



计 算 机 科 学 丛 书

原书第4版



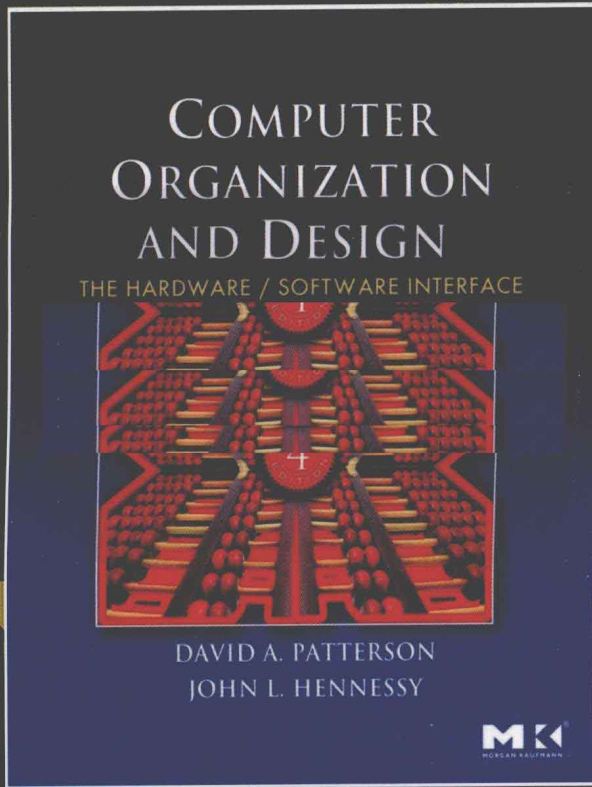
计算机组成与设计

硬件/软件接口

(美) David A. Patterson John L. Hennessy 著 康继吕 樊晓桢 安建峰 等译

Computer Organization and Design

The Hardware / Software Interface Fourth Edition



附光盘



机械工业出版社
China Machine Press

计 算 机 科 学 丛 书

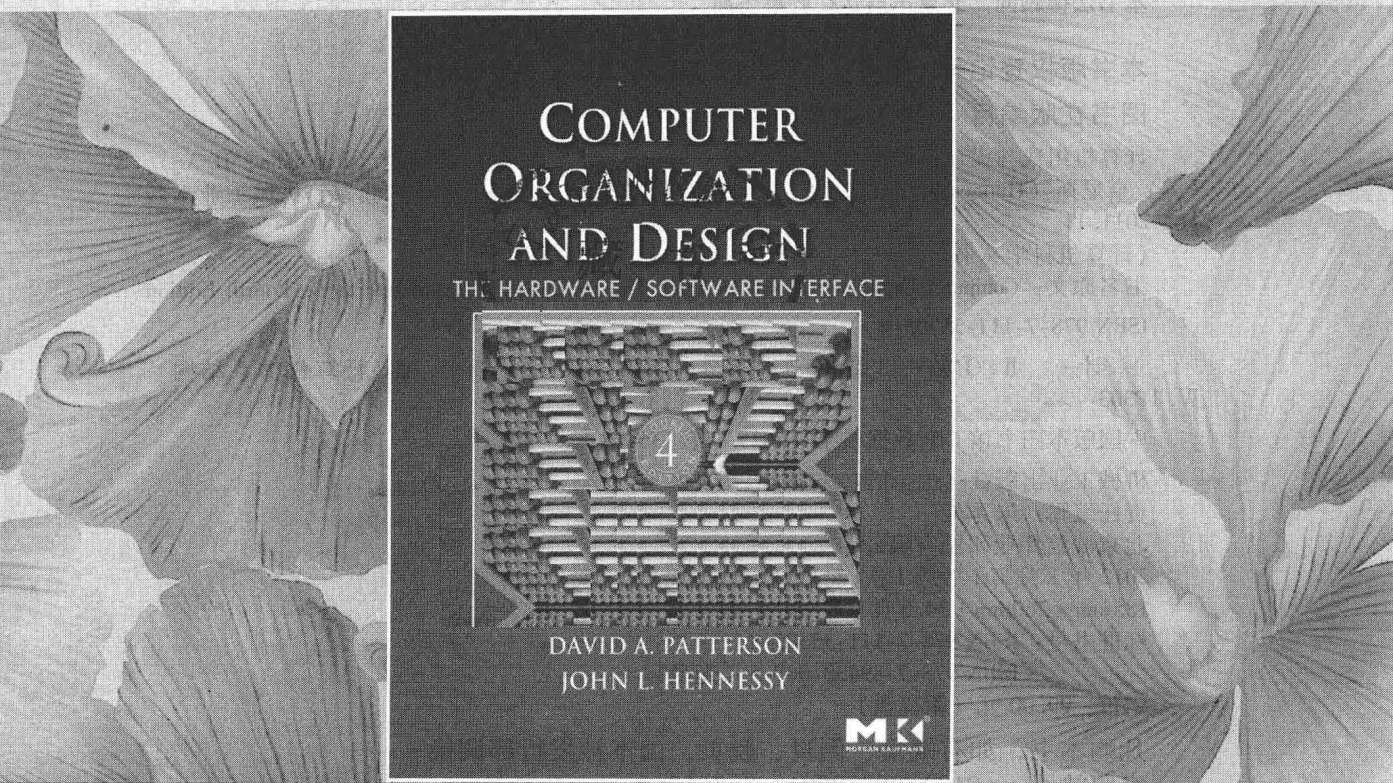
原书第4版

计算机组成与设计

硬件/软件接口

(美) David A. Patterson John L. Hennessy 著 康继昌 樊晓桠 安建峰 等译

Computer Organization and Design
The Hardware / Software Interface Fourth Edition



机械工业出版社
China Machine Press

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

同以往版本一样，本书采用 MIPS 处理器作为展示计算机硬件技术、汇编语言、计算机算术、流水线、存储器层次结构以及 I/O 等基本功能的核心。书中强调了计算机从串行到并行的最新革新，在每章中都纳入了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。

本书适合作为高等院校相关专业的本科生和研究生教材，对广大技术人员也有很高的参考价值。

David A. Patterson and John L. Hennessy: Computer Organization and Design; The Hardware/Software Interface, Fourth Edition (ISBN 978-0-12-374493-7) .

