HZ BOOKS
华章教育

ELSEVIER

经 典 原 版 书 库

计算机体系结构

量化研究方法

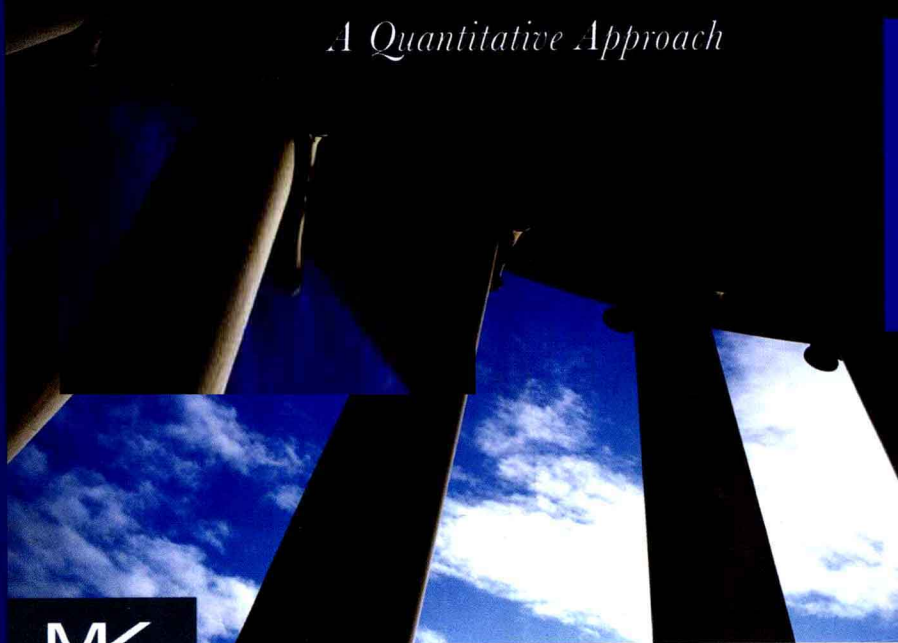
（美）John L. Hennessy 斯坦福大学
David A. Patterson 加州大学伯克利分校 等著

（英文版·第5版）

JOHN L. HENNESSY DAVID A. PATTERSON

COMPUTER ARCHITECTURE

A Quantitative Approach



机械工业出版社
China Machine Press

MK
MORGAN KAUFMANN

经 典 原 版 书 库

计算机体系结构

量化研究方法

（英文版·第5版）

Computer Architecture
A Quantitative Approach Fifth Edition

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

机械工业出版社
China Machine Press

本书版权登记号：图字：01-2011-6786

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 5 版）/（美）亨尼西（Hennessy, J. L.）等著. —北京：机械工业出版社，2011.11

（经典原版书库）

书名原文：Computer Architecture：A Quantitative Approach, Fifth Edition

ISBN 978-7-111-36458-0

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（2011）第 232733 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

北京瑞德印刷有限公司印刷

2012 年 1 月第 1 版第 1 次印刷

186mm×240mm · 53.5 印张

标准书号：ISBN 978-7-111-36458-0

定价：138.00 元

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

购书热线：（010）68326294；88379649；68995259

投稿热线：（010）88379604

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、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，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 Architecture: A Quantitative Approach* Fifth Edition

"The 5th edition of *Computer Architecture: A Quantitative Approach* continues the legacy, providing students of computer architecture with the most up-to-date information on current computing platforms, and architectural insights to help them design future systems. A highlight of the new edition is the significantly revised chapter on data-level parallelism, which demystifies GPU architectures with clear explanations using traditional computer architecture terminology."

—Krste Asanović, University of California, Berkeley

"*Computer Architecture: A Quantitative Approach* is a classic that, like fine wine, just keeps getting better. I bought my first copy as I finished up my undergraduate degree and it remains one of my most frequently referenced texts today. When the fourth edition came out, there was so much new material that I needed to get it to stay current in the field. And, as I review the fifth edition, I realize that Hennessy and Patterson have done it again. The entire text is heavily updated and Chapter 6 alone makes this new edition required reading for those wanting to really understand cloud and warehouse scale-computing. Only Hennessy and Patterson have access to the insiders at Google, Amazon, Microsoft, and other cloud computing and internet-scale application providers and there is no better coverage of this important area anywhere in the industry."

