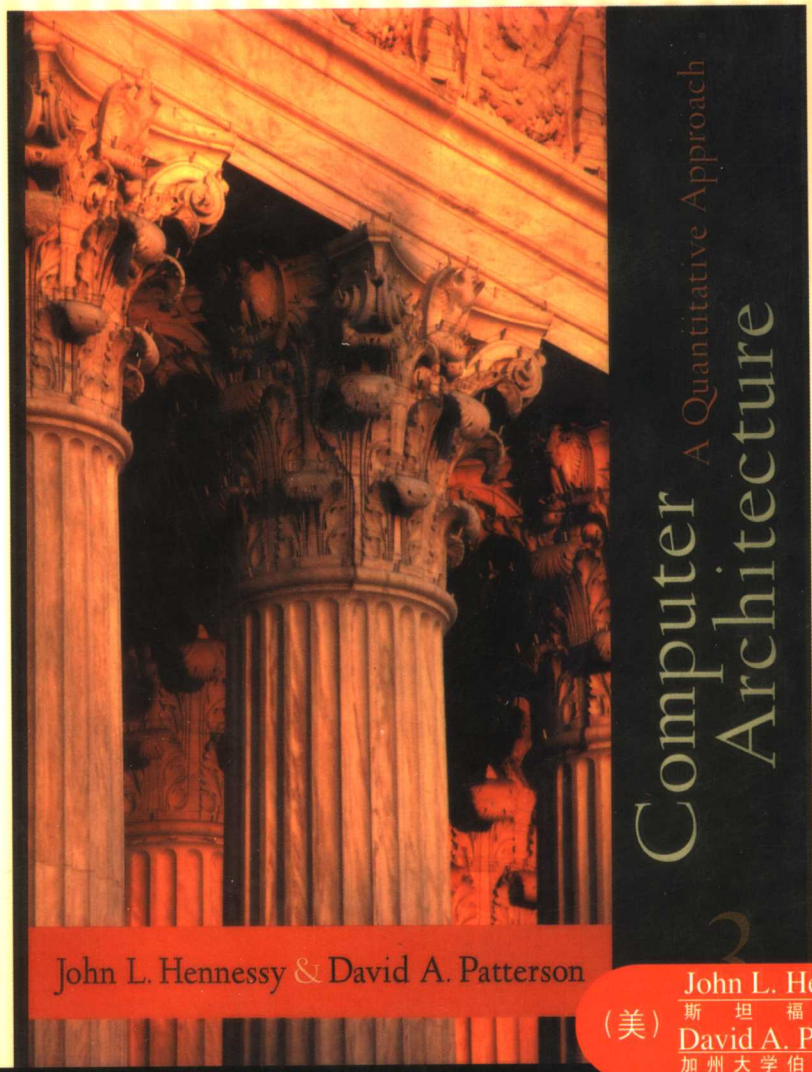
经 典 原 版 书 库

计算机体系结构
量化研究方法

（英文版 · 第3版）



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

机械工业出版社
China Machine Press

经 典 原 版 书 库

计算机体系结构
量化研究方法
(英文版 · 第3版)

Computer Architecture
A Quantitative Approach
(Third Edition)

John L. Hennessy, David A. Patterson: Computer Architecture: A Quantitative Approach, Third Edition.

ISBN: 1-55860-596-7.

本书版权登记号：图字：01-2002-3582

图书在版编目（CIP）数据

计算机体系结构：量化研究方法：第3版/（美）亨尼西（Hennessy, J. L.），（美）帕特森（Patterson, D. A.）著. -北京：机械工业出版社，2002.9

（经典原版书库）

书名原文：Computer Architecture: A Quantitative Approach

ISBN 7-111-10921-X

Ⅰ.计… Ⅱ.①亨… ②帕… Ⅲ.计算机体系结构-英文 Ⅳ.TP303

中国版本图书馆CIP数据核字（2002）第068889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：华 章

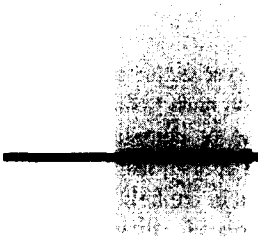
北京第二外国语学院印刷厂印刷·新华书店北京发行所发行

2002年9月第1版·2003年1月第2次印刷

787mm×1092mm 1/16·71印张

印数：3 001-5 000册

定价：99.00元



Preface

Why We Wrote This Book

Through three editions of this book, our goal has been to describe the basic principles underlying what will be tomorrow's technological developments. Our excitement about the opportunities in computer architecture has not abated, and we echo what we said about the field in the first edition: "It is not a dreary science of paper machines that will never work. No! It's a discipline of keen intellectual interest, requiring the balance of marketplace forces to cost-performance-power, leading to glorious failures and some notable successes."