Copyright © 2009 by Elsevier Inc. All rights reserved.

Authorized Simplified Chinese translation edition published by the Proprietor.

ISBN: 978-981-272-338-3

Copyright © 2012 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 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.

本书简体中文版由机械工业出版社与 Elsevier (Singapore) Pte Ltd. 在中国大陆境内合作出版。本版仅限在中国境内（不包括中国香港、澳门特别行政区及中国台湾地区）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。

封底无防伪标均为盗版

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：01-2009-3517

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4版）/（美）帕特森（Patterson, D. A.），（美）亨尼斯（Hennessy, J. L.）著；康继昌，樊晓桢，安建峰等译．—北京：机械工业出版社，2011.11

（计算机科学丛书）

书名原文：Computer Organization and Design: The Hardware/Software Interface, Fourth Edition
ISBN 978-7-111-35305-8

I. 计… II. ①帕… ②亨… ③康… ④樊… ⑤安… III. 计算机体系结构
IV. TP303

中国版本图书馆 CIP 数据核字（2011）第 136488 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：朱秀英

北京诚信伟业印刷有限公司印刷

2012 年 1 月第 1 版第 1 次印刷

185mm × 260mm · 34.5 印张

标准书号：ISBN 978-7-111-35305-8

ISBN 978-7-89433-047-5（光盘）

定价：99.00 元（附光盘）

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：(010) 88378991；88361066

购书热线：(010) 68326294；88379649；68995259

投稿热线：(010) 88379604

读者信箱：hzsj@hzbook.com

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

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

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



华章教育

华章科技图书出版中心

译者序

Computer Organization and Design: The Hardware/Software Interface, 4E

David A. Patterson 和 John L. Hennessy 是目前国际知名院校计算机专业领域的双巨擘。他们合著的《Computer Organization and Design: The Hardware/Software Interface》又发行了第4版。该书是他们对计算机组织研究和实践的全面而系统的总结。目前，世界上很多大学的计算机原理课程都采用这本教材，国内也有不少大学采用这本教材。

我们认为第4版最主要的特点是强调了计算机从串行到并行的最新变革。本版在每章中都强调了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。本版特别描述了一种评测多核性能的 Roofline 模型，使用 SPEC 2006 程序集更新了所有处理器的性能评测结果。此外，本版还首次描述了面向可视计算优化的高度多线程多处理器 GPU。

感谢清华大学郑纬民教授对前三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。

除封面署名之外，西北工业大学计算机学院的史莉雯、姚涛、任向隆、郑乔石、韩立敏等也参加了本书的翻译和校对工作。由于译者水平有限，文中肯定存在一些翻译不当或理解欠妥的地方，希望读者批评指正。

康继昌

2011年10月于西北工业大学

神秘是一种我们能够体验到的最美丽的东西。
它是所有真正艺术和科学的源泉。

——阿尔伯特·爱因斯坦，《我的世界观》，1930

关于本书

我们认为，在学习计算机科学与工程时，除了掌握计算的基本原理外，还应该了解该领域当今的最新状态。我们也感觉到，计算领域中各种读者希望有机会欣赏到计算机系统的组织范例。后者决定了计算机系统的功能、性能，甚至成功与否。

现代计算机技术需要各种计算方面的专家，他们不仅能理解硬件，而且能理解软件。硬件和软件之间在许多层次上的相互关系，提供了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，计算机组成与设计中的中心思想是相同的。因而，本书着重于展示硬件与软件之间的相互关系，重点介绍概念，这是当今计算机的基础。

最近单处理器已发展为多核微处理器，这也印证了本书自第1版就预测的这一发展前景。有些程序员忽视了这一发展趋势，他们仍希望计算机体系结构专家、编译器编写者和芯片工程师能够帮助他们，让程序不作任何改进就可以更快地运行在新型处理器上。但是，这样的时代已经过去了。为了使程序更快地运行，必须将其并行化。程序员在编程时不用考虑硬件的并行特性，这一目标要很多年才能实现。我们认为，至少在下一个十年里，大多数程序员必须理解软硬件接口，才能使程序在并行计算机上有效地运行。

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，需要理解基本的计算机组成的读者；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，或要进一步理解计算机系统是如何工作的读者。

与本书相关的另一本书

有些读者可能已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”（本书则经常被称为“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。我们以商用产品为例，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，未来软件设计人员对与软件系统一起工作的基本硬件技术的理解程度，将严重影响软件系统的性能和能效。因此，编译器编写者、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提出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应用软件的影响。

所以，本书的内容远多于“Hennessy and Patterson”，而且这些内容已大量修订，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，本书亦是如此。

^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》。

第4版的修订目的

第4版的修订目的包括：第一，描述微处理器的多核革命，全书将贯穿并行软硬件的思想；第二，梳理已有的内容以腾出篇幅介绍并行性；第三，从总体上提高教材水平；第四，更新技术内容，以反映自2004年第3版出版以来业界的新变化；第五，利用当今互联网时代的有利条件，提供了大量有用的练习题。

