

原书第5版

计算机组成与设计

硬件/软件接口

[美] 戴维 A. 帕特森 (David A. Patterson) 约翰 L. 亨尼斯 (John L. Hennessy) 著
加州大学伯克利分校 斯坦福大学
王党辉 康继昌 安建峰 等译
西北工业大学

Computer Organization and Design
The Hardware/Software Interface Fifth Edition

COMPUTER ORGANIZATION AND DESIGN

THE HARDWARE/SOFTWARE INTERFACE

FIFTH EDITION

DAVID A. PATTERSON
JOHN L. HENNESSY



MK
MOSKOWSKI & KUPCHENKO



机械工业出版社
China Machine Press

计 算 机 科 学 丛 书

原书第5版

计算机组成与设计

硬件/软件接口

[美] 戴维 A. 帕特森 (David A. Patterson) 约翰 L. 亨尼斯 (John L. Hennessy) 著
加州大学伯克利分校 斯坦福大学

王党辉 康继昌 安建峰 等译
西北工业大学

Computer Organization and Design

The Hardware/Software Interface Fifth Edition

COMPUTER
ORGANIZATION
AND DESIGN

THE HARDWARE/SOFTWARE INTERFACE

FIFTH EDITION

DAVID A. PATTERSON
JOHN L. HENNESSY



MK
McGraw-Hill Education



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

计算机组成与设计: 硬件 / 软件接口 (原书第 5 版) / (美) 帕特森 (Patterson, D. A.), (美) 亨尼斯 (Hennessy, J. L.) 著; 王党辉等译. —北京: 机械工业出版社, 2015.6
(计算机科学丛书)

书名原文: Computer Organization and Design: The Hardware/Software Interface, Fifth Edition

ISBN 978-7-111-50482-5

I. 计… II. ①帕… ②亨… ③王… III. 计算机体系结构 IV. TP303

中国版本图书馆 CIP 数据核字 (2015) 第 127169 号

本书版权登记号: 图字: 01-2014-2856

Computer Organization and Design: The Hardware/Software Interface, Fifth Edition

David A. Patterson and John L. Hennessy

ISBN:978-0-12-407726-3

Copyright © 2014 by Elsevier Inc. All rights reserved.

Authorized Simplified Chinese translation edition published by the Proprietor.

Copyright © 2015 by Elsevier (Singapore) Pte Ltd. All rights reserved.

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, Macau 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.

本书简体中文版由 Elsevier (Singapore) Pte Ltd. 授权机械工业出版社在中国大陆境内独家出版和发行。本版仅限在中国境内 (不包括香港特别行政区、澳门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。

本书封底贴有 Elsevier 防伪标签, 无标签者不得销售。

本书是计算机组成的经典教材, 着眼于当前计算机设计中最基本的概念, 详细展示软硬件间的关系, 介绍当代计算机系统发展的主流技术和最新成就。

本书以 MIPS 处理器为例介绍计算机硬件技术、汇编语言、计算机算术、流水线以及存储器层次结构等基本技术。书中强调从串行处理到并行处理的最新革新, 每章中都包含并行硬件和软件的主题, 以软硬件协同设计发挥多核性能为最终目标。另外, 本版与时俱进地使用了如 ARM Cortex A8 和 Intel Core i7 等现代设计来说明计算机设计的基本原理。

本书适合作为高等院校计算机专业教材, 对广大技术人员也有很高的参考价值。

出版发行: 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: 姚 蕾

责任校对: 殷 虹

印 刷: 北京诚信伟业印刷有限公司

版 次: 2015 年 7 月第 1 版第 1 次印刷

开 本: 185mm×260mm 1/16

印 张: 34.5

书 号: ISBN 978-7-111-50482-5

定 价: 99.00 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

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzjsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

文艺复兴以来，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；也正是这样的优势，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

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

机械工业出版社华章公司较早意识到“出版要为教育服务”。自1998年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与 Pearson, McGraw-Hill, Elsevier, MIT, John Wiley & Sons, Cengage 等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出 Andrew S. Tanenbaum, Bjarne Stroustrup, Brian W. Kernighan, Dennis Ritchie, Jim Gray, Alfred 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相助，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，有的还专门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。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，我们的联系方式如下：

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



华章科技图书出版中心

本书赞誉

Computer Organization and Design: The Hardware/Software Interface

教材的选择是一个非常令人沮丧的折衷过程——教学方法、内容的覆盖面、行文造句的质量、描述的精确程度、成本等都需要考虑。这是一本不需要在这些方面进行折衷且涉及各个方面的书。它不仅是一部计算机组成的教科书，也是所有计算机科学教科书的范本。

——Michael Goldweber, Xavier 大学

从第 1 版开始，我已经使用本书很多年了。第 5 版在现有的经典内容上做了显著的改进。从桌面计算到移动计算再到大数据，技术的发展开辟了新的研究领域，包括 ARM 之类的嵌入式处理器、软件和硬件如何进行交互以提高性能以及云计算等，这些在本版中都有所体现，且没有牺牲基础知识的内容。

——Ed Harcourt, St. Lawrence 大学

这是一本应该保存在书架上的计算机体系结构的教材。本书既陈旧又新颖，因为它不但介绍了基本原理——摩尔定律、抽象、加速大概率事件、冗余、存储器层级、并行和流水线，也使用如 ARM Cortex A8 和 Intel Core i7 等现代设计对这些基本原理进行了说明。

——Mark D. Hill, Wisconsin-Madison 大学

第 5 版与先进嵌入式和众核（GPU）系统的发展保持同步，它们的发展使平板电脑和智能手机很快变成新的桌面电脑。本书展示了这些变化，并提供了丰富的计算机组成与设计的基本原理，这些内容对于这类新型设备和系统的软硬件设计人员来说非常有用。