Our primary objective in writing our first book was to change the way people learn and think about computer architecture. We feel this goal is still valid and important. The field is changing daily and must be studied with real examples and measurements on real computers, rather than simply as a collection of definitions and designs that will never need to be realized. We offer an enthusiastic welcome to anyone who came along with us in the past, as well as to those who are joining us now. Either way, we can promise the same quantitative approach to, and analysis of, real systems.

As with earlier versions, we have strived to produce a new edition that will continue to be as relevant for professional engineers and architects as it is for those involved in advanced computer architecture and design courses. As much as its predecessors, this edition aims to demystify computer architecture through an emphasis on cost-performance-power trade-offs and good engineering design. We believe that the field has continued to mature and move toward the rigorous quantitative foundation of long-established scientific and engineering disciplines. Our greatest satisfaction derives from the fact that the principles described in our first edition in 1990 and the second edition in 1996 could be applied successfully to help predict the landscape of computing technology that exists today. We hope that this third edition will allow readers to apply the fundamentals for similar results as we look forward to the coming decades.

This Edition

The third edition of *Computer Architecture: A Quantitative Approach* should have been easy to write. After all, our quantitative approach hasn't changed, and we sought to continue our focus on the basic principles of computer design through two editions. The examples had to be updated, of course, just as we did for the second edition. The dramatic and ongoing advances in the field as well as the creation of new markets for computers and new approaches for those markets, however, led us to rewrite almost the entire book.

The pace of innovation in computer architecture continued unabated in the six years since the second edition. As when we wrote the second edition, we found that numerous new concepts needed to be introduced, and other material designated as more basic. Although this is officially the third edition of *Computer Architecture: A Quantitative Approach,* it is really our fifth book in a series that began with the first edition, continued with *Computer Organization and Design: The Hardware/Software Interface* (COD:HSI), and then the second edition of both books. Over time ideas that were once found here have moved to COD:HSI or to background tutorials in the appendices. This migration, combined with our goal to present concepts in the context of the most recent computers, meant there was remarkably little from the second edition that could be preserved intact, and practically nothing is left from the first edition.

Perhaps the biggest surprise for us was the realization that the computer architecture field had split into three related but different market segments, each with their own needs and somewhat different architectures to address them. The cost-performance theme of our first and second editions is currently best exemplified by desktop computers. The two new paths are embedded computers and server computers. This major shift in the field is reflected in this edition by two major changes. First, throughout the text we broaden the topics considered as well as the metrics of success. Second, a new section, called "Another View," supplements the more traditional examples in "Putting It All Together" with examples that include video games, digital cameras, and cell phones.

Embedded computers have much lower cost targets than do desktop computers. They are often employed in environments where they run a single application. Also, embedded computers often rely on batteries and cannot use active cooling mechanisms, and energy/power efficiency is thus critical. To illustrate the design trade-offs and approaches in embedded processors we made several additions: the EEMBC benchmarks are used to evaluate performance, media processor and DSP instruction set principles and measurements are examined, the most popular embedded instruction set architectures are surveyed in the appendices, and performance-power trade-offs are explored in several chapters. Power-sensitive examples include the Transmeta and low-power MIPS processors, and embedded systems examples include the PlayStation-2 video game, Sanyo digital camera, and Nokia cell phone.

Server computers place more emphasis on reliability, scalability, and on throughput rather than latency to measure performance. Thus, these systems typi-

cally include multiple processors and disks. This edition explains the concept of dependability and includes rarely found statistics on the frequency of component failures. In addition to the SPEC2000 benchmarks for processors, we examine the TPC database benchmarks and the SPEC benchmarks for file servers. Examples of server processors include the Intel IA-64 and the Sun UltraSPARC III, and examples of server systems include the Sun Fire 6800, the Sun Wildfire, EMC Symmetrix, EMC Celerra, the Google cluster, and an IBM cluster for transaction processing.

This edition continues the tradition of using real-world examples to demonstrate the ideas, and the "Putting It All Together" sections are essentially 100% new. The "Putting It All Together" sections of this book include the MIPS64 instruction set architecture, the Intel Pentium III and 4 pipeline organization, the Intel IA-64 architecture and microarchitecture, the Alpha 21264 memory hierarchy, the Sun Wildfire multiprocessor, the EMC Symmetrix storage array, the EMC Celerra file server, and the Google search engine. The "Another View" sections pick real-world examples from the embedded and server communities. This list has the Trimedia TMS media processor, a PowerPC multithreaded processor, the memory hierarchy of Emotion Engine in the Sony Playstation-2, Sun Fire 6800/UltraSPARC III memory hierarchy, EmpowerTel MXP embedded multiprocessor, Sanyo digital camera, and Nokia cell phone.

