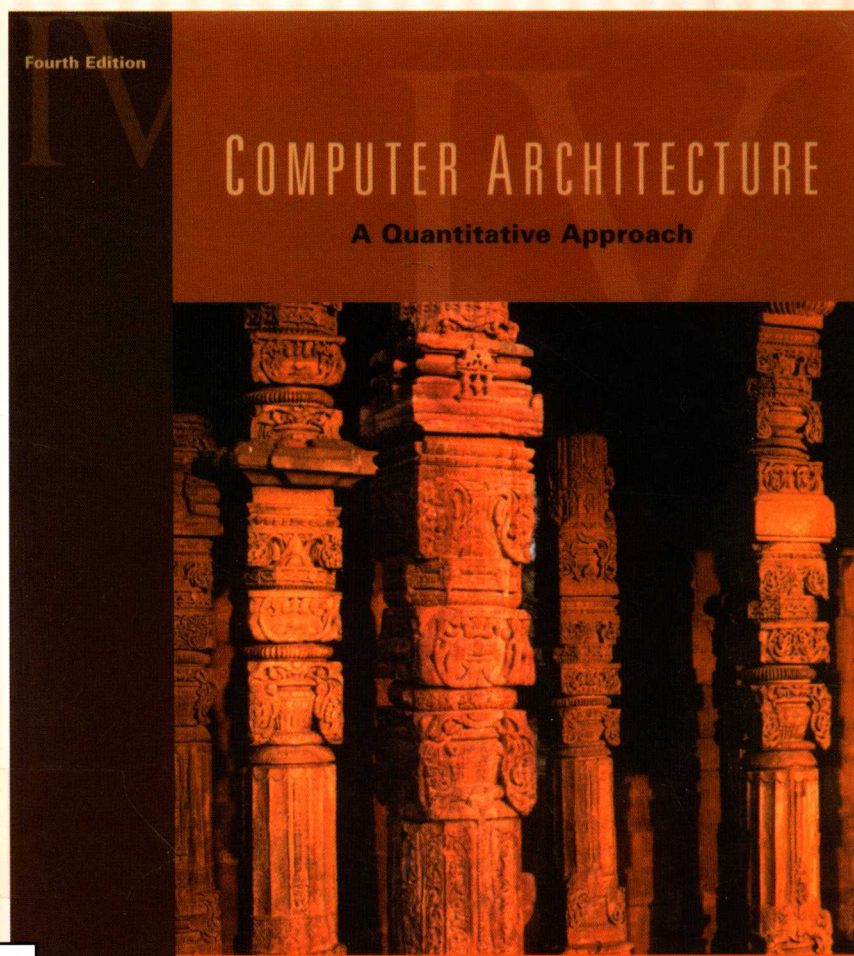


计算机体系结构 量化研究方法

(英文版 · 第4版)



MK
MOORE KOPPEL

John L. Hennessy and David A. Patterson

(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

0409344

经典原版书库

计算机体系结构 量化研究方法

(英文版·第4版)

Computer Architecture
A Quantitative Approach
(Fourth Edition)

John L. Hennessy

斯坦福大学

(美)

David A. Patterson

加州大学伯克利分校

著



机械工业出版社
China Machine Press

John L. Hennessy and David A. Patterson: Computer Architecture: A Quantitative Approach, Fourth Edition (ISBN 13: 978-0-12-370490-0 ISBN 10: 0-12-370490-1).

Original English language edition copyright ©1990, 1996, 2003, 2007 by Elsevier Inc. All rights reserved.

Authorized English language reprint edition published by the Proprietor.

ISBN 13: 978-981-259-693-2 ISBN 10: 981-259-693-3

Copyright © 1990, 1996, 2003, 2007 by Elsevier (Singapore) Pte Ltd.

Printed in China by China Machine Press under special arrangement with Elsevier (Singapore) Pte Ltd. This edition is authorized for sale in China only, excluding Hong Kong SAR and Taiwan.

Unauthorized export of this edition is a violation of the Copyright Act. Violation of this Law is subject to Civil and Criminal Penalties.