—James Hamilton, Amazon Web Services

"Hennessy and Patterson wrote the first edition of this book when graduate students built computers with 50,000 transistors. Today, warehouse-size computers contain that many servers, each consisting of dozens of independent processors and billions of transistors. The evolution of computer architecture has been rapid and relentless, but *Computer Architecture: A Quantitative Approach* has kept pace, with each edition accurately explaining and analyzing the important emerging ideas that make this field so exciting."

—James Larus, Microsoft Research

"This new edition adds a superb new chapter on data-level parallelism in vector, SIMD, and GPU architectures. It explains key architecture concepts inside mass-market GPUs, maps them to traditional terms, and compares them with vector and SIMD architectures. It's timely and relevant with the widespread shift to GPU parallel computing. *Computer Architecture: A Quantitative Approach* furthers its string of firsts in presenting comprehensive architecture coverage of significant new developments!"

—John Nickolls, NVIDIA

"The new edition of this now classic textbook highlights the ascendance of explicit parallelism (data, thread, request) by devoting a whole chapter to each type. The chapter on data parallelism is particularly illuminating: the comparison and contrast between Vector SIMD, instruction level SIMD, and GPU cuts through the jargon associated with each architecture and exposes the similarities and differences between these architectures."

—Kunle Olukotun, Stanford University

"The fifth edition of *Computer Architecture: A Quantitative Approach* explores the various parallel concepts and their respective tradeoffs. As with the previous editions, this new edition covers the latest technology trends. Two highlighted are the explosive growth of Personal Mobile Devices (PMD) and Warehouse Scale Computing (WSC)—where the focus has shifted towards a more sophisticated balance of performance and energy efficiency as compared with raw performance. These trends are fueling our demand for ever more processing capability which in turn is moving us further down the parallel path."

—Andrew N. Sloss, Consultant Engineer, ARM
Author of *ARM System Developer's Guide*

Foreword

by Luiz André Barroso, Google Inc.

The first edition of Hennessy and Patterson's *Computer Architecture: A Quantitative Approach* was released during my first year in graduate school. I belong, therefore, to that first wave of professionals who learned about our discipline using this book as a compass. Perspective being a fundamental ingredient to a useful Foreword, I find myself at a disadvantage given how much of my own views have been colored by the previous four editions of this book. Another obstacle to clear perspective is that the student-grade reverence for these two superstars of Computer Science has not yet left me, despite (or perhaps because of) having had the chance to get to know them in the years since. These disadvantages are mitigated by my having practiced this trade continuously since this book's first edition, which has given me a chance to enjoy its evolution and enduring relevance.

The last edition arrived just two years after the rampant industrial race for higher CPU clock frequency had come to its official end, with Intel cancelling its 4 GHz single-core developments and embracing multicore CPUs. Two years was plenty of time for John and Dave to present this story not as a random product line update, but as a defining computing technology inflection point of the last decade. That fourth edition had a reduced emphasis on instruction-level parallelism (ILP) in favor of added material on thread-level parallelism, something the current edition takes even further by devoting two chapters to thread- and data-level parallelism while limiting ILP discussion to a single chapter. Readers who are being introduced to new graphics processing engines will benefit especially from the new Chapter 4 which focuses on data parallelism, explaining the different but slowly converging solutions offered by multimedia extensions in general-purpose processors and increasingly programmable graphics processing units. Of notable practical relevance: If you have ever struggled with CUDA terminology check out Figure 4.24 (teaser: "Shared Memory" is really local, while "Global Memory" is closer to what you'd consider shared memory).

Even though we are still in the middle of that multicore technology shift, this edition embraces what appears to be the next major one: cloud computing. In this case, the ubiquity of Internet connectivity and the evolution of compelling Web services are bringing to the spotlight very small devices (smart phones, tablets)

and very large ones (warehouse-scale computing systems). The ARM Cortex A8, a popular CPU for smart phones, appears in Chapter 3's "Putting It All Together" section, and a whole new Chapter 6 is devoted to request- and data-level parallelism in the context of warehouse-scale computing systems. In this new chapter, John and Dave present these new massive clusters as a distinctively new class of computers—an open invitation for computer architects to help shape this emerging field. Readers will appreciate how this area has evolved in the last decade by comparing the Google cluster architecture described in the third edition with the more modern incarnation presented in this version's Chapter 6.