——Dave Kaeli, Northeastern 大学

当前，半导体加工工艺按比例缩小的困难在于系统功耗的限制，在移动系统和大数据处理器领域，该限制与不断增长的性能需求之间的矛盾越来越尖锐。第 5 版除了为读者提供计算机体系结构的介绍，还为读者准备了为应对该矛盾而必须在设计方面做出的修改。在这个计算技术的新领域，必须进行软硬件协同设计，另外，系统级体系结构优化与部件级优化一样重要。

——Christos Kozyrakis, Stanford 大学

Patterson 和 Hennessy 非常英明地指出持续发展的计算机硬件结构中的问题，强调硬件和软件模块在不同抽象层次上的交互。第 5 版通过一系列软硬件间的机制将 I/O 和并行概念贯穿全书，对后 PC 时代的计算机体系结构进行了全面的展示。对于在平板 PC 和云计算中面临能效问题和并行性挑战的软件和硬件专家而言，本书是一部必备的指导书。

——Jae C. Oh, Syracuse 大学

D. Patterson 和 J. Hennessy 是计算机领域的知名学者，为计算机学科和产业的发展做出了巨大贡献，他们合著的《Computer Organization and Design: The Hardware/Software Interface》一书现已发行了第5版。该书对计算机组成的研究和设计实践进行了全面系统的总结。目前，国际上许多大学的计算机原理或计算机体系结构课程都采用这本教材，国内也有不少大学采用这本教材。

第5版在保持计算机组成方面传统论题以及前4版特点的基础上，引入了许多近几年计算机领域发展的新论题，如移动计算、大数据等。另外，在实例方面也与时俱进地采用 ARM Cortex A8 和 Intel Core i7 等现代设计对计算机组成的基本原理进行说明。

感谢机械工业出版社华章公司一直关注本书的引进和中译本的出版工作，姚蕾编辑和朱秀英编辑为中译本的翻译工作提出了大量宝贵意见。

感谢清华大学郑纬民教授对前3版中译本所做的工作，是他使得这本重要教材在国内有了广泛的读者。感谢西北工业大学康继昌教授、樊晓桢教授和安建峰副教授对第4版中译本所做的工作，是他们使得第5版的翻译有了很好的基础，另外，康继昌教授和安建峰副教授也参与了本版的翻译工作。

西北工业大学计算机学院的研究生赵磊、刘朝锋、庄森、马鑫、朱迪、朱艳娜等也参加了本书的翻译和校对工作。本书的翻译工作还得到了国家自然科学基金项目 (No. 61472322) 的支持。

由于译者水平有限，书中难免存在一些翻译不当或理解欠妥的地方，希望读者批评指正。

王党辉

2015年2月于西北工业大学

我们能体验的最美好的事物是神秘，它是所有真正的艺术和科学的源泉。

——阿尔伯特·爱因斯坦，《我的信仰》，1930

关于本书

在学习计算机科学与工程时，除了掌握计算的基本原理外，还应该了解该领域的最新进展。各种计算领域中的读者应有机会学习计算机系统的组成理论，因为这是决定计算机系统的功能、性能甚至成功与否的关键。

现代计算机技术需要各种计算方面的专家，他们应对硬件和软件都有深入的理解。硬件和软件在多个层次上的相互关系成为理解计算基本原理的框架。无论你的主要兴趣是硬件还是软件，是计算机科学还是电气工程，计算机组成与设计的基本思想都是相同的。因此，本书着重展示硬件与软件的关系，并重点介绍当今计算机中的基础概念。

近年来，处理器已经由单核发展为多核，这也印证了本书自第1版就预测的发展趋势。有些程序员忽略了这一发展趋势，他们希望计算机体系结构专家、编译器设计者和芯片工程师能够帮助他们，让程序不做任何修改就可以更快或更高效地在新型处理器上运行。但是，这样的时代已经一去不返了。为了使程序运行得更快，必须将其并行化。然而，许多研究者的目标是希望程序员在编写程序时不需要考虑硬件的并行特征，这一目标要很多年才能实现。至少在下一个十年里，大多数程序员必须理解硬件/软件接口，才能编写出能在并行计算机上高效运行的程序。

本书适合以下读者：在汇编语言或逻辑设计方面只有少许经验，需要理解基本计算机组成的读者；具有汇编语言或逻辑设计的基础，需要学习如何设计计算机，或者要进一步理解计算机系统如何工作的读者。

与本书相关的另一本书

有些读者可能已经熟悉作者的另一本书《Computer Architecture: A Quantitative Approach》^①。该书已广为流传，经常以作者姓名命名，称为“Hennessy and Patterson”（本书则常称为“Patterson and Hennessy”）。我们写该书的目的是要用坚实的工程基础和量化的性价比权衡来描述计算机体系结构的原理。我们以商用系统为例，用测量的方法来描述实际的设计经验。我们的目标是用量化的方法而不是用描述的方法学习计算机体系结构，希望这一方法有助于培养能精确理解计算机的专业人才。

本书的大多数读者并不一定要成为计算机体系结构的设计者。但是，未来软件设计人员与软件系统协同工作的基本硬件技术的理解程度，将直接影响软件系统的性能和能效。因此，编译器设计者、操作系统设计者、数据库程序员以及其他大多数软件工程师对本书的原理必须有充分的了解。同样，硬件设计者也必须清楚地理解他们的工作对软件的相应影响。

