers, and the SPIM Simulator	B.1 to B.12		

Read carefully Read if have time Reference

Review or read Read for culture

languages. It includes material from Chapter 3 in the third edition so that the complete MIPS architecture is now in a single chapter, minus the floating-point instructions. Chapter 3 is for readers interested in constructing a datapath or in learning more about floating-point arithmetic. Some will skip Chapter 3, either because they don't need it or because it is a review. Chapter 4 combines two chapters from the third edition to explain pipelined processors. Sections 4.1, 4.5, and 4.10 give overviews for those with a software focus. Those with a hardware focus, however, will find that this chapter presents core material; they may also, depending on their background, want to read Appendix C on logic design first. Chapter 6 on storage is critical to readers with a software focus, and should be read by others if time permits. The last chapter on multicores, multiprocessors, and clusters is mostly new content and should be read by everyone.

The first goal was to make parallelism a first class citizen in this edition, as it was a separate chapter on the CD in the last edition. The most obvious example is Chapter 7. In particular, this chapter introduces the Roofline performance model, and shows its value by evaluating four recent multicore architectures on two kernels. This model could prove to be as insightful for multicore microprocessors as the 3Cs model is for caches.

Given the importance of parallelism, it wasn't wise to wait until the last chapter to talk about, so there is a section on parallelism in each of the preceding six chapters:

- *Chapter 1: Parallelism and Power.* It shows how power limits have forced the industry to switch to parallelism, and why parallelism helps.
- *Chapter 2: Parallelism and Instructions: Synchronization.* This section discusses locks for shared variables, specifically the MIPS instructions Load Linked and Store Conditional.
- *Chapter 3: Parallelism and Computer Arithmetic: Floating-Point Associativity.* This section discusses the challenges of numerical precision and floating-point calculations.
- *Chapter 4: Parallelism and Advanced Instruction-Level Parallelism.* It covers advanced ILP—superscalar, speculation, VLIW, loop-unrolling, and OOO—as well as the relationship between pipeline depth and power consumption.
- *Chapter 5: Parallelism and Memory Hierarchies: Cache Coherence.* It introduces coherency, consistency, and snooping cache protocols.
- *Chapter 6: Parallelism and I/O: Redundant Arrays of Inexpensive Disks.* It describes RAID as a parallel I/O system as well as a highly available ICO system.

Chapter 7 concludes with reasons for optimism why this foray into parallelism should be more successful than those of the past.

I am particularly excited about the addition of an appendix on Graphical Processing Units written by NVIDIA's chief scientist, David Kirk, and chief architect, John Nickolls. Appendix A is the first in-depth description of GPUs, which is a new and interesting thrust in computer architecture. The appendix builds upon the parallel themes of this edition to present a style of computing that allows the programmer to think MIMD yet the hardware tries to execute in SIMD-style whenever possible. As GPUs are both inexpensive and widely available—they are even found in many laptops—and their programming environments are freely available, they provide a parallel hardware platform that many could experiment with.

The second goal was to streamline the book to make room for new material in parallelism. The first step was simply going through all the paragraphs accumulated over three editions with a fine-toothed comb to see if they were still necessary. The coarse-grained changes were the merging of chapters and dropping of topics. Mark Hill suggested dropping the multicycle processor implementation and instead adding a multicycle cache controller to the memory hierarchy chapter. This allowed the processor to be presented in a single chapter instead of two, enhancing the processor material by omission. The performance material from a separate chapter in the third edition is now blended into the first chapter.

The third goal was to improve the pedagogy of the book. Chapter 1 is now meatier, including performance, integrated circuits, and power, and it sets the stage for the rest of the book. Chapters 2 and 3 were originally written in an evolutionary style, starting with a "single celled" architecture and ending up with the full MIPS architecture by the end of Chapter 3. This leisurely style is not a good match to the modern reader. This edition merges all of the instruction set material for the integer instructions into Chapter 2—making Chapter 3 optional for many readers—and each section now stands on its own. The reader no longer needs to read all of the preceding sections. Hence, Chapter 2 is now even better as a reference than it was in prior editions. Chapter 4 works better since the processor is now a single chapter, as the multicycle implementation is a distraction today. Chapter 5 has a new section on building cache controllers, along with a new CD section containing the Verilog code for that cache.