Return customers of this book will appreciate once again the work of two outstanding computer scientists who over their careers have perfected the art of combining an academic's principled treatment of ideas with a deep understanding of leading-edge industrial products and technologies. The authors' success in industrial interactions won't be a surprise to those who have witnessed how Dave conducts his biannual project retreats, forums meticulously crafted to extract the most out of academic–industrial collaborations. Those who recall John's entrepreneurial success with MIPS or bump into him in a Google hallway (as I occasionally do) won't be surprised by it either.

Perhaps most importantly, return and new readers alike will get their money's worth. What has made this book an enduring classic is that each edition is not an update but an extensive revision that presents the most current information and unparalleled insight into this fascinating and quickly changing field. For me, after over twenty years in this profession, it is also another opportunity to experience that student-grade admiration for two remarkable teachers.

Preface

Why We Wrote This Book

Through five 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. Like the first edition, this edition has a sharp focus on new platforms—personal mobile devices and warehouse-scale computers—and new architectures—multicore and GPUs. As much as its predecessors, this edition aims to demystify computer architecture through an emphasis on cost-performance-energy 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

We said the fourth edition of *Computer Architecture: A Quantitative Approach* may have been the most significant since the first edition due to the switch to multicore chips. The feedback we received this time was that the book had lost the sharp focus of the first edition, covering everthing equally but without emphasis and context. We're pretty sure that won't be said about the fifth edition.

We believe most of the excitement is at the extremes in size of computing, with personal mobile devices (PMDs) such as cell phones and tablets as the clients and warehouse-scale computers offering cloud computing as the server. (Observant readers may seen the hint for cloud computing on the cover.) We are struck by the common theme of these two extremes in cost, performance, and energy efficiency despite their difference in size. As a result, the running context through each chapter is computing for PMDs and for warehouse scale computers, and Chapter 6 is a brand-new chapter on the latter topic.

The other theme is parallelism in all its forms. We first idetify the two types of application-level parallelism in Chapter 1: *data-level parallelism (DLP)*, which arises because there are many data items that can be operated on at the same time, and *task-level parallelism (TLP)*, which arises because tasks of work are created that can operate independently and largely in parallel. We then explain the four architectural styles that exploit DLP and TLP: *instruction-level parallelism (ILP)* in Chapter 3; *vector architectures* and *graphic processor units (GPUs)* in Chapter 4, which is a brand-new chapter for this edition; *thread-level parallelism* in Chapter 5; and *request-level parallelism* (RLP) via warehouse-scale computers in Chapter 6, which is also a brand-new chapter for this edition. We moved memory hierarchy earlier in the book to Chapter 2, and we moved the storage systems chapter to Appendix D. We are particularly proud about Chapter 4, which contains the most detailed and clearest explanation of GPUs yet, and Chapter 6, which is the first publication of the most recent details of a Google Warehouse-scale computer.

As before, the first three appendices in the book give basics on the MIPS instruction set, memory hierachy, and pipelining for readers who have not read a book like *Computer Organization and Design*. To keep costs down but still supply supplemental material that are of interest to some readers, available online at http://booksite.mkp.com/9780123838728/ are nine more appendices. There are more pages in these appendices than there are in this book!

This edition continues the tradition of using real-world examples to demonstrate the ideas, and the "Putting It All Together" sections are brand new. The "Putting It All Together" sections of this edition include the pipeline organizations and memory hierarchies of the ARM Cortex A8 processor, the Intel core i7 processor, the NVIDIA GTX-280 and GTX-480 GPUs, and one of the Google warehouse-scale computers.

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.*)

An Overview of the Content

Chapter 1 has been beefed up in this edition. It includes formulas for energy, static power, dynamic power, integrated circuit costs, reliability, and availability. (These formulas are also found on the front inside cover.) 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 PIAT section has been upgraded to use the new SPECPower benchmark.

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 A. It still uses the MIPS64 architecture. (For quick review, a summary of the MIPS ISA can be found on the back inside cover.) For fans of ISAs, Appendix K covers 10 RISC architectures, the 80x86, the DEC VAX, and the IBM 360/370.