在详细介绍第4版的修订目的之前，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7章对两种读者是同样重要的。第1章更新了引言部分，增加了功耗重要性和由其引出的微处理器从单核转向多核的讨论，以及性能评价和基准测试程序的相关材料（这在第3版中是独立的一章）。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是重要的阅读材料，特别是想要深入学习编译器和面向对象语言的读者。它包括第3版中第3章的内容，介绍了完整的MIPS体系结构（浮点指令除外）。第3章适合对定点运算或者对浮点运算感兴趣的读者，有些人可能不需要学习第3章，可以跳过去。第4章是把第3版的两章合并起来介绍流水线处理器。其中，4.1节、4.5节和4.10节为关注软件的读者提供了流水线概述。关注硬件的读者将发现第4章提供了流水线处理器的核心技术，读者需要根据自己的专业背景，决定是否首先阅读附录C中提供的逻辑设计部分。第5章和第6章描述的存储器对关注软件的读者是极为重要的，如果时间允许，其他读者也应该尽量深入阅读。第7章介绍了多核、多处理器和集群，是业界最新的内容，每个人都应该阅读。

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 ~ 1.9 * 1.10 (历史)	① ④	① ④
第2章 指令：计算机的语言	2.1 ~ 2.14 * 2.15 (编译器 & Java) 2.16 ~ 2.19 * 2.20 (历史)	① ③ ① ④	② ② ④
附录 E RISC 指令集体系结构	* E.1 ~ E.19	③	
第3章 计算机的算术运算	3.1 ~ 3.9 * 3.10 (历史)	② ④	② ④
附录 C 逻辑设计基础	* C.1 ~ C.13		②
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、异常) 4.10 ~ 4.11 (并行、实例) * 4.12 (Verilog 流水线控制) 4.13 ~ 4.14 (谬误) * 4.15 (历史)	① ② ③ ② ① ④ ① ④	① ① ① ① ① ① ③ ① ④
附录 D 控制通路的硬件实现	* D.1 ~ D.6		③
第5章 大容量和高速度：开发存储器层次结构	5.1 ~ 5.8 5.9 (实现 cache 控制器) 5.10 ~ 5.12 * 5.13 (历史)	① ① ④	① ③ ① ④

(续)

章/附录	节	关注软件	关注硬件
第 6 章 存储器和其他 I/O 主题	6.1 ~ 6.10	①	③
	6.11 (网络)	③	③
	6.12 ~ 6.13	①	③
	6.14 (历史)	④	④
第 7 章 多核、多处理器和集群	7.1 ~ 7.13	①	①
	7.14 (历史)	④	④
附录 A 图形和计算 GPU	A.1 ~ A.12	③	③
附录 B 汇编器、链接器和 SPIM 仿真器	B.1 ~ B.12	⑤	⑤

仔细阅读: ①

有时间可读: ③

作为参考: ⑤

回顾或阅读: ②

拓展阅读: ④

第 4 版修订的第一个目的是使第 3 版位于光盘中作为单独一章的并行性成为本书最为重要的内容, 其中最为明显的例子是第 7 章。特别需要说明的是, 第 7 章引入了 Roofline (屋顶线) 性能模型, 并将之用于对 4 个新型多核体系结构的性能评价。评价结果表明屋顶线模型对于多核微处理器具有相当的洞察力, 可以媲美于 cache 的 3C 模型。

在明确了并行性的重要地位之后, 除了在第 7 章专门讲述并行之外, 本版在前 6 章中的每一章都专门开辟一节强调了并行性。

- 1.6 沧海巨变: 从单处理器向多处理器转变 指出功耗的限制如何迫使业界转向并行性以及并行性为什么是有益的。
- 2.11 并行与指令: 同步 讨论了共享变量的加锁, 尤其是 MIPS 的 Load Linked 和 Store Conditional 指令。
- 3.6 并行性和计算机算术: 结合律 讨论了数值精度与浮点运算的挑战。
- 4.10 并行和高级指令级并行 讨论了各种高级指令级并行 (ILP), 包括超标量、推测和超长指令字 (VLIW)、循环展开和乱序操作 (OOO), 同时也对流水线深度和功耗之间的关系进行了讨论。
- 5.8 并行与存储器层次结构: cache 一致性 讨论了 cache 一致性、连贯性和侦听协议等。
- 6.9 并行性与 I/O: 廉价磁盘冗余阵列。将 RAID 描述成 I/O 系统和高效可用的 ICO 系统。

第 7 章总结了发展并行性的乐观理由, 分析了为何本次并行性的发展应该比过去更加成功。

令我特别高兴的是, NVIDIA 的首席科学家 David Kirk 和首席架构师 John Nickolls 为本版撰写了关于图形处理器 GPU 的附录 A。GPU 是对计算机体系结构的一种新的、有趣的推动, 附录 A 第一次对 GPU 进行了深入介绍。该附录基于本版的并行主题, 提出了一种计算风格: 允许程序员以多指令多数据 (MIMD) 的方式思考, 然而硬件在任何可能的时候仍尽量以单指令多数据 (SIMD) 的风格执行。由于 GPU 价格便宜并且使用广泛——甚至在很多笔记本电脑中都可找到它们——并且它们的编程环境是免费可用的, 所以它们提供了一个可用于许多人进行实验的并行硬件平台。

第二个目的是梳理该书, 以便为介绍并行方面的新内容留出空间。第一步是简单地使用更细致的梳理方式对前三版累积下来的所有段落从前到后进行检查, 看它们是否仍有在书中存在的需要。粗略的改变是章节的合并以及主题的舍弃。Mark Hill 建议舍弃书中多周期处理器的实现这部分内容, 取而代之的是, 在存储器层次的章节中增加有关多周期 cache 控制器的内容。这