The accompanying CD-ROM introduced in the third edition allowed us to reduce the cost of the book by saving pages as well as to go into greater depth on topics that were of interest to some but not all readers. Alas, in our enthusiasm to save pages, readers sometimes found themselves going back and forth between the CD and book more often than they liked. This should not be the case in this edition. Each chapter now has the Historical Perspectives section on the CD and four chapters also have one advanced material section on the CD. Additionally, all exercises are in the printed book, so flipping between book and CD should be rare in this edition.

For those of you who wonder why we include a CD-ROM with the book, the answer is simple: the CD contains content that we feel should be easily and immediately accessible to the reader no matter where they are. If you are interested in the advanced content, or would like to review a VHDL tutorial (for example), it is on the CD, ready for you to use. The CD-ROM also includes a feature that should greatly enhance your study of the material: a search engine is included that allows you to search for any string of text, in the printed book or on the CD itself. If you are hunting for content that may not be included in the book's printed index, you can simply enter the text you're searching for and the page number it appears on will be displayed in the search results. This is a very useful feature that we hope you make frequent use of as you read and review the book.

This is a fast-moving field, and as is always the case for our new editions, an important goal is to update the technical content. The AMD Opteron X4 model 2356 (code named "Barcelona") serves as a running example throughout the book, and is found in Chapters 1, 4, 5, and 7. Chapters 1 and 6 add results from the new power benchmark from SPEC. Chapter 2 adds a section on the ARM architecture, which is currently the world's most popular 32-bit ISA. Chapter 5 adds a new section on Virtual Machines, which are resurging in importance. Chapter 5 has detailed cache performance measurements on the Opteron X4 multicore and a few details on its rival, the Intel Nehalem, which will not be announced until after this edition is published. Chapter 6 describes Flash Memory for the first time as well as a remarkably compact server from Sun, which crams 8 cores, 16 DIMMs, and 8 disks into a single 1U bit. It also includes the recent results on long-term disk failures. Chapter 7 covers a wealth of topics regarding parallelism—including multithreading, SIMD, vector, GPUs, performance models, benchmarks, multiprocessor networks—and describes three multicores plus the Opteron X4: Intel Xeon model e5345 (Clovertown), IBM Cell model QS20, and the Sun Microsystems T2 model 5120 (Niagara 2).

The final goal was to try to make the exercises useful to instructors in this Internet age, for homework assignments have long been an important way to learn material. Alas, answers are posted today almost as soon as the book appears. We have a two-part approach. First, expert contributors have worked to develop entirely new exercises for each chapter in the book. Second, most exercises have a qualitative description supported by a table that provides several alternative quantitative parameters needed to answer this question. The sheer number plus flexibility in terms of how the instructor can choose to assign variations of exercises will make it hard for students to find the matching solutions online. Instructors will also be able to change these quantitative parameters as they wish, again frustrating those students who have come to rely on the Internet to provide solutions for a static and unchanging set of exercises. We feel this new approach is a valuable new addition to the book—please let us know how well it works for you, either as a student or instructor!

We have preserved useful book elements from prior editions. To make the book work better as a reference, we still place definitions of new terms in the margins at their first occurrence. The book element called "Understanding Program Performance" sections helps readers understand the performance of their programs and how to improve it, just as the "Hardware/Software Interface" book element helped readers understand the tradeoffs at this interface. "The Big Picture" section remains so that the reader sees the forest even despite all the trees. "Check Yourself" sections help readers to confirm their comprehension of the material on the first time through with answers provided at the end of each chapter. This edition also includes the green MIPS reference card, which was inspired by the "Green Card" of the IBM System/360. The removable card has been updated and should be a handy reference when writing MIPS assembly language programs.