本书英文影印版由Elsevier (Singapore) Pte Ltd.授权机械工业出版社在中国大陆境内独家发行。本版仅限在中国境内（不包括香港特别行政区及台湾地区）出版及标价销售。未经许可之出口，视为违反著作权法，将受法律之制裁。

版权所有，侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号：图字：01-2006-6516

图书在版编目 (CIP) 数据

计算机体系结构：量化研究方法（英文版·第4版）/（美）亨尼西（Hennessy, J. L.）等著. —北京：机械工业出版社，2007.1

（经典原版书库）

书名原文：Computer Architecture: A Quantitative Approach, Fourth Edition
ISBN 7-111-20378-X

I. 计… II. 亨… III. 计算机体系结构-英文 IV. TP303

中国版本图书馆CIP数据核字（2006）第138892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

北京牛山世兴印刷厂印刷·新华书店北京发行所发行

2007年1月第1版第1次印刷

170mm × 242mm · 44.25印张

定价：78.00元（附光盘）

凡购本书，如有倒页、脱页、缺页，由本社发行部调换
本社购书热线：（010）68326294

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，还揭橥了学术的源变，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、从业人员较少的现状下，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图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall, Addison-Wesley, McGraw-Hill, Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum, Stroustrup, Kernighan, Jim Gray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、研究及度藏。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，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步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步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's

Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业的教学度身订造的。其中许多教材均已为M. I. T., Stanford, U.C. Berkeley, C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

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，但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。教材的出版只是我们的后续服务的起点。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，我们的联系方式如下：

电子邮件：hzjsj@hzbook.com

联系电话：(010) 68995264

联系地址：北京市西城区百万庄南街1号

邮政编码：100037

专家指导委员会

(按姓氏笔画顺序)

尤晋元	王 珊	冯博琴	史忠植	史美林
石教英	吕 建	孙玉芳	吴世忠	吴时霖
张立昂	李伟琴	李师贤	李建中	杨冬青
邵维忠	陆丽娜	陆鑫达	陈向群	周伯生
周克定	周傲英	孟小峰	岳丽华	范 明
郑国梁	施伯乐	钟玉琢	唐世渭	袁崇义
高传善	梅 宏	程 旭	程时端	谢希仁
裘宗燕	戴 葵			

In Praise of *Computer Architecture: A Quantitative Approach* Fourth Edition

“The multiprocessor is here and it can no longer be avoided. As we bid farewell to single-core processors and move into the chip multiprocessing age, it is great timing for a new edition of Hennessy and Patterson’s classic. Few books have had as significant an impact on the way their discipline is taught, and the current edition will ensure its place at the top for some time to come.”

—Luiz André Barroso, Google Inc.

“What do the following have in common: Beatles’ tunes, HP calculators, chocolate chip cookies, and *Computer Architecture*? They are all classics that have stood the test of time.”

—Robert P. Colwell, Intel lead architect

“Not only does the book provide an authoritative reference on the concepts that all computer architects should be familiar with, but it is also a good starting point for investigations into emerging areas in the field.”

—Krisztián Flautner, ARM Ltd.

“The best keeps getting better! This new edition is updated and very relevant to the key issues in computer architecture today. Plus, its new exercise paradigm is much more useful for both students and instructors.”

—Norman P. Jouppi, HP Labs

“*Computer Architecture* builds on fundamentals that yielded the RISC revolution, including the enablers for CISC translation. Now, in this new edition, it clearly explains and gives insight into the latest microarchitecture techniques needed for the new generation of multithreaded multicore processors.”

—Marc Tremblay, Fellow & VP, Chief Architect, Sun Microsystems

“This is a great textbook on all key accounts: pedagogically superb in exposing the ideas and techniques that define the art of computer organization and design, stimulating to read, and comprehensive in its coverage of topics. The first edition set a standard of excellence and relevance; this latest edition does it again.”