In response to numerous comments, considerable effort was focused on revising and enhancing the exercises. In particular, all the exercises were reviewed to try to reduce ambiguities and eliminate unproductive exercises, and many new exercises were developed. As many readers requested, Appendix B provides answers to selected exercises.

We also added some new features that should help readers. We replaced the synthetic 32-bit DLX architecture with the popular 64-bit MIPS architecture, as it just made more sense to use existing software rather than recreate and maintain compilers ourselves. We also added a large set of appendices that contains descriptions of a dozen instruction set architectures plus tutorials on basic pipelining, vector processors, and floating-point arithmetic.

Topic Selection and Organization

As before, we have taken a conservative approach to topic selection, for there are many more interesting ideas in the field than can reasonably be covered in a treatment of basic principles. We have steered away from a comprehensive survey of every architecture a reader might encounter. Instead, our presentation focuses on core concepts likely to be found in any new machine. The key criterion remains that of selecting ideas that have been examined and utilized successfully enough to permit their discussion in quantitative terms.

Our first dilemma in determining the new topic selections was that topics requiring only a few pages in the prior editions have since exploded in their importance. Second, topics that we excluded previously have matured to a point where they can be discussed based on our quantitative criteria and their success in

the marketplace. To allow for this new material, we reduced the extent of introductory material, assuming the knowledge of the concepts in our introductory text *Computer Organization and Design: The Hardware/Software Interface*. Appendix A on pipelining was added as a valuable tutorial for readers not familiar with the basics of pipelining. (Readers interested strictly in a more basic introduction to computer architecture should read *Computer Organization and Design: The Hardware/Software Interface*.)

Our intent has always been to focus on material that is not available in equivalent form from other sources, so we continue to emphasize advanced content wherever possible. Indeed, there are several systems here whose descriptions cannot be found in the literature.

An Overview of the Content

Chapter 1 covers the basic quantitative principles of computer design and performance measurement. It also addresses the role of technology and the factors affecting the cost of computer systems. It concludes by examining performance and price-performance measurements of processors designed for the desktop, server, and embedded markets, as well as considering the power efficiency of embedded processors.

Chapter 2 covers instruction set design principles and examples. In addition to giving quantitative data on instruction set usage based on the SPEC2000 benchmarks, it describes the MIPS64 architecture used throughout the book. New to this edition are principles of digital signal processor architectures, including common features and measurements. It describes the structure of modern compilers and how that affects the utility of instruction sets for traditional computers, DSPs, and media extensions. It also gives the Trimedia TM5200 as a contrasting example of a media processor, offering instruction mixes for both it and MIPS. Appendices C to G extend this chapter by describing a dozen other popular instruction sets.

Chapters 3 and 4 cover the exploitation of instruction-level parallelism in high-performance processors, including superscalar execution, branch prediction, speculation, dynamic scheduling, and the relevant compiler technology. These topics have grown so much that, even with the creation of a 100-page appendix based on Chapter 3 of the second edition, we still needed two chapters to cover the advanced material. Chapter 3 of this edition focuses on hardware-based approaches to exploiting instruction-level parallelism, while Chapter 4 focuses on more static approaches that rely on more sophisticated compiler technology. The Intel Pentium series is used as the major example in Chapter 3, while Chapter 4 examines the IA-64 architecture and its first implementation in Itanium.

Chapter 5 starts with an introductory review of cache principles. It then reorganizes the optimizations in memory hierarchy design to what are the major challenges today. In addition to real-world examples from traditional computers such as the Alpha 21264, AMD Athlon, and Intel Pentium III and 4, it describes the memory hierarchy of the Emotion Engine in the Sony Playstation-2 video game

and the Sun Fire 6800 server with its UltraSPARC III processor. This edition describes the techniques of the bandwidth-optimized DRAM chips such as RAMBUS, and comments on their cost-performance. It also includes cache performance of multimedia and server applications in addition to the SPEC2000 benchmarks for the desktop.

Chapter 6 discusses multiprocessor systems, focusing on shared-memory architectures. The chapter begins by examining the properties of different application domains with thread-level parallelism. It then explores symmetric and distributed memory architectures, examining both organizational principles and performance. Topics in synchronization, memory consistency models, and multithreading (including simultaneous multithreading) complete the foundational chapters. Sun's Wildfire design, which uses a distributed memory architecture to extend the reach of a symmetric approach, is discussed and analyzed.