使得处理器可以由单独的一章而不是两章来呈现，并且有关处理器的内容通过删除得到了加强。在第3版中单独作为一章的有关性能的内容，在本版中被合并到了第1章。

第三个目的是提升本书的教学方法。现在第1章变得更加充实，内容包括性能、集成电路、功耗，为全书奠定了基础。第2、3章原本以演进的风格进行编写，以“单细胞”的体系结构开始，并以第3章最后的完整MIPS体系结构结束。这种松散的写作风格不能很好地适应现代读者的需要。本版将所有的整型指令集归并到第2章，使第3章成为多数读者可以选读的内容，并且每节各自独立，读者不再需要阅读之前的所有节。因此，与之前版本相比，现在第2章是一个更好的参考资料。由于多周期实现会分散读者的注意力，而处理器现在变为单独的一章，所以第4章的编写效果更好。第5章新增了构建cache控制器的部分，此外，CD中新增的部分包含了该cache的Verilog代码。

第3版配有CD，使得书的页数减少，从而降低了书的价格。而且，有兴趣的读者还可更深入地阅读其中的参考资料。但是，我们积极减少页数的同时，读者却要过于频繁地在书和CD之间来回使用。本版中将不会出现这个问题。现在，CD中有每章的拓展阅读，以及四个章节的更加深入的材料。另外，所有练习都集中在书中，在书和CD之间进行交替使用的次数应该比较少了。

对于那些想知道为什么我们在本书中包含了CD的读者，答案也很简单：CD中包含了那些我们觉得无论读者在哪里都应该很容易并且即刻可以获得的内容。如果你对更进一步的内容感兴趣，或者你想复习一个VHDL教程（举例来说），它就在CD中，可供你使用。CD的另一个特点是，能极大地加强你对材料的学习：它包含了一个搜索引擎，使你可以搜索书中或CD本身的文本中的任何字符串。如果你正在寻找书的索引中没有包含的内容，你可以简单地输入你要搜索的文本和想要显示在搜索结果中的页码。这是一个非常有用的特点，我们希望当你阅读和回顾本书的时候，可以经常使用。

这是一个快速发展的领域，并且对于本书新的版本也是同样的情况，编写新版的一个重要的目的就是更新技术内容。AMD Opteron X4 模型 2356（代码为 Barcelona）用来运行书中第1、4、5和7章的示例。第1、6章增加了SPEC中新的功耗测试程序的结果。第2章增加了ARM体系结构的部分，ARM是当前世界上最流行的32位指令集体系结构。第5章新增了一部分内容介绍虚拟机，其重要性再次呈现出来。第5章对Opteron X4多核的cache性能测量进行了详细的描述，以及对其竞争对手Intel Nehalem（将在本版书出版之后进行发布）的性能测量进行了一些细节描述。第6章第一次描述了Flash存储器，同时也对Sun公司的卓越的小型服务器进行了描述，它包含由8个核、16个DIMM和8个碟片组成的1Ubit的磁盘。第6章还描述了关于长期磁盘失效的最新研究结果。第7章涵盖了有关并行的大量话题，包括多线程、SIMD、向量、GPU、性能模型、测试程序和多处理器网络，并描述了Opteron X4额外的3个多核处理器：Intel Xeon 模型 e5345（Clovertown）、IBM Cell 模型 OS20 和 Sun 微系统 T2 模型 5120（Niagara 2）。

最后的目的是，在这个网络时代，尽量使习题对教师有用，因为布置家庭作业一直是学习资料的一个重要方式。然而，几乎是在本书出现的同时，习题答案就会立刻被贴出。对此，我们采取两种方式。首先，专家撰稿人一直在努力为书中的每一章编写全新的习题。第二，大多数习题都有一个含有多种可供替换的量化参数的表格，这些参数用于回答该问题，这种方式为练习题提供了量化描述支持。对指导教师如何选择布置练习而言，绝对的数量加上灵活性使得学生很难在线找到与习题对应的答案。指导教师还可以按照自己的意愿改变这些量化参数，有效阻止那些依赖互联网寻找固定不变的习题集答案的学生。我们认为这种新方法是对本书有价值的补充，无论你是学生还是教师，请让我们知道对于你来说它的效果如何。

我们保留了以往版本中有用的书本元素。为使本书更好地作为参考书，我们还在新术语第

一次出现的页的页脚放置了定义。书中标题为“理解程序性能”部分的内容用于帮助读者理解他们的程序性能，以及如何进行提高，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。“宏观图”部分仍然存在，以使读者看到整个“森林”而不是每一棵“树”。“小测验”部分通过在每章的最后提供答案，帮助读者在第一时间加强他们对内容的理解。本版同样提供了 MIPS 参考数据，这是从 IBM System/360 得到的灵感，并对可去掉的数据进行了更新，在编写 MIPS 汇编语言程序时，应该是一个方便的参考。

教学支持

我们已收集了大量材料供教师授课使用，包括题解、各章测验、本书的图表、讲义注解和幻灯片等，都可从出版商处获得。如需更多信息，请访问以下网址：

textbooks.elsevier.com/9780123744937

结语

从下面的致谢中，你会知道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：cod4bugs@mkp.com。

本版标志了 Hennessy 和 Patterson 自 1989 年以来长期合作的中止。由于要管理一所世界知名大学，Hennessy 校长将不能继续实质性地承担新版本的编写工作。留下的作者感觉像个总是和伙伴一起演出的演员，突然被推到戏台上独自表演。所以，在致谢名单中列出的人和 Berkeley 的同行们在撰写本书的过程中甚至起了更大的作用。