We then move onto memory hierarchy in Chapter 2, since it is easy to apply the cost-performance-energy principles to this material and memory is a critical resource for the rest of the chapters. As in the past edition, Appendix B contains an introductory review of cache principles, which is available in case you need it. Chapter 2 discusses 10 advanced optimizations of caches. The chapter includes virtual machines, which offers advantages in protection, software management, and hardware management and play an important role in cloud computing. In addition to covering SRAM and DRAM technologies, the chapter includes new material on Flash memory. The PIAT examples are the ARM Cortex A8, which is used in PMDs, and the Intel Core i7, which is used in servers.

Chapter 3 covers the exploitation of instruction-level parallelism in high-performance processors, including superscalar execution, branch prediction, speculation, dynamic scheduling, and multithreading. As mentioned earlier, Appendix C is a review of pipelining in case you need it. Chapter 3 also surveys the limits of ILP. Like Chapter 2, the PIAT examples are again the ARM Cortex A8 and the Intel Core i7. While the third edition contained a great deal

on Itanium and VLIW, this material is now in Appendix H, indicating our view that this architecture did not live up to the earlier claims.

The increasing importance of multimedia applications such as games and video processing has also increased the importance of achitectures that can exploit data-level parallelism. In particular, there is a rising interest in computing using graphical processing units (GPUs), yet few architects understand how GPUs really work. We decided to write a new chapter in large part to unveil this new style of computer architecture. Chapter 4 starts with an introduction to vector architectures, which acts as a foundation on which to build explanations of multimedia SIMD instrution set extensions and GPUs. (Appendix G goes into even more depth on vector architectures.) The section on GPUs was the most difficult to write in this book, in that it took many iterations to get an accurate description that was also easy to understand. A significant challenge was the terminology. We decided to go with our own terms and then provide a translation between our terms and the official NVIDIA terms. (A copy of that table can be found in the back inside cover pages.) This chapter introduces the Roofline performance model and then uses it to compare the Intel Core i7 and the NVIDIA GTX 280 and GTX 480 GPUs. The chapter also describes the Tegra 2 GPU for PMDs.

Chapter 5 describes multicore processors. It 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 Intel Core i7. Readers interested in interconnection networks on a chip should read Appendix F, and those interested in larger scale multiprocessors and scientific applications should read Appendix I.

As mentioned earlier, Chapter 6 describes the newest topic in computer architecture, warehouse-scale computers (WSCs). Based on help from engineers at Amazon Web Services and Google, this chapter integrates details on design, cost, and performance of WSCs that few architects are aware of. It starts with the popular MapReduce programming model before describing the architecture and physical implemention of WSCs, including cost. The costs allow us to explain the emergence of cloud computing, whereby it can be cheaper to compute using WSCs in the cloud than in your local datacenter. The PIAT example is a description of a Google WSC that includes information published for the first time in this book.

This brings us to Appendices A through L. Appendix A covers principles of ISAs, including MIPS64, and Appendix K 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 H is related, in that it covers architectures and compilers for VLIW ISAs.

As mentioned earlier, Appendices B and C are tutorials on basic caching and pipelining concepts. Readers relatively new to caching should read Appendix B before Chapter 2 and those new to pipelining should read Appendix C before Chapter 3.

Appendix D, "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. 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.

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

Appendix F, on interconnection networks, has been revised by Timothy M. Pinkston and José Duato. Appendix G, written originally by Krste Asanović, 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 VLIW and EPIC, the architecture of Itanium.

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

Appendix L collects the "Historical Perspective and References" from each chapter 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.

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:

- *Memory Hierarchy:* Appendix B, Chapter 2, and Appendix D.
- *Instruction-Level Parallelism:* Appendix C, Chapter 3, and Appendix H
- *Data-Level Parallelism:* Chapters 4 and 6, Appendix G
- *Thread-Level Parallelism:* Chapter 5, Appendices F and I
- *Request-Level Parallelism:* Chapter 6
- *ISA:* Appendices A and K

Appendix E can be read at any time, but it might work best if read after the ISA and cache sequences. Appendix J can be read whenever arithmetic moves you. You should read the corresponding portion of Appendix L after you complete each chapter.

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. Exercises are rated, to give the reader a sense of the amount of time required to complete an exercise:

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

[15] 5–15 minutes for a full answer

[20] 15–20 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

[Discussion] Topic for discussion with others

Solutions to the case studies and exercises are available for instructors who register at *textbooks.elsevier.com*.