Chapter 7, "Storage Systems," saw a surprising amount of revision. There is an expansion of reliability and availability, a tutorial on RAID, availability benchmarks, and rarely found failure statistics of real systems. It continues to provide an introduction to queuing theory and I/O performance benchmarks. It extends the description of traditional buses with embedded and server buses. The five design examples in later sections evolve an I/O system through increasingly realistic performance assumptions, plus an evaluation of the mean time to failure. EMC supplies the examples that put it all together, which is the first time these systems have been documented publicly. The anatomy of a digital camera offers an embedded perspective on storage systems, and the historical perspective includes a ringside view of the development and popularity of RAID.

A goal of Chapter 8 is to provide an introduction to networks from the computer architecture point of view. Since this field is vast and quickly moving, the emphasis here is on an introduction to the terminology and principles. It starts with providing a common framework for the design principles in local area networks, storage area networks, and wide area networks, concluding with a description of the technology of the Internet. The second part of Chapter 8 is an in-depth exploration of clusters and the pros and cons of the use of clusters in both scientific computing and database applications. There is a detailed evaluation of the cost-performance of clusters, including the cost of machine room space and network bandwidth. The first description of the cluster used to provide the popular Google search engine puts this chapter together.

This brings us to Appendices A through I. Appendix A is a tutorial on basic pipelining concepts. Readers relatively new to pipelining should read this appendix before Chapters 3 and 4. As mentioned earlier, Appendix B contains solutions to selected exercises. Given the ubiquity of the Web today, the remaining appendices are online, which allows us to add relevant information without increasing the weight or cost of the book. Appendix C updates the second edition RISC appendix, describing 64-bit versions of Alpha, MIPS, PowerPC, and SPARC and their multimedia extensions. Also included in this appendix are popular embedded instruction sets: ARM, Thumb, SuperH, MIPS16, and Mitsubishi M32R. Appendix D describes the 80x86 architecture. Since we have no page budget for

the online appendices, we include two architectures of more historical interest: the VAX (Appendix E) and IBM 360/370 (Appendix F). Appendix G includes an updated description of vector processors. Finally, Appendix H describes computer arithmetic, and Appendix I describes implementing coherence protocols.

In summary, about 70% of the pages are new to this edition. The third edition is also about 10% longer than the first if we don't include the online appendices, and about 30% longer if we do.

Navigating the text

There is no single best order in which to approach these chapters. We wrote the text so that it can be covered in several ways, the only real restriction being that some chapters should be read in sequence, namely, Chapters 2, 3, and 4 (pipelining) and Chapters 7 and 8 (storage systems, interconnection networks, and clusters). Readers should start with Chapter 1 and should read Chapter 5 (memory hierarchy design) before Chapter 6 (multiprocessors). Appendices C, D, E, F, and H should be read after Chapter 2. If Appendix A is going to be read, it should be read before Chapters 3 and 4. Appendix G is an interesting contrast to the ideas in Chapters 3 and 4.

Despite the many ways to read this book, we expect two primary paths:

1. *Inside out:* The philosophy of this choice is that processor design is still the cornerstone of computer architecture, and the nonprocessor topics are covered as time permits. Start with Chapter 1, then inside the processor (Chapters 2, 3, 4), then memory hierarchy (5), followed by multiprocessors (6), storage (7), and finish with networks and clusters (8).
2. *Outside in:* The philosophy of this path is that the most interesting challenges in computer architecture today are outside the processor, and that processor internals are covered as time permits. Start again with Chapter 1, then memory hierarchy (5), followed by multiprocessors (6), storage (7), networks and clusters (8), and conclude with instruction sets and pipelining (Chapters 2, 3, 4).

Chapter Structure and Exercises

The material we have selected has been stretched upon a consistent framework that is followed in each chapter. We start by explaining the ideas of a chapter. These ideas are followed by a "Crosscutting Issues" section, a feature that shows how the ideas covered in one chapter interact with those given in other chapters. This is followed by a "Putting It All Together" section that ties these ideas together by showing how they are used in a real machine. This is followed by one or two sections titled "Another View," a new feature for the third edition that gives a real-world example from the embedded or server space.

Next in the sequence is "Fallacies and Pitfalls," which lets readers learn from the mistakes of others. We show examples of common misunderstandings and

architectural traps that are difficult to avoid even when you know they are lying in wait for you. Each chapter ends with a "Concluding Remarks" section, followed by a "Historical Perspective and References" section that attempts to give proper credit for the ideas in the chapter and a sense of the history surrounding the inventions. We like to think of this as presenting the human drama of computer design. It also supplies references that the student of architecture may want to pursue. If you have time, we recommend reading some of the classic papers in the field that are mentioned in these sections. It is both enjoyable and educational to hear the ideas directly from the creators. The "Fallacies and Pitfalls" and "Historical Perspective" sections are two of the most popular sections of prior editions.