—Miloš Ercegovac, UCLA

“They’ve done it again. Hennessy and Patterson emphatically demonstrate why they are the doyens of this deep and shifting field. Fallacy: Computer architecture isn’t an essential subject in the information age. Pitfall: You don’t need the 4th edition of *Computer Architecture*.”

—Michael D. Smith, Harvard University

“Hennessy and Patterson have done it again! The 4th edition is a classic encore that has been adapted beautifully to meet the rapidly changing constraints of ‘late-CMOS-era’ technology. The detailed case studies of real processor products are especially educational, and the text reads so smoothly that it is difficult to put down. This book is a must-read for students and professionals alike!”

—Pradip Bose, IBM

“This latest edition of *Computer Architecture* is sure to provide students with the architectural framework and foundation they need to become influential architects of the future.”

—Ravishankar Iyer, Intel Corp.

“As technology has advanced, and design opportunities and constraints have changed, so has this book. The 4th edition continues the tradition of presenting the latest in innovations with commercial impact, alongside the foundational concepts: advanced processor and memory system design techniques, multithreading and chip multiprocessors, storage systems, virtual machines, and other concepts. This book is an excellent resource for anybody interested in learning the architectural concepts underlying real commercial products.”

—Gurindar Sohi, University of Wisconsin–Madison

“I am very happy to have my students study computer architecture using this fantastic book and am a little jealous for not having written it myself.”

—Mateo Valero, UPC, Barcelona

“Hennessy and Patterson continue to evolve their teaching methods with the changing landscape of computer system design. Students gain unique insight into the factors influencing the shape of computer architecture design and the potential research directions in the computer systems field.”

—Dan Connors, University of Colorado at Boulder

“With this revision, *Computer Architecture* will remain a must-read for all computer architecture students in the coming decade.”

—Wen-mei Hwu, University of Illinois at Urbana–Champaign

“The 4th edition of *Computer Architecture* continues in the tradition of providing a relevant and cutting edge approach that appeals to students, researchers, and designers of computer systems. The lessons that this new edition teaches will continue to be as relevant as ever for its readers.”

—David Brooks, Harvard University

“With the 4th edition, Hennessy and Patterson have shaped *Computer Architecture* back to the lean focus that made the 1st edition an instant classic.”

—Mark D. Hill, University of Wisconsin–Madison

Foreword

by Fred Weber, President and CEO of MetaRAM, Inc.

I am honored and privileged to write the foreword for the fourth edition of this most important book in computer architecture. In the first edition, Gordon Bell, my first industry mentor, predicted the book's central position as the definitive text for computer architecture and design. He was right. I clearly remember the excitement generated by the introduction of this work. Rereading it now, with significant extensions added in the three new editions, has been a pleasure all over again. No other work in computer architecture—frankly, no other work I have read in any field—so quickly and effortlessly takes the reader from ignorance to a breadth and depth of knowledge.

This book is dense in facts and figures, in rules of thumb and theories, in examples and descriptions. It is stuffed with acronyms, technologies, trends, formulas, illustrations, and tables. And, this is thoroughly appropriate for a work on architecture. The architect's role is not that of a scientist or inventor who will deeply study a particular phenomenon and create new basic materials or techniques. Nor is the architect the craftsman who masters the handling of tools to craft the finest details. The architect's role is to combine a thorough understanding of the state of the art of what is possible, a thorough understanding of the historical and current styles of what is desirable, a sense of design to conceive a harmonious total system, and the confidence and energy to marshal this knowledge and available resources to go out and get something built. To accomplish this, the architect needs a tremendous density of information with an in-depth understanding of the fundamentals and a quantitative approach to ground his thinking. That is exactly what this book delivers.

As computer architecture has evolved—from a world of mainframes, minicomputers, and microprocessors, to a world dominated by microprocessors, and now into a world where microprocessors themselves are encompassing all the complexity of mainframe computers—Hennessy and Patterson have updated their book appropriately. The first edition showcased the IBM 360, DEC VAX, and Intel 80x86, each the pinnacle of its class of computer, and helped introduce the world to RISC architecture. The later editions focused on the details of the 80x86 and RISC processors, which had come to dominate the landscape. This latest edition expands the coverage of threading and multiprocessing, virtualization

and memory hierarchy, and storage systems, giving the reader context appropriate to today's most important directions and setting the stage for the next decade of design. It highlights the AMD Opteron and SUN Niagara as the best examples of the x86 and SPARC (RISC) architectures brought into the new world of multiprocessing and system-on-a-chip architecture, thus grounding the art and science in real-world commercial examples.