所以，本书的内容远多于“Hennessy and Patterson”，而且进行了大量修订，以适应不同专业的读者。我们对再版“Hennessy and Patterson”时删除的大量介绍性材料的效果感到满意，这使得新版与第1版内容的重叠大大降低，本书亦如此。

^① 机械工业出版社已出版了本书的第3版、第4版和第5版影印书，书名为《计算机体系结构：量化研究方法》。——编辑注









第5版的变化






第5版有6个主要的目标：使用运行例子的方法论证理解硬件的重要性；对前面已经提到的重要技术采用黑体的方式进行强调；对例子进行了更新，以反映从PC时代到后PC时代的发展；将I/O吞吐率方面的材料贯穿在整个本书中，而不是集中在一章中；对技术内容进行了更新，以反映自2009年第4版出版以来工业界的变化；将附录和可选章节（目录中带有📖图标的章节）的内容放在互联网上（booksite.elsevier.com/9780124077263/index.php），而不是放在CD上，降低了本书的成本，也使该版本变成了一部电子书。

在详细介绍第5版的修订目标之前，首先看下表。该表给出了本书的主要内容，并为关注硬件和关注软件的两种读者分别进行了导读。其中，第1、4、5和6章对两种读者同等重要。第1章讨论了能耗的重要性和由其引发的处理器从单核向多核的转变，并介绍了计算机设计中的8个伟大思想。第2章对于硬件读者来说很可能是复习性材料，而对于软件读者来说则是重要的阅读材料，特别是希望深入学习编译器和面向对象编程语言的读者。第3章适合对定点运算或者浮点运算感兴趣的读者，有些读者可能不需要学习第3章。然而，我们将在本章介绍矩阵乘法运行的例子，展示如何采用子字并行方法将性能提高4倍，因此不要跳过3.6~3.8节。第4章介绍了流水线处理器。其中，4.1、4.5和4.10节给出了流水线概述，4.12节给出了进一步提高矩阵乘法运算性能的方法，这些小节对于软件设计者来说比较重要。对于硬件设计者，第4章是核心内容。另外，根据读者知识背景的不同，可以选择是否首先阅读附录C中的逻辑设计部分。最后一章是多核、多处理器和集群系统，这一章是全新的内容，因此所有读者都应该阅读。本版的重要组织结构是使许多思想的引入更加自然，例如GPU、仓储式计算机和集群系统中的关键——网络接口卡的软硬件接口。

章/附录	节	关注软件	关注硬件
第1章 计算机概要与技术	1.1~1.11	🕶️	🕶️
	📖1.12 (历史)	🕶️	🕶️
第2章 指令：计算机的语言	2.1~2.14	🕶️	🕶️
	📖2.15 (编译器和Java)	🕶️	
	2.16~2.20	🕶️	🕶️
附录E RISC指令集体系结构	📖2.21 (历史)	🕶️	🕶️
	📖E.1~E.17	🕶️	
第3章 计算机的算术运算	3.1~3.5	🕶️	🕶️
	3.6~3.8 (子字并行)	🕶️	🕶️
	3.9~3.10 (谬误)	🕶️	🕶️
	📖3.11 (历史)	🕶️	🕶️
附录B 逻辑设计基础	B.1~B.13		🕶️
第4章 处理器	4.1 (引言)	🕶️	🕶️
	4.2 (逻辑设计惯例)		🕶️
	4.3~4.4 (简单实现)	🕶️	🕶️
	4.5 (流水线概述)	🕶️	🕶️
	4.6 (流水线数据通路)	🕶️	🕶️
	4.7~4.9 (冒险和异常)		🕶️
	4.10~4.12 (并行和实例)	🕶️	🕶️
	📖4.13 (Verilog 流水线控制)		🕶️
	4.14~4.15 (谬误)	🕶️	🕶️
	📖4.16 (历史)	🕶️	🕶️

(续)

章/附录	节	关注软件	关注硬件
附录 D 控制通路的硬件实现	§ D.1 ~ D.6		
第 5 章 大容量和高速度: 开发存储器层次结构	5.1 ~ 5.10		
	§ 5.11 (廉价冗余磁盘阵列)		
	§ 5.12 (Verilog cache 控制器)		
	5.13 ~ 5.16		
	§ 5.17 (历史)		
第 6 章 从客户端到云的并行处理器	6.1 ~ 6.8		
	§ 6.9 (网络)		
	6.10 ~ 6.14		
	§ 6.15 (历史)		
附录 A 汇编器、链接器和 SPIM 仿真器	A.1 ~ A.11		
附录 C 图形处理单元	§ C.1 ~ C.13		

仔细阅读  有时间阅读  作为参考  回顾或阅读  拓展阅读 

第 5 版的第一个目标是使用一个例子来论证理解硬件对提高性能和能效的重要性。正如前面所述, 在第 3 章, 我们采用子字并行将矩阵乘法加速 4 倍, 在第 4 章通过循环展开将性能翻倍, 证明了指令集并行的价值。第 5 章通过分块技术对 cache 进行优化, 再次将性能翻倍。第 6 章通过在 16 个处理器上的线程级并行获得了 14 倍的加速比。这 4 种优化技术仅仅在原始的矩阵乘法例子中的 C 代码上增加了 24 行。

第二个目标是通过提早介绍计算机体系结构设计中的 8 个伟大思想并在整本书中明确指出它们的应用之处来帮助读者理解计算机设计的精髓。我们采用黑体文字的方式向读者提醒这 8 个思想的应用情况, 在书中大约有 100 次引用。每一章中至少有 7 处应用这些思想的例子, 并且每个思想至少被引用 5 次。通过并行提高性能、流水线和预测技术是引用次数最多的 3 个思想, 紧接着是摩尔定律。第 4 章讲述处理器, 是例子最多的一章, 也是最吸引计算机设计者的一章。每一章都能找到的伟大思想是通过并行提高性能, 这是近年来计算机领域中的一个重要发展方向。