Each chapter ends with exercises, over 200 in total, which vary from one-minute reviews to term projects. Brackets for each exercise (<chapter.section>) indicate the text sections of primary relevance to answering the question. We hope this helps readers to avoid exercises for which they haven't read the corresponding section, in addition to providing the source for review. Note that we provide solutions to selected exercises in Appendix B, which we indicate with the ✪ symbol. We also rate the exercises, estimating the amount of time a problem might take:

[10] Less than 5 minutes (to read and understand)

[20] 15 to 20 minutes for a full answer

[25] 1 hour for a full written answer

[30] Short programming project: less than 1 full day of programming

[40] Significant programming project: 2 weeks of elapsed time

[50] Term project (2 to 4 weeks by a team)

[Discussion] Topic for discussion with others

Supplements

An instructor's manual with fully worked-out solutions to the exercises in the book is available from the publisher to official instructors teaching from this book.

The numerous appendices are available to readers at the Morgan Kaufmann home page on the Web at *www.mkp.com*. Since we are now using a standard instruction set architecture, we no longer need supply special software. The Web page includes pointers to simulators, compilers, assemblers, and so on for the MIPS architecture. This page also contains a list of errata, eps versions of the numbered figures in the book, and pointers to related material that readers may enjoy. In response to your continued support, the publisher will add new materials and establish links to other sites on a regular basis.

Helping Improve this Book

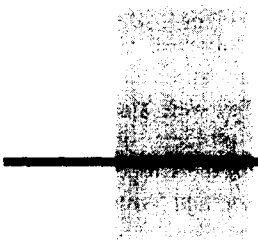
Finally, it is possible to make money while reading this book. (Talk about cost-performance!) If you read the Acknowledgments that follow, you will see that we went to great lengths to correct mistakes. Since a book goes through many printings, we have the opportunity to make even more corrections. If you uncover any remaining resilient bugs, please contact the publisher by electronic mail (*ca3bugs@mkp.com*). The first reader to report an error with a fix that we incorporate in a future printing will be rewarded with a $1.00 bounty. Please check the errata sheet on the home page (*www.mkp.com*) to see if the bug has already been reported. We process the bugs and send the checks about once a year or so, so please be patient.

We welcome general comments to the text and invite you to send them to a separate email address at *ca3comments@mkp.com*.

Concluding Remarks

Once again this book is a true co-authorship, with each of us writing half the chapters and an equal share of the appendices. We can't imagine how long it would have taken without someone else doing half the work, offering inspiration when the task seemed hopeless, providing the key insight to explain a difficult concept, supplying reviews over the weekend of 100-page chapters, and commiserating when the weight of our other obligations made it hard to pick up the pen. (These obligations have escalated exponentially with the number of editions, as one of us is now in charge of a university.) Thus, once again we share equally the blame for what you are about to read.

John Hennessy ■ *David Patterson*



Acknowledgments

Although this is only the third edition of this book, we have actually created seven different versions of the text: three versions of the first edition (alpha, beta, and final), two versions of this edition (beta and final), and two versions of the third edition (beta and final). Along the way, we have received help from hundreds of reviewers and users. Each of these people has helped make this book better. Thus, we have chosen to list all of the people who have made contributions to some version of this book, as well as those who have been the leading bug reporters.

Contributors to the Third Edition

As you can see from the preface, this edition had conceptual changes as well as updates to the material. The insight of those changes came from looking at the comments of people who evaluated an initial proposal in the summer of 1999: Mark D. Hill and Guri Sohi, University of Wisconsin at Madison; James R. Larus, Microsoft Research; Mateo Valero, Universidad Politécnica de Cataluña, Barcelona; and Maurice Wilkes, who needs no affiliation. Reflection on these comments led to a revised proposal the following winter, which led to a more enthusiastic response from George Adams, Purdue University; Mark D. Hill and Jim Smith, University of Wisconsin at Madison; Josh Fisher, Hewlett-Packard Labs, Cambridge; Kai Li, Princeton University; Kourosh Gharachorloo, Compaq Western Research Laboratory; and James R. Larus, Microsoft Research. Don Knuth provided us with a detailed review from his thorough reading of the second edition; he uncovered a number of bugs and ambiguities, a few of which had escaped detection since the first edition!

With an outline in place, we received help on the ideas within some chapters, new real-world examples to populate "Putting It All Together" and "Another View," and the numbers to drive scores of tables and figures in this book. Note to potential authors: the downside of a quantitative approach is that it takes CPU centuries to update the figures each edition, and real-world examples rely on friends and strangers to understand the anatomy of systems that have never been described publicly. Hence we thank each group by chapter.