The first chapter, in less than 60 pages, introduces the reader to the taxonomies of computer design and the basic concerns of computer architecture, gives an overview of the technology trends that drive the industry, and lays out a quantitative approach to using all this information in the art of computer design. The next two chapters focus on traditional CPU design and give a strong grounding in the possibilities and limits in this core area. The final three chapters build out an understanding of system issues with multiprocessing, memory hierarchy, and storage. Knowledge of these areas has always been of critical importance to the computer architect. In this era of system-on-a-chip designs, it is essential for every CPU architect. Finally the appendices provide a great depth of understanding by working through specific examples in great detail.

In design it is important to look at both the forest and the trees and to move easily between these views. As you work through this book you will find plenty of both. The result of great architecture, whether in computer design, building design or textbook design, is to take the customer's requirements and desires and return a design that causes that customer to say, "Wow, I didn't know that was possible." This book succeeds on that measure and will, I hope, give you as much pleasure and value as it has me.



Preface

Why We Wrote This Book

Through four 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.

This Edition

The fourth edition of *Computer Architecture: A Quantitative Approach* may be the most significant since the first edition. Shortly before we started this revision, Intel announced that it was joining IBM and Sun in relying on multiple processors or cores per chip for high-performance designs. As the first figure in the book documents, after 16 years of doubling performance every 18 months, sin-

gle-processor performance improvement has dropped to modest annual improvements. This fork in the computer architecture road means that for the first time in history, no one is building a much faster sequential processor. If you want your program to run significantly faster, say, to justify the addition of new features, you're going to have to parallelize your program.

Hence, after three editions focused primarily on higher performance by exploiting instruction-level parallelism (ILP), an equal focus of this edition is thread-level parallelism (TLP) and data-level parallelism (DLP). While earlier editions had material on TLP and DLP in big multiprocessor servers, now TLP and DLP are relevant for single-chip multicores. This historic shift led us to change the order of the chapters: the chapter on multiple processors was the sixth chapter in the last edition, but is now the fourth chapter of this edition.

The changing technology has also motivated us to move some of the content from later chapters into the first chapter. Because technologists predict much higher hard and soft error rates as the industry moves to semiconductor processes with feature sizes 65 nm or smaller, we decided to move the basics of dependability from Chapter 7 in the third edition into Chapter 1. As power has become the dominant factor in determining how much you can place on a chip, we also beefed up the coverage of power in Chapter 1. Of course, the content and examples in all chapters were updated, as we discuss below.

In addition to technological sea changes that have shifted the contents of this edition, we have taken a new approach to the exercises in this edition. It is surprisingly difficult and time-consuming to create interesting, accurate, and unambiguous exercises that evenly test the material throughout a chapter. Alas, the Web has reduced the half-life of exercises to a few months. Rather than working out an assignment, a student can search the Web to find answers not long after a book is published. Hence, a tremendous amount of hard work quickly becomes unusable, and instructors are denied the opportunity to test what students have learned.

To help mitigate this problem, in this edition we are trying two new ideas. First, we recruited experts from academia and industry on each topic to write the exercises. This means some of the best people in each field are helping us to create interesting ways to explore the key concepts in each chapter and test the reader's understanding of that material. Second, each group of exercises is organized around a set of case studies. Our hope is that the quantitative example in each case study will remain interesting over the years, robust and detailed enough to allow instructors the opportunity to easily create their own new exercises, should they choose to do so. Key, however, is that each year we will continue to release new exercise sets for each of the case studies. These new exercises will have critical changes in some parameters so that answers to old exercises will no longer apply.