第三个目标是通过例子和材料来识别计算技术从 PC 时代进入后 PC 时代的变化。因此, 第 1 章直接介绍了平板电脑而没有介绍 PC, 第 6 章介绍了云计算的基础设施。另外, 在指令集方面, 我们介绍了后 PC 时代中个人移动设备里使用的 ARM 指令集, 以及在 PC 时代和云计算中占主导地位的 x86 指令集。

第四个目标是将 I/O 吞吐率方面的材料贯穿在整本书中, 而不是集中在一章中, 这与第 4 版中将并行性贯穿全书一样, 因此, 本版可在 1.4、4.9、5.2、5.5、5.11 和 6.9 节中找到 I/O 相关的材料。我们的想法是如果不把这些内容集中在一章, 则读者 (和教师) 能更好地学习与掌握 I/O。

计算机是一个快速发展的领域, 对于本书新的版本也是如此, 编写新版的一个重要目的是更新技术内容。实际的例子就是反映后 PC 时代特点的 ARM Cortex A8 和 Intel Core i7, 其他的亮点包括新的 ARMv8 64 位指令集、讲解 GPU 特有术语的教程、组成云的仓储式计算机的内涵以及对 10G 以太网卡的理解。

为了保持纸质书的厚度及其与电子书的兼容性, 我们一改以前版本的做法, 将可选内容由随书 CD 改为网络在线的形式。

最后, 我们更新了本书的所有练习题。

在对内容进行修订的同时，我们保留了以往版本中有用的元素。为使本书更好地作为参考书，我们还在新术语第一次出现时给出了定义。书中标题为“理解程序性能”部分的内容有助于读者理解程序的性能，以及如何提高性能，就像书中“硬件/软件接口”部分会帮助读者理解有关接口的权衡问题一样。“重点”部分仍然存在，以使读者看到整个“森林”而不是每一棵“树”。每章最后提供“小测验”部分的答案，帮助读者在第一时间加强对内容的理解。本版同样提供了 MIPS 参考数据（这是从 IBM System/360 “绿卡”得到的灵感），并对数据进行了更新，在编写 MIPS 汇编语言程序时，这应该是一个很好的参考。

教学支持[⊖]

我们收集了大量材料供教师授课使用，包括题解、图表、幻灯片等，可从出版商处获得。如需更多信息，请访问网址：textbooks.elsevier.com/9780124077263。

结语

从下面的致谢中，你可以发现我们花费了大量精力去修改本书的错误。由于本书印刷了多次，因此我们有机会做更多的校正。如果你发现有遗留的错误，请通过电子邮件与出版社联系：cod5bugs@mkp.com。

本版标志着 Hennessy 和 Patterson 自 1989 年以来长期合作的第二次中止。由于要管理一所世界知名的大学，Hennessy 校长将不能继续承担新版本的实际编写工作。留下 Patterson 一人感觉自己像是和伙伴们一起演出的演员，突然被推到前台独自表演。所以，在致谢名单中列出的人和 Berkeley 的同行们在本书的撰写过程中甚至起了更大的作用。

第 5 版致谢

在本书的每一版中，我们都非常幸运地得到了来自许多读者、评审者和其他人员的帮助。每个人的帮助都使本书更加完美。

由于第 6 章做了巨大的修改，因此我们对其思想和内容进行了单独的评审，并基于每位评审人的反馈意见做了修改。感谢 Stanford 大学的 Christos Kozyrakis，他建议在集群中使用网络接口来论证 I/O 的软硬件接口，并对该章的组织提出了意见。还要感谢 Stanford 大学的 Mario Flajsilk，他提供了 NetFPGA NIC 的细节、表格以及性能评估。另外，以下人员对本章提出了修改建议：Northeastern 大学的 David Kaeli、HP 实验室的 Partha Ranganathan、Wisconsin 大学的 David Wood 以及 Berkeley 大学的同事 Siamak Faridani、Shoab Kamil、Yunsup Lee、Zhangxi Tan 和 Andrew Waterman。

我们要对 UC Berkeley 的 Rimas Avizenis 表示特别的感谢，他开发了不同版本的矩阵乘法程序，并提供了相应的性能数据。当我在 UCLA 读研究生时，我与他的父亲一起工作，能够与他一起在 UCB 共事是一件美好的事情。

我也要对我的长期合作伙伴——UC Berkeley 的 Randy Katz 表示感谢。我们共同讲授研究生的计算机体系结构课程，他在开发计算机体系结构的伟大思想方面提供了很大的帮助。

感谢 David Kirk、John Nickolls 和他们在 NVIDIA 的同事们（Michael Garland、John Mon-

⊖ 爱思唯尔 (ELS)：关于本书教辅资源，使用教材的老师需通过爱思唯尔的教材网站 (www.textbooks.elsevier.com) 注册并通过审批后才能获取相关资源。具体方法如下：在 www.textbooks.elsevier.com 教材网站查找到该书后，点击“instructor manual”便可申请查看该教师手册。有任何问题，请致电 010-85208853。

——编辑注

trym、Doug Voorhies、Lars Nyland、Erik Lindholm、Paulius Micikevicius、Massimiliano Fatica、Stuart Oberman、Vasily Volkov) 提供了第一个深入介绍 GPU 的附录。再次感谢 Jim Larus, 他现在是 EPFL 计算机与通信科学学院的院长, 为本书发挥了在汇编语言方面的专长, 欢迎本书读者使用他所开发和维护的模拟器。