Chapter 2 relied on measurements of SPEC2000 benchmarks on real machines to collect the instruction set statistics. Thanks go to Dave Albonesi, Alper Buyuktosunoglu, Lei Chen, Wael El-Essawy, and Greg Semeraro, all of the University of Rochester, for collecting this vast amount of data. They were aided by John Henning of Compaq, who helped them resolve compiler problems and interpret the results. Kees A. Vissers and Sebastian Mirolo of Trimedia Corporation supplied the instruction mix statistics on their media processor, and Sarita Adve and Chris Hughes of the University of Illinois at Urbana-Champaign supplied cache measurements on the media benchmarks. Finally, Jeff Bier of BDTI supplied the history of DSPs. Without the collective efforts of all these people, Chapter 2 would be antiquated.

The detailed examination of the Pentium III data path in Chapter 3 is based on work by Bhandarkar and Ding. Dileep Bhandarkar provided the original data, which greatly eased the creation of the plots in the chapter. Kees A. Vissers of Trimedia Corporation graciously updated the architectural description we had written of the Trimedia architecture in Chapter 4 so that it describes the latest version.

We expected Chapter 5 to be one of our easier chapters, but out-of-order CPUs meant we needed to move from miss rates and average memory access times to misses per instruction and performance to evaluate memory hierarchies. Susan Eggers of the University of Washington and Mark Hill of the University of Wisconsin at Madison helped us see the light. Mark Hill and Jason F. Cantin then proceeded to collect the vast information on SPEC2000 for these new efforts. Resources were made available and computing resources provided by Condor, Midship (NSF CDA-9623632), and Multifacet (NSF EIA-9971256) projects and by Compaq Computer Corp. through John Kowaleski. Richard Kessler and Zarka Cvetanovic supplied measurements and hitherto unknown details of the Alpha 21264, as did Gary Lauterbach of Sun Microsystems for the UltraSPARC III and the Sun Fire 6800.

The performance analysis data on commercial workloads in Chapter 6 came from the work of L. Barroso, K. Gharachorloo, and E. Bugnion at Compaq's Western Research Lab. They supplied the data in raw form, which made generation of the plots much easier. Advice and a review of the material on simultaneous multithreading (SMT) came from Susan Eggers of the University of Washington. The data on SMT performance came from the SMT research group at the University of Washington, and special simulations to obtain the data were performed by Steve Swanson and Luke McDowell. Finally, Lisa Noordergraaf of Sun Microsystems provided the raw data on Wildfire performance from her paper on the Wildfire prototype.

Chapter 7 was shaped by discussions with Howard Alt, before he resigned as CTO of Sun Microsystems Storage Division to create a start-up company, and by Ric Wheeler of EMC. We are indebted to David Black, Dan Lambright, and Ric Wheeler of EMC, as well as to their employer, for supplying the description and measurements of the EMC Symmetrix and Celerra computers. Mike Dahlin of the University of Texas, Austin, supplied data on the history of cost and capacity

of disks, and John Best of IBM explained some of the quirks of disk technology and reliability. Aaron Brown of U.C. Berkeley supplied the availability benchmark results, and Patty Enriquez of Mills supplied the update on FCC failure data. A final thanks goes to Alexander Thomasian of the New Jersey Institute of Technology for guidance on queuing theory.

Chapter 8 was shaped by discussions with Alan Mainwaring of Intel Berkeley Research Labs and Rich Martin of Rutgers. We are indebted to Urs Hoelzle of Google, and once again his employer, for allowing us to describe their cluster. Vern Paxson of the Center for Internet Research supplied Internet measurements and advice on reading material in networking.

The vector appendix (G) was revised by Krste Asanovic of the Massachusetts Institute of Technology, and the floating-point appendix (H) was written originally by David Goldberg of Xerox PARC.

In addition to updating the contents of the chapters, we needed to update the exercises. George Adams of Purdue University generously enhanced and revised the exercises and provided solutions for most of the chapters. Others who contributed exercises include Todd M. Bezenek of the University of Wisconsin at Madison (in remembrance of his grandmother Ethel Eshom); Ethan L. Miller, University of California, Santa Cruz; Brandon Schwartz, University of Wisconsin at Madison; and Parthasarathy Ranganathan, Compaq Western Research Laboratory.

For the third time we made substantial changes to the book as a result of class testing. The class test site institutions and instructors for Fall 2000 were

Sarita Adve, University of Illinois at Urbana-Champaign

Doug Burger, University of Texas at Austin

David Patterson, University of California at Berkeley

Arnold L. Rosenberg, University of Massachusetts at Amherst

For Spring 2001 the group included

George Adams, Purdue University

Sarita Adve, University of Illinois at Urbana-Champaign

Lori Liebrock, University of Alaska, Fairbanks

Mikko Lipasti, University of Wisconsin at Madison

Steve Reinhardt, University of Michigan

Hank Walker, Texas A&M

We thank the students and instructors for helping us improve this edition.