第 4 版致谢

感谢 **David Kirk**、**John Nickolls** 和他们在 NVIDIA 的同事们（Michael Garland、John Monttrym、Doug Voorhies、Lars Nyland、Erik Lindholm、Paulius Micikevicius、Massimiliano Fatica、Stuart Oberman 和 Vasily Volkov）提供了第一个深入介绍 GPU 的附录 A。再次感谢 Microsoft Research 的 **Jim Larus** 贡献了他在汇编语言方面的专长，并欢迎本书读者使用他开发并维护的仿真器。

也要感谢许多专家的贡献，他们为新版编写了大量新的练习题。写出好的练习题并不是一件容易的任务，在此感谢每位贡献者长期而艰苦地开发具有挑战性并吸引人的题目：

- 第 1 章：**Javier Bruguera** (Universidade de Santiago de Compostela)
- 第 2 章：**John Oliver** (Cal Poly, San Luis Obispo)、**Nicole Kaiyan** (University of Adelaide) 和 **Milos Prvulovic** (Georgia Tech)
- 第 3 章：**Matthew Farrens** (University of California, Davis)
- 第 4 章：**Milos Prvulovic** (Georgia Tech)
- 第 5 章：**Jichuan Chang**、**Jacob Leverich**、**Kevin Lim** 和 **Partha Ranganathan** (均来自 Hewlett-Packard)，以及 **Nicole Kaiyan** (University of Adelaide)
- 第 6 章：**Perry Alexander** (University of Kansas)
- 第 7 章：**David Kaeli** (Northeastern University)

感谢 **Peter Ashenden** 编辑和评价了所有的练习题，并完成本书的 CD 和新的幻灯片的制作。

感谢普林斯顿大学的 **David August** 和 **Prakash Prabhu** 提供了每章测验题。

感谢硅谷的同行们提供了大量新的技术数据：

- **AMD**——Opteron X4 (Barcelona) 的详细数据和测量数据：**William Brantley**、**Vasileios Liaskovitis**、**Chuck Moore** 和 **Brian Waldecker**。

- **Intel**——在 Intel Nehalem 上的预报信息：**Faye Briggs**。
- **Micron**——第 6 章中闪存背景信息：**Dean Klein**。
- **Sun Microsystems**——第 2 章中 SPEC2006 基准测试程序的混合指令测量和第 6 章中 Sun Server x4150 的详细数据和测量数据：**Yan Fisher**、**John Fowler**、**Darryl Gove**、**Paul Joyce**、**Shenik Mehta**、**Pierre Reynes**、**Dimitry Stuve**、**Durgam Vahia** 和 **David Weaver**。
- **U. C. Berkeley**——**Krste Asanovic**（在第 7 章中提供了软件并发与硬件并行的思想），**James Demmel** 和 **Velvel Kahan**（有关并行性和浮点计算的注释），**Zhangxi Tan**（在第 5 章中设计了 cache 控制器及其 Verilog 程序），**Sam Williams**（在第 7 章中提供了屋顶线模型及其多核测量结果的数据），以及我在 **Par Lab** 中的所有同事，他们对全书的并行性主题给出了大量的建议和回馈。

感谢许多教师的贡献，他们回答出版商的问卷调查，评审我们的提议，出席小组会议，对第 4 版计划进行分析和回答。他们是中心组：**Mark Hill**（Wisconsin 大学，Madison），**E. J. Kim**（Texas A&M 大学），**Jihong Kim**（Seoul National 大学），**Lu Peng**（Louisiana 州立大学），**Dean Tullsen**（UC San Diego），**Ken Vollmar**（Missouri 州立大学），**David Wood**（Wisconsin 大学，Madison），**Ki Hwan Yum**（Texas 大学，San Antonio），以及评审：**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 at Stony Brook），**Mac Baker**（Virginia Military Institute），**Ron Barne**（George Mason 大学），**Douglas Blough**（Georgia Institute of Technology），**Kevin Bolding**（Seattle Pacific 大学），**Miodrag Bolic**（Ottawa 大学），**John Bonomo**（Westminster College），**Jeff Braun**（Montana Tech），**Tom Briggs**（Shippensburg 大学），**Scott Burgess**（Humboldt 州立大学），**Fazli Can**（Bilkent 大学），**Warren R. Carithers**（Rochester Institute of Technology），**Bruce Carlton**（Mesa Community College），**Nicholas Carter**（Illinois 大学 at Urbana-Champaign），**Anthony Cocchi**（City 大学 of New York），**Don Cooley**（Utah 州立大学），**Robert D. Cupper**（Allegheny College），**Edward W. Davis**（North Carolina 州立大学），**Nathaniel J. Davis**（Air Force Institute of Technology），**Molisa Derk**（Oklahoma City 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 Gerousis**（Christopher Newport 大学），**Paul Gillard**（Memorial 大学 of Newfoundland），**Michael Goldweber**（Xavier 大学），**Georgia Grant**（College of San Mateo），**Merrill Hall**（The Master's College），**Tyson Hall**（Southern Adventist 大学），**Ed Harcourt**（Lawrence 大学），**Justin E. Harlow**（South Florida 大学），**Paul F. Hemler**（Hempden-Sydney College），**Martin Herbordt**（Boston 大学），**Steve J. Hodges**（Cabrillo College），**Kenneth Hopkinson**（Cornell 大学），**Dalton Hunkins**（St. Bonaventure 大学），**Baback Izadi**（州立大学 of New York—New Paltz），**Reza Jafari**，**Robert W. Johnson**（Colorado Technical 大学），**Bharat Joshi**（North Carolina 大学，Charlotte），**Nargarajan Kandasamy**（Drexel 大学），**Rajiv Kapadia**，**Ryan Kastner**（California 大学，Santa Barbara），**Jim Kirk**（Union 大学），**Geoffrey S. Knauth**（Lycoming College），**Manish M. Kochhal**（Wayne 州立大学），**Suzan Koknar-Tezel**（Saint Joseph's 大学），**Angkul Kongmunvattana**（Columbus 州立大学），**April Kontostathis**（Ursinus College），**Christos Kozyrakis**（Stanford 大学），**Danny Krizanc**（Wesleyan 大学），**Ashok Kumar**，**S. Kumar**（Texas 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**，**Duoglas W. Lynn**（Oregon Institute of Technology），**Yashwant K. Malaiya**（Colorado 州立大学），**Bill Mark**（Texas 大学 at Austin），**Annanda Mondal**（Claflin 大学），**Alvin Moser**（Seattle 大学），**Walid Najjar**（California 大学，Riverside），**Danial J. Neebel**（Loras College），**John Nestor**（Lafayette College），**Joe Oldham**（Centre College），**Timour Paltashev**，**James Parkerson**（Ar-