Instructor Support

We have collected a great deal of material to help instructors teach courses using this book. Solutions to exercises, chapter quizzes, figures from the book, lecture notes, lecture slides, and other materials are available to adopters from the publisher. Check the publisher's Web site for more information:

textbooks.elsevier.com/9780123744937

Concluding Remarks

If you read the following acknowledgments section, you will see that we went to great lengths to correct mistakes. Since a book goes through many printings, we have the opportunity to make even more corrections. If you uncover any remaining, resilient bugs, please contact the publisher by electronic mail at *cod4bugs@mkp.com* or by low-tech mail using the address found on the copyright page.

This edition marks a break in the long-standing collaboration between Hennessy and Patterson, which started in 1989. The demands of running one of the world's great universities meant that President Hennessy could no longer make the substantial commitment to create a new edition. The remaining author felt like a juggler who had always performed with a partner who suddenly is thrust on the stage as a solo act. Hence, the people in the acknowledgments and Berkeley colleagues played an even larger role in shaping the contents of this book. Nevertheless, this time around there is only one author to blame for the new material in what you are about to read.

Acknowledgments for the Fourth Edition

I'd like to thank **David Kirk**, **John Nickolls**, and their colleagues at NVIDIA (Michael

Garland, John Montrym, Doug Voorhies, Lars Nyland, Erik Lindholm, Paulius Micikevicius, Massimiliano Fatica, Stuart Oberman, and Vasily Volkov) for writing the first in-depth appendix on GPUs. I'd like to express again my appreciation to **Jim Larus** of Microsoft Research for his willingness in contributing his expertise on assembly language programming, as well as for welcoming readers of this book to use the simulator he developed and maintains.

I am also very grateful for the contributions of the many experts who developed the new exercises for this new edition. Writing good exercises is not an easy task, and each contributor worked long and hard to develop problems that are both challenging and engaging:

- *Chapter 1:* **Javier Bruguera** (Universidade de Santiago de Compostela)
- *Chapter 2:* **John Oliver** (Cal Poly, San Luis Obispo), with contributions from **Nicole Kaiyan** (University of Adelaide) and **Milos Prvulovic** (Georgia Tech)
- *Chapter 3:* **Matthew Farrens** (University of California, Davis)
- *Chapter 4:* **Milos Prvulovic** (Georgia Tech)
- *Chapter 5:* **Jichuan Chang**, **Jacob Leverich, Kevin Lim**, and **Partha Ranganathan** (all from Hewlett-Packard), with contributions from Nicole Kaiyan (University of Adelaide)
- *Chapter 6:* **Perry Alexander** (The University of Kansas)
- *Chapter 7:* **David Kaeli** (Northeastern University)

Peter Ashenden took on the Herculean task of editing and evaluating *all* of the new exercises. Moreover, he even added the substantial burden of developing the companion CD and new lecture slides.

Thanks to **David August** and **Prakash Prabhu** of Princeton University for their work on the chapter quizzes that are available for instructors on the publisher's Web site.

I relied on my Silicon Valley colleagues for much of the technical material that this book relies upon:

- **AMD**—for the details and measurements of the Opteron X4 (Barcelona): **William Brantley**, **Vasileios Liaskovitis**, **Chuck Moore**, and **Brian Waldecker.**
- **Intel**—for the prereleased information on the Intel Nehalem: **Faye Briggs**.
- **Micron**—for background on Flash Memory in Chapter 6: **Dean Klein**.
- **Sun Microsystems**—for the measurements of the instruction mixes for the SPEC2006 benchmarks in Chapter 2 and details and measurements of the Sun Server x4150 in Chapter 6: **Yan Fisher**, **John Fowler**, **Darryl Gove**, **Paul**

Joyce, Shenik Mehta, Pierre Reynes, Dimitry Stuve, Durgam Vahia, and **David Weaver.**

- **U.C. Berkeley—Krste Asanovic** (who supplied the idea for software concurrency versus hardware parallelism in Chapter 7), **James Demmel** and **Velvel Kahan** (who commented on parallelism and floating-point calculations), **Zhangxi Tan** (who designed the cache controller and wrote the Verilog for it in Chapter 5), **Sam Williams** (who supplied the roofline model and the multicore measurements in Chapter 7), and the rest of my colleagues in the **Par Lab** who gave extensive suggestions and feedback on parallelism topics found throughout the book.