非常感谢 South Carolina 大学的 Jason Bakos, 他在第 4 版的基础上对本版的练习题进行了更新。第 4 版的练习题由以下人员编写: Perry Alexander (Kansas 大学)、Javier Bruguera (de Santiago de Compostela 大学)、Matthew Farrens (California 大学 Davis 分校)、David Kaeli (Northeastern 大学)、Nicole Kaiyan (Adelaide 大学)、John Oliver (Cal Poly, San Luis Obispo)、Milos Prvulovic (Georgia 理工大学) 和 HP 的 Jichuan Chang、Jacob Leverich、Kevin Lim、Partha Ranganathan。

感谢 Jason Bakos 开发了新的幻灯片。

感谢许多教师的贡献, 他们回答出版商的调查问卷、评审我们的提议、出席小组会议, 并对本版的计划进行分析和反馈。他们是:

2012 焦点小组: Bruce Barton (Suffolk County Community 学院), Jeff Braun (Montana 理工大学), Ed Gehringer (North Carolina State), Michael Goldweber (Xavier 大学), Ed Harcourt (St. Lawrence 大学), Mark Hill (Wisconsin 大学 Madison 分校), Patrick Homer (Arizona 大学), Norm Jouppi (HP 实验室), Dave Kaeli (Northeastern 大学), Christos Kozyrakis (Stanford 大学), Zachary Kurmas (Grand Valley 州立大学), Jae C. Oh (Syracuse 大学), Lu Peng (Louisiana 州立大学), Milos Prvulovic (Georgia 理工), Partha Ranganathan (HP 实验室), David Wood (Wisconsin 大学), Craig Zilles (Illinois 大学香槟分校)。

参考调查和审阅的学者: Mahmoud Abou-Nasr (Wayne 州立大学), Perry Alexander (Kansas 大学), Hakan Aydin (George Mason 大学), Hussein Badr (New York 州立大学 Stony Brook 分校), Mac Baker (Virginia Military 学院), Ron Barnes (George Mason 大学), Douglas Blough (Georgia 理工), Kevin Bolding (Seattle Pacific 大学), Miodrag Bolic (Ottawa 大学), John Bonomo (Westminster 学院), Jeff Braun (Montana 理工大学), Tom Briggs (Shippensburg 大学), Scott Burgess (Humboldt 州立大学), Fazli Can (Bilkent 大学), Warren R. Carithers (Rochester 理工学院), Bruce Carlton (Mesa Community 学院), Nicholas Carter (Illinois 大学香槟分校), Anthony Cocchi (New York 城市大学), Don Cooley (Utah 州立大学), Robert D. Cupper (Allegheny 学院), Edward W. Davis (North Carolina 州立大学), Nathaniel J. Davis (Air Force 理工学院), Molisa Derk (Oklahoma 城市大学), Derek Eager (Saskatchewan 大学), Ernest Ferguson (Northwest Missouri 州立大学), Rhonda Kay Gaede (Alabama 大学), Etienne M. Gagnon (UQAM), Costa Gerosis (Christopher Newport 大学), Paul Gillard (Newfoundland 纪念大学), Michael Goldweber (Xavier 大学), Georgia Grant (San Mateo 学院), Merrill Hall (The Master's 学院), Tyson Hall (Southern Adventist 大学), Ed Harcourt (St. Lawrence 大学), Justin E. Harlow (South Florida 大学), Paul F. Hemler (Hampden-Sydney 学院), Martin Herbordt (Boston 大学), Steve J. Hodges (Cabrillo 学院), Kenneth Hopkinson (Cornell 大学), Dalton Hunkins (St. Bonaventure 大学), Baback Izadi (New York 州立大学 New Paltz 分校), Reza Jafari, Robert W. Johnson (Colorado Technical 大学), Bharat Joshi (North Carolina 大学 Charlotte 分校), Nagarajan Kandasamy (Drexel 大学), Rajiv Kapadia, Ryan Kastner (California 大学 Santa Barbara 分校), E. J. Kim (Texas A&M 大学), Jihong Kim (Seoul 国立大学), Jim Kirk (Union 大学), Geoffrey S. Knauth (Lycoming 学院), Manish M. Kochhal (Wayne 州立大学), Suzan Koknar-Tezel (Saint Joseph 大学), Angkul Kongmunvattana (Columbus 州立大学), April Kontostathis (Ursinus