kansas 大学), Shaunak Pawagi (SUNY at Stony Brook), Steve Pearce, Ted Pedersen (Minnesota 大学), Gregory D Peterson (Tennessee 大学), Dejan Raskovic (Alaska 大学, Fairbanks), 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 College), Carolyn J. C. Schauble (Colorado 州立大学), Keith Schubert (CSU San Bernardino), 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), Philip Snyder (Johns Hopkins 大学), Alex Sprintson (Texas A&M), Timothy D. Stanley (Brigham Young 大学), Dean Stevens (Morningside College), Norza 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), Rama Viswanathan (Beloit College), Guoping Wang (Indiana-Purdue 大学), Patricia Wenner (Bucknell 大学), Kent Wilken (California 大学, Davis), David Wolfe (Gustavus Adolphus College), David Wood (Wisconsin 大学, Madison), Mohamed Zahran (City College of New York), 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 Berkeley 大学的相关人士, 他们为本版最具挑战性的内容 (第 7 章和附录 A) 提供了大量回馈信息。他们是 **Krste Asanovic**、**Christopher Batten**、**Rastilav Bodik**、**Bryan Caltanzaro**、**Jike Chong**、**Kaushik Data**、**Greg Giebling**、**Anik Jain**、**Jae Lee**、**Vasily Volkov** 和 **Samuel Williams**。

感谢 **Mark Smotherman** 一遍又一遍地反复寻找本书中的技术错误和写作错误, 显著改进了这一版的写作质量。鉴于本版像是个人演出, 他所发挥的作用就更为重要了。

感谢 Morgan Kaufmann 公司的 **Denise Penrose** 同意再次出版本书。**Nathaniel McFadden** 是本版的策划编辑, 每周都与我讨论本书内容。**Kimberlee Honjo** 负责组织用户调查和回馈。

感谢 **Dawnmarie Simpson** 管理本书的出版过程, 同时感谢对本版做出贡献的许多自由职业者, 特别是 Multiscience 出版社的 Alan Rose 和 diacriTech 公司, 为本书完成了排版。

以上提到的近 200 名人士为本版提供了大量帮助, 使之成为我所希望的最好的书。

David A. Patterson

目 录

Computer Organization and Design: The Hardware/Software Interface, 4E

出版者的话	
译者序	
前言	
第 1 章 计算机概要与技术	1
1.1 引言	1
1.1.1 计算应用的分类及其特性	2
1.1.2 你能从本书学到什么	3
1.2 程序概念入门	4
1.3 硬件概念入门	7
1.3.1 剖析鼠标	8
1.3.2 显示器	8
1.3.3 打开机箱	9
1.3.4 数据安全	12
1.3.5 与其他计算机通信	13
1.3.6 处理器和存储器制造技术	14
1.4 性能	15
1.4.1 性能的定义	15
1.4.2 性能的测量	17
1.4.3 CPU 性能及其因素	18
1.4.4 指令的性能	19
1.4.5 经典的 CPU 性能公式	19
1.5 功耗墙	21
1.6 沧海巨变: 从单处理器向多处理器 转变	23
1.7 实例: 制造以及 AMD Opteron X4 基准	25
1.7.1 SPEC CPU 基准测试程序	27
1.7.2 SPEC 功耗基准测试程序	28
1.8 谬误与陷阱	29
1.9 本章小结	31
1.10 拓展阅读	32
1.11 练习题	32
第 2 章 指令: 计算机的语言	42
2.1 引言	42
2.2 计算机硬件的操作	43
2.3 计算机硬件的操作数	46
2.3.1 存储器操作数	47
2.3.2 常数或立即数操作数	49
2.4 有符号和无符号数	50
2.5 计算机中指令的表示	54
2.6 逻辑操作	59
2.7 决策指令	61
2.7.1 循环	62
2.7.2 case/switch 语句	64
2.8 计算机硬件对过程的支持	65
2.8.1 使用更多的寄存器	66
2.8.2 嵌套过程	68
2.8.3 在栈中为新数据分配空间	69
2.8.4 在堆中为新数据分配空间	70
2.9 人机交互	72
2.10 MIPS 中 32 位立即数和地址的寻址	75
2.10.1 32 位立即数	75
2.10.2 分支和跳转中的寻址	76
2.10.3 MIPS 寻址模式总结	78
2.10.4 机器语言解码	79
2.11 并行与指令: 同步	81
2.12 翻译并执行程序	83
2.12.1 编译器	84
2.12.2 汇编器	84
2.12.3 链接器	85
2.12.4 加载器	87
2.12.5 动态链接库	87
2.12.6 启动一个 Java 程序	88
2.13 以一个 C 排序程序为例	89
2.13.1 swap 过程	89
2.13.2 sort 过程	90
2.14 数组与指针	95
2.14.1 用数组实现 clear	96
2.14.2 用指针实现 clear	96
2.14.3 比较两个版本的 clear	97
2.15 高级内容: 编译 C 语言和解释 Java 语言	98
2.16 实例: ARM 指令集	98
2.16.1 寻址模式	99
2.16.2 比较和条件分支	100
2.16.3 ARM 的特色	100