I am grateful to the many instructors who answered the publisher's surveys, reviewed our proposals, and attended focus groups to analyze and respond to our plans for this edition. They include the following individuals: *Focus Group:* Mark Hill (University of Wisconsin, Madison), E.J. Kim (Texas A&M University), Jihong Kim (Seoul National University), Lu Peng (Louisiana State University), Dean Tullsen (UC San Diego), Ken Vollmar (Missouri State University), David Wood (University of Wisconsin, Madison), Ki Hwan Yum (University of Texas, San Antonio); *Surveys and Reviews:* Mahmoud Abou-Nasr (Wayne State University), Perry Alexander (The University of Kansas), Hakan Aydin (George Mason University), Hussein Badr (State University of New York at Stony Brook), Mac Baker (Virginia Military Institute), Ron Barnes (George Mason University), Douglas Blough (Georgia Institute of Technology), Kevin Bolding (Seattle Pacific University), Miodrag Bolic (University of Ottawa), John Bonomo (Westminster College), Jeff Braun (Montana Tech), Tom Briggs (Shippensburg University), Scott Burgess (Humboldt State University), Fazli Can (Bilkent University), Warren R. Carithers (Rochester Institute of Technology), Bruce Carlton (Mesa Community College), Nicholas Carter (University of Illinois at Urbana-Champaign), Anthony Cocchi (The City University of New York), Don Cooley (Utah State University), Robert D. Cupper (Allegheny College), Edward W. Davis (North Carolina State University), Nathaniel J. Davis (Air Force Institute of Technology), Molisa Derk (Oklahoma City University), Derek Eager (University of Saskatchewan), Ernest Ferguson (Northwest Missouri State University), Rhonda Kay Gaede (The University of Alabama), Etienne M. Gagnon (UQAM), Costa Gerousis (Christopher Newport University), Paul Gillard (Memorial University of Newfoundland), Michael Goldweber (Xavier University), Georgia Grant (College of San Mateo), Merrill Hall (The Master's College), Tyson Hall (Southern Adventist University), Ed Harcourt (Lawrence University), Justin E. Harlow (University of South Florida), Paul F. Hemler (Hampden-Sydney College), Martin Herbordt (Boston University), Steve J. Hodges (Cabrillo College), Kenneth Hopkinson (Cornell University), Dalton Hunkins (St. Bonaventure University), Baback Izadi (State University of New York—New Paltz), Reza Jafari, Robert W. Johnson (Colorado Technical University), Bharat Joshi (University of North Carolina,