学院), Christos Kozyrakis (Stanford 大学), Danny Krizanc (Wesleyan 大学), Ashok Kumar, S. Kumar (Texas 大学), Zachary Kurmas (Grand Valley 州立大学), Robert N. Lea (Houston 大学), Baoxin Li (Arizona 州立大学), Li Liao (Delaware 大学), Gary Livingston (Massachusetts 大学), Michael Lyle, Douglas W. Lynn (Oregon 理工学院), Yashwant K Malaiya (Colorado 州立大学), Bill Mark (Texas 大学 Austin 分校), Ananda Mondal (Claflin 大学), Alvin Moser (Seattle 大学), Walid Najjar (California 大学 Riverside 分校), Danial J. Neebel (Loras 学院), John Nestor (Lafayette 学院), Jae C. Oh (Syracuse 大学), Joe Oldham (Centre 学院), Timour Paltashev, James Parkerson (Arkansas 大学), Shaunak Pawagi (SUNY Stony Brook 分校), Steve Pearce, Ted Pedersen (Minnesota 大学), Lu Peng (Louisiana 州立大学), Gregory D Peterson (Tennessee 大学), Milos Prvulovic (Georgia 理工), Partha Ranganathan (HP 实验室), Dejan Raskovic (Alaska 大学 Fairbanks 分校), Brad Richards (Puget Sound 大学), Roman Rozanov, Louis Rubinfeld (Villanova 大学), Md Abdus Salam (Southern 大学), Augustine Samba (Kent 州立大学), Robert Schaefer (Daniel Webster 学院), Carolyn J. C. Schauble (Colorado 州立大学), Keith Schubert (CSU San Bernardino 分校), William L. Schultz, Kelly Shaw (Richmond 大学), Shahram Shirani (McMaster 大学), Scott Sigman (Drury 大学), Bruce Smith, David Smith, Jeff W. Smith (Georgia 大学, Athens), Mark Smotherman (Clemson 大学), Philip Snyder (Johns Hopkins 大学), Alex Sprintson (Texas A&M), Timothy D. Stanley (Brigham Young 大学), Dean Stevens (Morningside 学院), Nozar Tabrizi (Kettering 大学), Yuval Tamir (UCLA), Alexander Taubin (Boston 大学), Will Thacker (Winthrop 大学), Mithuna Thottethodi (Purdue 大学), Manghui Tu (Southern Utah 大学), Dean Tullsen (UC San Diego 分校), Rama Viswanathan (Beloit 学院), Ken Vollmar (Missouri 州立大学), Guoping Wang (Indiana-Purdue 大学), Patricia Wenner (Bucknell 大学), Kent Wilken (California 大学 Davis 分校), David Wolfe (Gustavus Adolphus 学院), David Wood (Wisconsin 大学 Madison 分校), Ki Hwan Yum (Texas 大学 San Antonio 分校), Mohamed Zahran (New York 城市学院), Gerald D. Zarnett (Ryerson 大学), Nian Zhang (South Dakota School of Mines & Technology), Jiling Zhong (Troy 大学), Huiyang Zhou (Central Florida 大学), Weiyu Zhu (Illinois Wesleyan 大学)。

特别感谢 Mark Smotherman 一遍又一遍地查找本书中的技术错误和写作错误, 他的工作显著地改进了这一版的质量。

还要感谢 Morgan Kaufmann 公司同意在 Todd Green 和 Nate McFadden 的领导下对本书进行再版, 没有他们的工作, 我不可能完成本书。我们还要感谢 Lisa Jones 和 Russell Purdy, 她们分别负责管理出版过程和封面设计。新封面将本版中后 PC 时代的内容和第 1 版的封面呼应了起来。

以上提到的近 150 名人士为本版提供了大量帮助, 使之成为我们希望的最好的书。

David A. Patterson

作者简介

Computer Organization and Design: The Hardware/Software Interface

David A. Patterson

自从 1977 年在加州大学伯克利分校任职开始一直讲授计算机体系结构课程，他现在是计算机科学系的执行主席。他的教学工作获得了加州大学优秀教学奖、ACM Karlstrom 奖、IEEE 的 Mulligan 教育奖章和本科生教学奖。因为对 RISC 的贡献，Patterson 获得了 IEEE 技术进步奖和 ACM Eckert-Mauchly 奖，另外因为对 RAID 的贡献，他与别人分享了 IEEE Johnson 信息存储奖。他和 John Hennessy 分享了 IEEE John von Neumann 奖章与 C & C 奖励。Patterson 是美国艺术与科学院、计算机历史博物馆、ACM 和 IEEE 院士，并且入选国家工程院、国家科学院和硅谷工程名人堂。他曾任美国总统信息咨询委员会成员、伯克利 EECS 系 CS 分会主席、计算研究学会主席和 ACM 主席。这个纪录使他获得了 ACM 和 CRA 的杰出服务奖。

在伯克利，Patterson 领导了 RISC I 的设计与实现，这是第一款精简指令系统计算机，并且是商用 SPARC 体系结构的基础。他是廉价磁盘冗余阵列（RAID）项目的负责人，RAID 技术引导许多公司开发出了高可靠存储系统。他也参加了工作站网络（NOW）项目，该项目先引导互联网公司使用集群技术，再引导了后来的云计算。这些项目获得了三个 ACM 最佳论文奖。他当前的研究包括算法-机器-人、面向可证明优化实现的高可靠高效算法与专家系统。AMP 实验室正在开发可扩展的机器学习算法、仓储式计算机编程模型以及密集资源工具，以从云中的大数据获得有价值的信息。ASPIRE 实验室使用深度的硬件和软件协同技术在移动和货架计算系统中获得最高性能和能效。

John L. Hennessy

斯坦福大学的第十任校长，他从 1977 年开始任职于斯坦福大学电子工程与计算机科学系。Hennessy 是 ACM 和 IEEE 会士、国家工程院成员、国家科学院成员、美国艺术与科学院院士。他获得了许多奖项，其中包括：因为对 RISC 的贡献获得的 2001 年 ACM Eckert-Mauchly 奖，2001 年 Seymour Cray 计算机工程奖，与 Patterson 分享了 2000 年 John von Neumann 奖。他还获得了七个荣誉博士学位。

1981 年，他在斯坦福大学与几个研究生开始了 MIPS 项目。在 1984 年完成了该项目后，他离开大学，与他人共同创建了 MIPS 计算机系统（现在的 MIPS 技术公司），该公司开发了第一款商用 RISC 微处理器。2006 年，MIPS 微处理器销售了 20 亿片，应用范围从视频游戏和掌上计算机到激光打印机和网络交换机。后来，Hennessy 领导了 DASH（共享存储器的主导体系结构）项目，该项目建立了第一个可扩展 cache 一致性多处理器的原形系统，其许多关键思想已经应用在先进的多处理器中。除了技术活动与大学工作外，他还是多家创业公司的早期顾问和投资者。