2.17 实例: x86 指令集	101	3.11 练习题	173
2.17.1 Intel x86 的改进	101	第4章 处理器	182
2.17.2 x86 寄存器和数据寻址模式	103	4.1 引言	182
2.17.3 x86 整数操作	104	4.1.1 一个基本的 MIPS 实现	183
2.17.4 x86 指令编码	106	4.1.2 实现方式概述	183
2.17.5 x86 总结	107	4.2 逻辑设计惯例	185
2.18 谬误与陷阱	107	4.3 建立数据通路	187
2.19 本章小结	108	4.4 一个简单的实现机制	192
2.20 拓展阅读	110	4.4.1 ALU 控制	192
2.21 练习题	110	4.4.2 主控制单元的设计	194
第3章 计算机的算术运算	135	4.4.3 数据通路的操作	197
3.1 引言	135	4.4.4 控制的结束	199
3.2 加法和减法	135	4.4.5 为什么不使用单周期实现方式	201
3.2.1 多媒体算术运算	137	4.5 流水线概述	202
3.2.2 小结	138	4.5.1 面向流水线的指令集设计	205
3.3 乘法	139	4.5.2 流水线冒险	205
3.3.1 顺序的乘法算法和硬件	139	4.5.3 对流水线概述的小结	210
3.3.2 有符号乘法	141	4.6 流水线数据通路及其控制	211
3.3.3 更快速的乘法	142	4.6.1 图形化表示的流水线	219
3.3.4 MIPS 中的乘法	142	4.6.2 流水线控制	222
3.3.5 小结	142	4.7 数据冒险: 转发与阻塞	225
3.4 除法	143	4.8 控制冒险	234
3.4.1 除法算法及其硬件结构	143	4.8.1 假定分支不发生	234
3.4.2 有符号除法	145	4.8.2 缩短分支的延迟	235
3.4.3 更快速的除法	146	4.8.3 动态分支预测	237
3.4.4 MIPS 中的除法	146	4.8.4 流水线小结	239
3.4.5 小结	147	4.9 异常	240
3.5 浮点运算	148	4.9.1 异常在 MIPS 体系结构中的处理	241
3.5.1 浮点表示	149	4.9.2 在流水线实现中的异常	242
3.5.2 浮点加法	152	4.10 并行和高级指令级并行	245
3.5.3 浮点乘法	154	4.10.1 推测的概念	246
3.5.4 MIPS 中的浮点指令	157	4.10.2 静态多发射处理器	247
3.5.5 算术精确性	162	4.10.3 动态多发射处理器	250
3.5.6 小结	164	4.11 实例: AMD Opteron X4 (Barcelona) 流水线	253
3.6 并行性和计算机算术: 结合律	165	4.12 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	255
3.7 实例: x86 的浮点	165	4.13 谬误与陷阱	255
3.7.1 x86 浮点体系结构	166	4.14 本章小结	256
3.7.2 Intel SIMD 流扩展 2 (SSE2) 浮点体系结构	167	4.15 拓展阅读	257
3.8 谬误与陷阱	168	4.16 练习题	257
3.9 本章小结	170		
3.10 拓展阅读	172		

第5章 大容量和高速度：开发存储器

层次结构	280
5.1 引言	280
5.2 cache 的基本原理	283
5.2.1 cache 访问	285
5.2.2 cache 缺失处理	288
5.2.3 写操作处理	289
5.2.4 一个 cache 的例子：内置 FastMATH 处理器	290
5.2.5 设计支持 cache 的存储系统	292
5.2.6 小结	294
5.3 cache 性能的评估和改进	295
5.3.1 通过更灵活地放置块来减少 cache 缺失	297
5.3.2 在 cache 中查找一个块	300
5.3.3 替换块的选择	302
5.3.4 使用多级 cache 结构减少缺失 代价	302
5.3.5 小结	305
5.4 虚拟存储器	305
5.4.1 页的存放和查找	308
5.4.2 缺页	309
5.4.3 关于写	312
5.4.4 加快地址转换：TLB	312
5.4.5 集成虚拟存储器、TLB 和 cache	315
5.4.6 虚拟存储器中的保护	317
5.4.7 处理 TLB 缺失和缺页	318
5.4.8 小结	322
5.5 存储器层次结构的一般架构	323
5.5.1 问题 1：一个块可以被放在何处	323
5.5.2 问题 2：如何找到一个块	324
5.5.3 问题 3：当 cache 缺失时替换 哪一块	325
5.5.4 问题 4：写操作如何处理	325
5.5.5 3C：一种理解存储器层次结构 行为的直观模型	326
5.6 虚拟机	328
5.6.1 虚拟机监视器的必备条件	329
5.6.2 指令集系统结构（缺乏）对 虚拟机的支持	329
5.6.3 保护和指令集系统结构	329
5.7 使用有限状态机来控制简单的 cache	330