Another significant change is that we followed the lead of the third edition of *Computer Organization and Design (COD)* by slimming the text to include the material that almost all readers will want to see and moving the appendices that

some will see as optional or as reference material onto a companion CD. There were many reasons for this change:

1. Students complained about the size of the book, which had expanded from 594 pages in the chapters plus 160 pages of appendices in the first edition to 760 chapter pages plus 223 appendix pages in the second edition and then to 883 chapter pages plus 209 pages in the paper appendices and 245 pages in online appendices. At this rate, the fourth edition would have exceeded 1500 pages (both on paper and online)!
2. Similarly, instructors were concerned about having too much material to cover in a single course.
3. As was the case for *COD*, by including a CD with material moved out of the text, readers could have quick access to all the material, regardless of their ability to access Elsevier's Web site. Hence, the current edition's appendices will always be available to the reader even after future editions appear.
4. This flexibility allowed us to move review material on pipelining, instruction sets, and memory hierarchy from the chapters and into Appendices A, B, and C. The advantage to instructors and readers is that they can go over the review material much more quickly and then spend more time on the advanced topics in Chapters 2, 3, and 5. It also allowed us to move the discussion of some topics that are important but are not core course topics into appendices on the CD. Result: the material is available, but the printed book is shorter. In this edition we have 6 chapters, none of which is longer than 80 pages, while in the last edition we had 8 chapters, with the longest chapter weighing in at 127 pages.
5. This package of a slimmer core print text plus a CD is far less expensive to manufacture than the previous editions, allowing our publisher to significantly lower the list price of the book. With this pricing scheme, there is no need for a separate international student edition for European readers.

Yet another major change from the last edition is that we have moved the embedded material introduced in the third edition into its own appendix, Appendix D. We felt that the embedded material didn't always fit with the quantitative evaluation of the rest of the material, plus it extended the length of many chapters that were already running long. We believe there are also pedagogic advantages in having all the embedded information in a single appendix.

This edition continues the tradition of using real-world examples to demonstrate the ideas, and the "Putting It All Together" sections are brand new; in fact, some were announced after our book was sent to the printer. The "Putting It All Together" sections of this edition include the pipeline organizations and memory hierarchies of the Intel Pentium 4 and AMD Opteron; the Sun T1 ("Niagara") 8-processor, 32-thread microprocessor; the latest NetApp Filer; the Internet Archive cluster; and the IBM Blue Gene/L massively parallel processor.

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 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. (Readers interested strictly in a more basic introduction to computer architecture should read *Computer Organization and Design: The Hardware/Software Interface*, third edition.)

An Overview of the Content

Chapter 1 has been beefed up in this edition. It includes formulas for static power, dynamic power, integrated circuit costs, reliability, and availability. We go into more depth than prior editions on the use of the geometric mean and the geometric standard deviation to capture the variability of the mean. Our hope is that these topics can be used through the rest of the book. In addition to the classic quantitative principles of computer design and performance measurement, the benchmark section has been upgraded to use the new SPEC2006 suite.

Our view is that the instruction set architecture is playing less of a role today than in 1990, so we moved this material to Appendix B. It still uses the MIPS64 architecture. For fans of ISAs, Appendix J covers 10 RISC architectures, the 80x86, the DEC VAX, and the IBM 360/370.

Chapters 2 and 3 cover the exploitation of instruction-level parallelism in high-performance processors, including superscalar execution, branch prediction, speculation, dynamic scheduling, and the relevant compiler technology. As mentioned earlier, Appendix A is a review of pipelining in case you need it. Chapter 3 surveys the limits of ILP. New to this edition is a quantitative evaluation of multithreading. Chapter 3 also includes a head-to-head comparison of the AMD Athlon, Intel Pentium 4, Intel Itanium 2, and IBM Power5, each of which has made separate bets on exploiting ILP and TLP. While the last edition contained a great deal on Itanium, we moved much of this material to Appendix G, indicating our view that this architecture has not lived up to the early claims.