In addition to class testing, we had many reviewers give us feedback on the first manuscript, which also guided our revisions:

Krste Asanovic, Massachusetts Institute of Technology

James Goodman, University of Wisconsin at Madison

David Harris, Harvey Mudd College

Norm Jouppi, Compaq

Jim Larus, Microsoft Research

David Kaeli, Northeastern University

David Nagle, Carnegie Mellon University

Emilio Salgueiro, Unysis

Guri Sohi, University of Wisconsin at Madison

Shlomo Weiss, Tel Aviv University

Finally, a special thanks to Mark Smotherman of Clemson University, who gave a final technical reading of our revised manuscript just before it was sent to the publisher. Mark found numerous bugs and ambiguities, and the book is much cleaner as a result.

This book could not have been published without a publisher, of course. We selected Morgan Kaufmann Publishers when we wrote the first edition, and we have not regretted that decision. We are not aware of another publisher who could have kept pace with such a rigorous schedule and we thank all the Morgan Kaufmann staff for their efforts and support. For this third edition, we particularly want to thank Cheri Palmer, our production editor, who ran the show, and Alyson Day, who coordinated all the reviews. Courtney Garnaas also helped with early versions of the manuscript. Our warmest thanks to our editor, Denise Penrose, for her inspiration and perseverance in our continuing writing saga.

We must also thank our university staff, Margaret Rowland and Willa Walker, for countless express mailings, as well as for holding down the fort at Stanford and Berkeley while we worked on the book.

Our final thanks go to our families for their suffering through long nights and early mornings of reading, thinking, and typing.

Contributors to the Second Edition

The development process for this edition was equal in extent to that of the first edition, and we have many people to thank for their contributions.

Before we started extensive work on this edition, we received valuable comments on an outline of our ideas from a number of people: Jim Archibald of Brigham Young University, Jean-Loup Baer of the University of Washington, Paul Barr of Northeastern University, Barry Fagin of Dartmouth, Joel Ferguson of the University of California at Santa Cruz, David Goldberg of the Xerox Palo Alto Research Center, Mark Hill of the University of Wisconsin at Madison, Jim Larus of the University of Wisconsin, William Michalson of Worcester Polytechnic Institute, Richard Reid of Michigan State University, Jim Smith of the University of Wisconsin, Mark Smotherman of Clemson University, Arun Somani of the University of Washington, Thorsten von Eicken of Cornell University, and Shlomo Weiss of the University of Maryland and Tel Aviv.

Our manuscript went through several iterations during its development, and at each stage the chapters were carefully reviewed by contributors from both industry and academia. The following people reviewed one or more chapters of the beta or final manuscript: George Adams of Purdue University, Rajendra V. Boppana of the University of Texas at San Antonio, John Burger of SGI, Peter Chen of the University of Michigan, Tim Coe of Vitesse Semiconductor, Bob Colwell of Intel, Josh Fisher of Hewlett-Packard Laboratories, Rob Fowler of DIKU, Kourosh Gharachorloo of DEC, Mark Heinrich of Stanford, Mark Hill of the University of Wisconsin, Martin Hopkins of IBM, Jerry Huck of Hewlett-Packard Laboratories, Norm Jouppi of DEC, Jeff Kuskin of Stanford, Corinna Lee of the University of Toronto, Gyula Mago of the University of North Carolina, Trevor Mudge of the University of Michigan, Greg Papadapoulous of Sun, Steven Przybylski, Dan Siewiorek of Carnegie Mellon University, J. P. Singh of Princeton, Ashok Singhal, Mike Smith of Harvard University, Mark Smotherman of Clemson University, Guri Sohi of the University of Wisconsin, Evan Tick of the University of Oregon, Thorsten von Eicken of Cornell University, Roy Want of the Xerox Palo Alto Research Center, David Weaver of Sun, Mike Westall of Clemson University, and Larry Wittie of SUNY Stony Brook. Thanks also to Jorge Stolfi for inspiring the section on DIV and MOD for negative numbers in Appendix A.

As with both of our previous books, we enlisted the help of instructors, teaching assistants, and hundreds of students who used the beta in the 1994–95 academic year. And once again, we made substantial changes to the book as a result of the beta testing. The beta test site institutions and instructors were Clemson University—Gene Tagliarin, Cornell University—Anthony Reeves, Georgia Institute of Technology—Kishore Ramachandran and Ellen Witte Zegura, Purdue University—George Adams and Kaushik Roy, Stanford University—Kunle Olukotun, State University of New York at Stony Brook—Larry Wittie, University of California at Berkeley—David Patterson, University of California at Los Angeles—Dave Rennels, University of California at Santa Cruz—Anujan Varma, University of North Carolina at Chapel Hill—Gyula A. Mago, University of Oregon—Evan Tick, University of Texas at San Antonio—Rajendra V. Boppana, University of Washington—Susan Eggers, University of Wisconsin at Madison—Mark Hill.