Charlotte), Nagarajan Kandasamy (Drexel University), Rajiv Kapadia, Ryan Kastner (University of California, Santa Barbara), Jim Kirk (Union University), Geoffrey S. Knauth (Lycoming College), Manish M. Kochhal (Wayne State), Suzan Koknar-Tezel (Saint Joseph's University), Angkul Kongmunvattana (Columbus State University), April Kontostathis (Ursinus College), Christos Kozyrakis (Stanford University), Danny Krizanc (Wesleyan University), Ashok Kumar, S. Kumar (The University of Texas), Robert N. Lea (University of Houston), Baoxin Li (Arizona State University), Li Liao (University of Delaware), Gary Livingston (University of Massachusetts), Michael Lyle, Douglas W. Lynn (Oregon Institute of Technology), Yashwant K Malaiya (Colorado State University), Bill Mark (University of Texas at Austin), Ananda Mondal (Claflin University), Alvin Moser (Seattle University), Walid Najjar (University of California, Riverside), Danial J. Neebel (Loras College), John Nestor (Lafayette College), Joe Oldham (Centre College), Timour Paltashev, James Parkerson (University of Arkansas), Shaunak Pawagi (SUNY at Stony Brook), Steve Pearce, Ted Pedersen (University of Minnesota), Gregory D Peterson (The University of Tennessee), Dejan Raskovic (University of Alaska, Fairbanks) Brad Richards (University of Puget Sound), Roman Rozanov, Louis Rubinfield (Villanova University), Md Abdus Salam (Southern University), Augustine Samba (Kent State University), Robert Schaefer (Daniel Webster College), Carolyn J. C. Schauble (Colorado State University), Keith Schubert (CSU San Bernardino), William L. Schultz, Kelly Shaw (University of Richmond), Shahram Shirani (McMaster University), Scott Sigman (Drury University), Bruce Smith, David Smith, Jeff W. Smith (University of Georgia, Athens), Philip Snyder (Johns Hopkins University), Alex Sprintson (Texas A&M), Timothy D. Stanley (Brigham Young University), Dean Stevens (Morningside College), Nozar Tabrizi (Kettering University), Yuval Tamir (UCLA), Alexander Taubin (Boston University), Will Thacker (Winthrop University), Mithuna Thottethodi (Purdue University), Manghui Tu (Southern Utah University), Rama Viswanathan (Beloit College), Guoping Wang (Indiana-Purdue University), Patricia Wenner (Bucknell University), Kent Wilken (University of California, Davis), David Wolfe (Gustavus Adolphus College), David Wood (University of Wisconsin, Madison), Mohamed Zahran (City College of New York), Gerald D. Zarnett (Ryerson University), Nian Zhang (South Dakota School of Mines & Technology), Jiling Zhong (Troy University), Huiyang Zhou (The University of Central Florida), Weiyu Zhu (Illinois Wesleyan University).

I would especially like to thank the Berkeley people who gave key feedback for Chapter 7 and Appendix A, which were the most challenging pieces to write for this edition: **Krste Asanovic**, **Christopher Batten**, **Rastilav Bodik**, **Bryan Catanzaro, Jike Chong**, **Kaushik Data**, **Greg Giebling**, **Anik Jain**, **Jae Lee, Vasily Volkov**, and **Samuel Williams.**

A special thanks also goes to **Mark Smotherman** for making multiple passes to find technical and writing glitches that significantly improved the quality of this

edition. He played an even more important role this time given that this edition was done as a solo act.

We wish to thank the extended Morgan Kaufmann family for agreeing to publish this book again under the able leadership of **Denise Penrose**. **Nathaniel McFadden** was the developmental editor for this edition and worked with me weekly on the contents of the book. **Kimberlee Honjo** coordinated the surveying of users and their responses.

Dawnmarie Simpson managed the book production process. We thank also the many freelance vendors who contributed to this volume, especially Alan Rose of Multiscience Press and diacriTech, our compositor.

The contributions of the nearly 200 people we mentioned here have helped make this fourth edition what I hope will be our best book yet. Enjoy!

David A. Patterson

Contents

Preface v

CHAPTERS

1 **Computer Abstractions and Technology 2**

1.1 Introduction 3
1.2 Below Your Program 10
1.3 Under the Covers 13
1.4 Performance 26
1.5 The Power Wall 39
1.6 The Sea Change: The Switch from Uniprocessors to Multiprocessors 41
1.7 Real Stuff: Manufacturing and Benchmarking the AMD Opteron X4 44
1.8 Fallacies and Pitfalls 51
1.9 Concluding Remarks 54
1.10 Historical Perspective and Further Reading 55
1.11 Exercises 56

2 **Instructions: Language of the Computer 74**

2.1 Introduction 76
2.2 Operations of the Computer Hardware 77
2.3 Operands of the Computer Hardware 80
2.4 Signed and Unsigned Numbers 87
2.5 Representing Instructions in the Computer 94
2.6 Logical Operations 102
2.7 Instructions for Making Decisions 105
2.8 Supporting Procedures in Computer Hardware 112
2.9 Communicating with People 122
2.10 MIPS Addressing for 32-Bit Immediates and Addresses 128
2.11 Parallelism and Instructions: Synchronization 137
2.12 Translating and Starting a Program 139
2.13 A C Sort Example to Put It All Together 149