Supplemental Materials

A variety of resources are available online at http://booksite.mkp.com/9780123838728/, including the following:

- Reference appendices—some guest authored by subject experts—covering a range of advanced topics
- Historical Perspectives material that explores the development of the key ideas presented in each of the chapters in the text
- Instructor slides in PowerPoint
- Figures from the book in PDF, EPS, and PPT formats
- Links to related material on the Web
- List of errata

New materials and links to other resources available on the Web will be added 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 (*ca5bugs@mkp.com*).

We welcome general comments to the text and invite you to send them to a separate email address at *ca5comments@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 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 the biographies attest.) 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 fifth edition of this book, we have actually created ten different versions of the text: three versions of the first edition (alpha, beta, and final) and two versions of the second, third, and fourth editions (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.

Contributors to the Fifth Edition

Like prior editions, this is a community effort that involves scores of volunteers. Without their help, this edition would not be nearly as polished.

Reviewers

Jason D. Bakos, University of South Carolina; Diana Franklin, The University of California, Santa Barbara; Norman P. Jouppi, HP Labs; Gregory Peterson, University of Tennessee; Parthasarathy Ranganathan, HP Labs; Mark Smotherman, Clemson University; Gurindar Sohi, University of Wisconsin–Madison; Mateo Valero, Universidad Politécnica de Cataluña; Sotirios G. Ziavras, New Jersey Institute of Technology

Members of the University of California–Berkeley Par Lab and RAD Lab who gave frequent reviews of Chapter 1, 4, and 6 and shaped the explanation of GPUs and WSCs: Krste Asanović, Michael Armbrust, Scott Beamer, Sarah Bird, Bryan Catanzaro, Jike Chong, Henry Cook, Derrick Coetzee, Randy Katz, Yunsup Lee, Leo Meyervich, Mark Murphy, Zhangxi Tan, Vasily Volkov, and Andrew Waterman

Advisory Panel

Luiz André Barroso, Google Inc.; Robert P. Colwell, R&E Colwell & Assoc. Inc.; Krisztian Flautner, VP of R&D at ARM Ltd.; Mary Jane Irwin, Penn State;

David Kirk, NVIDIA; Grant Martin, Chief Scientist, Tensilica; Gurindar Sohi, University of Wisconsin–Madison; Mateo Valero, Universidad Politécnica de Cataluña

Appendices

Krste Asanović, University of California, Berkeley (Appendix G); Thomas M. Conte, North Carolina State University (Appendix E); José Duato, Universitat Politècnica de València and Simula (Appendix F); David Goldberg, Xerox PARC (Appendix J); Timothy M. Pinkston, University of Southern California (Appendix F)

José Flich of the Universidad Politécnica de Valencia provided significant contributions to the updating of Appendix F.

Case Studies with Exercises

Jason D. Bakos, University of South Carolina (Chapters 3 and 4); Diana Franklin, University of California, Santa Barbara (Chapter 1 and Appendix C); Norman P. Jouppi, HP Labs (Chapter 2); Naveen Muralimanohar, HP Labs (Chapter 2); Gregory Peterson, University of Tennessee (Appendix A); Parthasarathy Ranganathan, HP Labs (Chapter 6); Amr Zaky, University of Santa Clara (Chapter 5 and Appendix B)

Jichuan Chang, Kevin Lim, and Justin Meza assisted in the development and testing of the case studies and exercises for Chapter 6.

Additional Material

John Nickolls, Steve Keckler, and Michael Toksvig of NVIDIA (Chapter 4 NVIDIA GPUs); Victor Lee, Intel (Chapter 4 comparison of Core i7 and GPU); John Shalf, LBNL (Chapter 4 recent vector architectures); Sam Williams, LBNL (Roofline model for computers in Chapter 4); Steve Blackburn of Australian National University and Kathryn McKinley of University of Texas at Austin (Intel performance and power measurements in Chapter 5); Luiz Barroso, Urs Hölzle, Jimmy Clidaris, Bob Felderman, and Chris Johnson of Google (the Google WSC in Chapter 6); James Hamilton of Amazon Web Services (power distribution and cost model in Chapter 6)

Jason D. Bakos of the University of South Carolina developed the new lecture slides for this edition.

Finally, a special thanks once again to Mark Smotherman of Clemson University, who gave a final technical reading of our manuscript. 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 wish to thank all the Morgan Kaufmann/Elsevier staff for their efforts and support. For this fifth edition, we particularly want to thank our editors Nate McFadden