Given the switch in the field from exploiting only ILP to an equal focus on thread- and data-level parallelism, we moved multiprocessor systems up to Chapter 4, which focuses on shared-memory architectures. The chapter begins with the performance of such an architecture. It then explores symmetric and distributed-memory architectures, examining both organizational principles and performance. Topics in synchronization and memory consistency models are

next. The example is the Sun T1 (“Niagara”), a radical design for a commercial product. It reverted to a single-instruction issue, 6-stage pipeline microarchitecture. It put 8 of these on a single chip, and each supports 4 threads. Hence, software sees 32 threads on this single, low-power chip.

As mentioned earlier, Appendix C contains an introductory review of cache principles, which is available in case you need it. This shift allows Chapter 5 to start with 11 advanced optimizations of caches. The chapter includes a new section on virtual machines, which offers advantages in protection, software management, and hardware management. The example is the AMD Opteron, giving both its cache hierarchy and the virtual memory scheme for its recently expanded 64-bit addresses.

Chapter 6, “Storage Systems,” has an expanded discussion of reliability and availability, a tutorial on RAID with a description of RAID 6 schemes, and rarely found failure statistics of real systems. It continues to provide an introduction to queuing theory and I/O performance benchmarks. Rather than go through a series of steps to build a hypothetical cluster as in the last edition, we evaluate the cost, performance, and reliability of a real cluster: the Internet Archive. The “Putting It All Together” example is the NetApp FAS6000 filer, which is based on the AMD Opteron microprocessor.

This brings us to Appendices A through L. As mentioned earlier, Appendices A and C are tutorials on basic pipelining and caching concepts. Readers relatively new to pipelining should read Appendix A before Chapters 2 and 3, and those new to caching should read Appendix C before Chapter 5.

Appendix B covers principles of ISAs, including MIPS64, and Appendix J describes 64-bit versions of Alpha, MIPS, PowerPC, and SPARC and their multimedia extensions. It also includes some classic architectures (80x86, VAX, and IBM 360/370) and popular embedded instruction sets (ARM, Thumb, SuperH, MIPS16, and Mitsubishi M32R). Appendix G is related, in that it covers architectures and compilers for VLIW ISAs.

Appendix D, updated by Thomas M. Conte, consolidates the embedded material in one place.

Appendix E, on networks, has been extensively revised by Timothy M. Pinkston and José Duato. Appendix F, updated by Krste Asanovic, includes a description of vector processors. We think these two appendices are some of the best material we know of on each topic.

Appendix H describes parallel processing applications and coherence protocols for larger-scale, shared-memory multiprocessing. Appendix I, by David Goldberg, describes computer arithmetic.

Appendix K collects the “Historical Perspective and References” from each chapter of the third edition into a single appendix. It attempts to give proper credit for the ideas in each 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. “Historical Perspective” was one of the most popular sections of prior editions.

Appendix L (available at textbooks.elsevier.com/0123704901) contains solutions to the case study exercises in the book.

Navigating the Text

There is no single best order in which to approach these chapters and appendices, except that all readers should start with Chapter 1. If you don’t want to read everything, here are some suggested sequences:

- *ILP*: Appendix A, Chapters 2 and 3, and Appendices F and G
- *Memory Hierarchy*: Appendix C and Chapters 5 and 6
- *Thread-and Data-Level Parallelism*: Chapter 4, Appendix H, and Appendix E
- *ISA*: Appendices B and J

Appendix D can be read at any time, but it might work best if read after the ISA and cache sequences. Appendix I can be read whenever arithmetic moves you.

Chapter Structure

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.

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. The “Fallacies and Pitfalls” sections is one of the most popular sections of the book. Each chapter ends with a “Concluding Remarks” section.

Case Studies with Exercises

Each chapter ends with case studies and accompanying exercises. Authored by experts in industry and academia, the case studies explore key chapter concepts and verify understanding through increasingly challenging exercises. Instructors should find the case studies sufficiently detailed and robust to allow them to create their own additional exercises.

Brackets for each exercise (<chapter.section>) indicate the text sections of primary relevance to completing the exercise. 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 the case study