出版者的话	1	1.8 沧海巨变：从单处理器向多处理器 转变	29
本书赞誉		1.9 实例：Intel Core i7 基准	31
译者序		1.9.1 SPEC CPU 基准测试程序	31
前言		1.9.2 SPEC 功耗基准测试程序	32
作者简介		1.10 谬误与陷阱	33
第1章 计算机概要与技术	1	1.11 本章小结	35
1.1 引言	1	1.12 历史观点和拓展阅读	36
1.1.1 计算应用的分类及其特性	2	1.13 练习题	36
1.1.2 欢迎来到后 PC 时代	3	第2章 指令：计算机的语言	40
1.1.3 你能从本书学到什么	4	2.1 引言	40
1.2 计算机系统结构中的8个伟大 思想	6	2.2 计算机硬件的操作	43
1.2.1 面向摩尔定律的设计	6	2.3 计算机硬件的操作数	44
1.2.2 使用抽象简化设计	6	2.3.1 存储器操作数	45
1.2.3 加速大概率事件	6	2.3.2 常数或立即数操作数	47
1.2.4 通过并行提高性能	7	2.4 有符号数和无符号数	48
1.2.5 通过流水线提高性能	7	2.5 计算机中指令的表示	53
1.2.6 通过预测提高性能	7	2.6 逻辑操作	58
1.2.7 存储器层次	7	2.7 决策指令	60
1.2.8 通过冗余提高可靠性	7	2.7.1 循环	61
1.3 程序概念入门	7	2.7.2 case/switch 语句	63
1.4 硬件概念入门	10	2.8 计算机硬件对过程的支持	64
1.4.1 显示器	11	2.8.1 使用更多的寄存器	66
1.4.2 触摸屏	12	2.8.2 嵌套过程	67
1.4.3 打开机箱	12	2.8.3 在栈中为新数据分配空间	69
1.4.4 数据安全	15	2.8.4 在堆中为新数据分配空间	70
1.4.5 与其他计算机通信	16	2.9 人机交互	72
1.5 处理器和存储器制造技术	17	2.10 MIPS 中 32 位立即数和寻址	75
1.6 性能	20	2.10.1 32 位立即数	75
1.6.1 性能的定义	20	2.10.2 分支和跳转中的寻址	76
1.6.2 性能的度量	22	2.10.3 MIPS 寻址模式总结	78
1.6.3 CPU 性能及其因素	23	2.10.4 机器语言解码	79
1.6.4 指令的性能	24	2.11 并行与指令：同步	81
1.6.5 经典的 CPU 性能公式	25	2.12 翻译并执行程序	83
1.7 功耗墙	27	2.12.1 编译器	83
		2.12.2 汇编器	84

2.12.3	链接器	85	3.4.1	除法算法及其硬件结构	125
2.12.4	加载器	87	3.4.2	有符号除法	128
2.12.5	动态链接库	87	3.4.3	更快速的除法	128
2.12.6	启动一个 Java 程序	89	3.4.4	MIPS 中的除法	129
2.13	以一个 C 排序程序作为完整的例子	90	3.4.5	小结	129
2.13.1	swap 过程	90	3.5	浮点运算	130
2.13.2	sort 过程	91	3.5.1	浮点表示	131
2.14	数组与指针	96	3.5.2	浮点加法	135
2.14.1	用数组实现 clear	96	3.5.3	浮点乘法	138
2.14.2	用指针实现 clear	97	3.5.4	MIPS 中的浮点指令	139
2.14.3	比较两个版本的 clear	97	3.5.5	算术精确性	145
2.15	高级内容: 编译 C 语言和解释 Java 语言	98	3.5.6	小结	146
2.16	实例: ARMv7 (32 位) 指令集	98	3.6	并行性和计算机算术: 子字并行	148
2.16.1	寻址模式	99	3.7	实例: x86 中流处理 SIMD 扩展和高级向量扩展	149
2.16.2	比较和条件分支	100	3.8	加速: 子字并行和矩阵乘法	150
2.16.3	ARM 的特色	100	3.9	谬误与陷阱	153
2.17	实例: x86 指令集	102	3.10	本章小结	155
2.17.1	Intel x86 的改进	102	3.11	历史观点和拓展阅读	158
2.17.2	x86 寄存器和数据寻址模式	103	3.12	练习题	159
2.17.3	x86 整数操作	105			
2.17.4	x86 指令编码	107	第 4 章 处理器		162
2.17.5	x86 总结	108	4.1	引言	162
2.18	实例: ARMv8 (64 位) 指令集	108	4.2	逻辑设计的一般方法	165
2.19	谬误与陷阱	109	4.3	建立数据通路	167
2.20	本章小结	110	4.4	一个简单的实现机制	173
2.21	历史观点和拓展阅读	111	4.4.1	ALU 控制	173
2.22	练习题	112	4.4.2	主控制单元的设计	175
			4.4.3	为什么不使用单周期实现方式	181
第 3 章 计算机的算术运算		117	4.5	流水线概述	182
3.1	引言	117	4.5.1	面向流水线的指令集设计	186
3.2	加法和减法	117	4.5.2	流水线冒险	186
3.3	乘法	121	4.5.3	对流水线概述的小结	191
3.3.1	顺序的乘法算法和硬件	121	4.6	流水线数据通路及其控制	192
3.3.2	有符号乘法	124	4.6.1	图形化表示的流水线	200
3.3.3	更快速的乘法	124	4.6.2	流水线控制	203
3.3.4	MIPS 中的乘法	124	4.7	数据冒险: 旁路与阻塞	206
3.3.5	小结	125	4.8	控制冒险	214
3.4	除法	125	4.8.1	假定分支不发生	215