We would like to acknowledge everyone who participated for their efforts in hunting for numerous bugs in the beta. The following people reported the most bugs: Ravi Murthy, Yung-Hsiang Lu, Chen-Chung Chang, Ajay Sreekanth, Mark Callaghan, Lewis Jordan, Lawrence Prince, Darren Staples, Chao-Huang Lin, Biswadeep Nag, Peter Ashenden, and Ronald Greenberg. We would also like to thank those who have continued to submit bugs for the first edition of the book.

In pursuing our goal of publishing the cleanest book possible, we submitted our manuscript to a final technical review, which consisted of a detailed examination of the final version. This daunting task was undertaken by the following individuals: Nikolas Gloy of Harvard University, Evan Tick of the University of Oregon, and George Adams, Allan Knies, and Thomas Willis of Purdue University. Our thanks go to them for greatly improving the accuracy of this edition.

Contributors to the First Edition

The first edition had its roots in an alpha version developed in 1988 and used in the 1988–89 school year. We thank the instructors, teaching assistants, and reviewers of the earliest version of this book, including Thomas Casavant, David Douglas, Joel Emer, Jim Goodman, Truman Joe, Roger Kieckhafer, Earl Killian, Hank Levy, Bryan Martin, Norman Matloff, David Meyer, Trevor Mudge, Todd Narter, Victor Nelson, Richard Reid, Margo Seltzer, Jim Smith, Mark Smotherman, David Wells, and Eric Williams. We also extend a special thanks to the students at Stanford and Berkeley who endured our first attempts at creating this book.

Following the alpha, a beta version of the text was created and used in the 1989–90 school year. Many people reviewed portions of the beta version, including Paul Barr, Bill Dally, Susan Eggers, Jim Goodman, Mark Hill, Hank Levy, David Meyer, James Mooney, Joseph Pfeiffer, and Larry Wittie. Input and material for individual chapters was sought from a number of academic and industrial experts, including Tom Adams, Anant Agarwal, Mitch Alsup, Dave Anderson, David Bailey, Andy Bechtolsheim, Fred Berkowitz, Dileep Bhandarkar, Mark Birman, David Boggs, Jim Brady, Forrest Brewer, Paul Carrick, Pete Chen, Nhan Chu, Bob Cmelik, John Crawford, Merrick Darley, John DeRosa, Lloyd Dickman, Milos Ercegovac, Robert Garner, Garth Gibson, Ben Hao, Danny Hillis, David Hodges, David Hough, Ed Hudson, Mark Johnson, Norm Jouppi, William Kahan, Randy Katz, Ed Kelly, Les Kohn, Corinna Lee, Ruby Lee, Don Lewine, Ken Lutz, Al Marston, John Mashey, Steven Przybylski, Chris Rowen, Bill Shannon, Behrooz Shirazi, Robert Shomler, Jim Slager, Charles Stapper, Peter Stoll, Bob Supnik, Paul Taysom, Shreekant Thakkar, David Weaver, and Richard Zimmermann. We appreciate the willingness of our colleagues to lend their expertise and advice.

The beta test site institutions and instructors for the first edition were Carnegie Mellon University—Daniel Siewiorek, Clemson University—Mark Smotherman, Cornell University—Keshav Pingali, Pennsylvania State University—Mary Jane Irwin and Bob Owens, San Francisco State University—Vojin Oklobdzija, Southeast Missouri State University—Anthony Duben, Southern Methodist University—Behrooz Shirazi, Stanford University—John Hennessy, State University of New York at Stony Brook—Larry Wittie, University of California at Berkeley—Vojin Oklobdzija, University of California at Los Angeles—David Rennels, University of California at Santa Cruz—Daniel Helman, University of Nebraska—Roger Kieckhafer, University of North Carolina at Chapel Hill—Akhilesh Tyagi, University of Texas at Austin—Joseph Rameh, University of Waterloo—Bruno Preiss, University of Wisconsin at Madison—Mark Hill, Washington University (St. Louis)—Mark Franklin.

Special mention should be given to Daniel Helman, Mark Hill, Mark Smotherman, and Larry Wittie, who were especially generous with their feedback. The classes at SUNY Stony Brook, Carnegie Mellon, Stanford, Clemson, and Wisconsin supplied us with the greatest number of bug discoveries in the beta version