4.8.2	缩短分支的延迟	215	5.4	cache 性能的评估和改进	270
4.8.3	动态分支预测	216	5.4.1	通过更灵活地放置块来减少 cache 缺失	272
4.8.4	流水线小结	220	5.4.2	在 cache 中查找一个块	275
4.9	异常	221	5.4.3	替换块的选择	276
4.9.1	MIPS 体系结构中的异常 处理	221	5.4.4	使用多级 cache 结构减少 缺失代价	277
4.9.2	在流水线实现中的异常	222	5.4.5	通过分块进行软件优化	280
4.10	指令级并行	226	5.4.6	小结	283
4.10.1	推测的概念	227	5.5	可信存储器层次	283
4.10.2	静态多发射处理器	227	5.5.1	失效的定义	283
4.10.3	动态多发射处理器	231	5.5.2	纠正一位错、检测两位错的 汉明编码 (SEC/DED)	284
4.10.4	能耗效率与高级流水线	233	5.6	虚拟机	287
4.11	实例: ARM Cortex-A8 和 Intel Core i7 流水线	234	5.6.1	虚拟机监视器的必备 条件	289
4.11.1	ARM Cortex-A8	235	5.6.2	指令集系统结构 (缺乏) 对虚拟机的支持	289
4.11.2	Intel Core i7 920	236	5.6.3	保护和指令集系统结构	289
4.11.3	Intel Core i7 920 的性能	238	5.7	虚拟存储器	290
4.12	运行更快: 指令级并行和矩阵 乘法	240	5.7.1	页的存放和查找	293
4.13	高级主题: 通过硬件设计语言 描述和建模流水线来介绍数字 设计以及更多流水线示例	242	5.7.2	缺页故障	294
4.14	谬误与陷阱	242	5.7.3	关于写	297
4.15	本章小结	243	5.7.4	加快地址转换: TLB	297
4.16	历史观点和拓展阅读	243	5.7.5	集成虚拟存储器、TLB 和 cache	300
4.17	练习题	243	5.7.6	虚拟存储器中的保护	302
			5.7.7	处理 TLB 缺失和缺页	303
			5.7.8	小结	307
			5.8	存储器层次结构的一般框架	309
第 5 章	大容量和高速度: 开发 存储器层次结构	252	5.8.1	问题 1: 一个块可以被放在 何处	309
5.1	引言	252	5.8.2	问题 2: 如何找到一个块	310
5.2	存储器技术	255	5.8.3	问题 3: 当 cache 缺失时 替换哪一块	311
5.2.1	SRAM 技术	256	5.8.4	问题 4: 写操作如何处理	311
5.2.2	DRAM 技术	256	5.8.5	3C: 一种理解存储器层次 结构行为的直观模型	312
5.2.3	闪存	258	5.9	使用有限状态机来控制简单的 cache	314
5.2.4	磁盘存储器	258	5.9.1	一个简单的 cache	314
5.3	cache 的基本原理	259	5.9.2	有限状态机	315
5.3.1	cache 访问	261			
5.3.2	cache 缺失处理	265			
5.3.3	写操作处理	266			
5.3.4	一个 cache 的例子: 内置 FastMATH 处理器	267			
5.3.5	小结	269			

5.9.3 一个简单的 cache 控制器的 有限状态机	316
5.10 并行与存储器层次结构: cache 一致性	317
5.10.1 实现一致性的基本方案	318
5.10.2 监听协议	319
5.11 并行与存储器层次结构: 冗余 廉价磁盘阵列	320
5.12 高级内容: 实现 cache 控制器	320
5.13 实例: ARM Cortex-A8 和 Intel Core i7 的存储器层次结构	320
5.14 运行更快: cache 分块和矩阵 乘法	324
5.15 谬误和陷阱	326
5.16 本章小结	329
5.17 历史观点和拓展阅读	329
5.18 练习题	329
第 6 章 从客户端到云的并行 处理器	340
6.1 引言	340
6.2 创建并行处理程序的难点	342
6.3 SISD、MIMD、SIMD、SPMD 和向量机	345
6.3.1 在 x86 中的 SIMD: 多媒体 扩展	346
6.3.2 向量机	346
6.3.3 向量与标量的对比	347
6.3.4 向量与多媒体扩展的 对比	348
6.4 硬件多线程	350
6.5 多核和其他共享内存 多处理器	352
6.6 图形处理单元简介	355
6.6.1 NVIDIA GPU 体系结构 简介	356
6.6.2 NVIDIA GPU 存储结构	357
6.6.3 GPU 展望	358
6.7 集群、仓储级计算机和其他消息 传递多处理器	360
6.8 多处理器网络拓扑简介	363
6.9 与外界通信: 集群网络	366
6.10 多处理器测试集程序和性能 模型	366
6.10.1 性能模型	368
6.10.2 Roofline 模型	369
6.10.3 两代 Opteron 的比较	370
6.11 实例: 评测 Intel Core i7 960 和 NVIDIA Tesla GPU 的 Roofline 模型	373
6.12 运行更快: 多处理器和 矩阵乘法	376
6.13 谬误与陷阱	378
6.14 本章小结	379
6.15 历史观点和拓展阅读	381
6.16 练习题	382
附录 A 汇编器、链接器和 SPIM 仿真器	389
附录 B 逻辑设计基础	437
索引	494