5.7.1 一个简单的 cache	330
5.7.2 有限状态机	331
5.7.3 一个简单的 cache 控制器的 有限状态机	333
5.8 并行与存储器层次结构：cache 一致性	334
5.8.1 实现一致性的基本方案	335
5.8.2 监听协议	335
5.9 高级内容：实现 cache 控制器	336
5.10 实例：AMD Opteron X4 (Barcelona) 和 Intel Nehalem 的存储器层次结构	337
5.10.1 Nehalem 和 Opteron 的存储器 层次结构	337
5.10.2 减少缺失代价的技术	339
5.11 谬误和陷阱	340
5.12 本章小结	342
5.13 拓展阅读	343
5.14 练习题	343
第6章 存储器和其他 I/O 主题	355
6.1 引言	355
6.2 可信度、可靠性和可用性	357
6.3 磁盘存储器	359
6.4 快闪式存储器	362
6.5 连接处理器、内存以及 I/O 设备	363
6.5.1 互联基础	364
6.5.2 x86 处理器的 I/O 互联	365
6.6 为处理器、内存和操作系统提供 I/O 设备接口	366
6.6.1 给 I/O 设备发送指令	367
6.6.2 与处理器通信	368
6.6.3 中断优先级	369
6.6.4 在设备与内存之间传输数据	370
6.6.5 直接存储器访问和内存系统	371
6.7 I/O 性能度量：磁盘和文件系统 的例子	372
6.7.1 事务处理 I/O 基准程序	372
6.7.2 文件系统和 Web I/O 的 基准程序	373
6.8 设计 I/O 系统	373
6.9 并行性与 I/O：廉价磁盘冗余阵列	374
6.9.1 无冗余 (RAID 0)	376
6.9.2 镜像 (RAID 1)	376

6.9.3 错误检测和纠错码 (RAID 2) ...	376	7.15 练习题	429
6.9.4 位交叉奇偶校验 (RAID 3)	376	附录 A 图形和计算 GPU	439
6.9.5 块交叉奇偶校验 (RAID 4)	376	A.1 引言	439
6.9.6 分布式块交叉奇偶校验 (RAID 5)	377	A.1.1 GPU 发展简史	439
6.9.7 P+Q 冗余 (RAID 6)	378	A.1.2 异构系统	440
6.9.8 RAID 小结	378	A.1.3 GPU 发展成了可扩展的 并行处理器	440
6.10 实例: Sun Fire x4150 服务器	379	A.1.4 为什么使用 CUDA 和 GPU 计算	440
6.11 高级主题: 网络	383	A.1.5 GPU 统一了图形和计算	441
6.12 谬误与陷阱	383	A.1.6 GPU 可视化计算的应用	441
6.13 本章小结	386	A.2 GPU 系统架构	441
6.14 拓展阅读	387	A.2.1 异构 CPU-GPU 系统架构	442
6.15 练习题	387	A.2.2 GPU 接口和驱动	443
第 7 章 多核、多处理器和集群	394	A.2.3 图形逻辑流水线	443
7.1 引言	394	A.2.4 将图形流水线映射到统一的 GPU 处理器	443
7.2 创建并行处理程序的难点	396	A.2.5 基本的统一 GPU 结构	444
7.3 共享存储多处理器	398	A.3 可编程 GPU	445
7.4 集群和其他消息传递多处理器	400	A.3.1 为实时图形编程	446
7.5 硬件多线程	403	A.3.2 逻辑图形流水线	446
7.6 SISD、MIMD、SIMD、SPMD 和 向量机	404	A.3.3 图形渲染程序	447
7.6.1 在 x86 中的 SIMD: 多媒体扩展 ...	405	A.3.4 像素渲染示例	447
7.6.2 向量机	406	A.3.5 并行计算应用编程	448
7.6.3 向量与标量的对比	407	A.3.6 使用 CUDA 进行可扩展并行 编程	449
7.6.4 向量与多媒体扩展的对比	408	A.3.7 一些限制	453
7.7 图形处理单元简介	408	A.3.8 体系结构隐含的问题	453
7.7.1 NVIDIA GPU 体系结构简介	410	A.4 多线程的多处理器架构	454
7.7.2 深入理解 GPU	411	A.4.1 大规模多线程	454
7.8 多处理器网络拓扑简介	412	A.4.2 多处理器体系结构	455
7.9 多处理器基准测试程序	415	A.4.3 单指令多线程 (SIMT)	456
7.10 Roofline: 一个简单的性能模型	417	A.4.4 SIMT warp 执行和分支	457
7.10.1 Roofline 模型	418	A.4.5 管理线程和线程块	457
7.10.2 两代 Opteron 的比较	419	A.4.6 线程指令	458
7.11 实例: 使用屋顶线模型评估四种 多核处理器	422	A.4.7 指令集架构 (ISA)	458
7.11.1 4 个多核系统	422	A.4.8 流处理器 (SP)	461
7.11.2 稀疏矩阵	424	A.4.9 特殊功能单元 (SFU)	461
7.11.3 结构化网格	425	A.4.10 与其他多处理器的比较	461
7.11.4 生产率	426	A.4.11 多线程多处理器总结	462
7.12 谬误与陷阱	427	A.5 并行存储系统	462
7.13 本章小结	428	A.5.1 DRAM 的考虑	462
7.14 拓展阅读	429		

A. 5.2	cache	463	A. 10	小结	489
A. 5.3	MMU	463	A. 11	拓展阅读	489
A. 5.4	存储器空间	463	附录 B	汇编器、链接器和 SPIM	
A. 5.5	全局存储器	463	仿真器	490	
A. 5.6	共享存储器	464	B. 1	引言	490
A. 5.7	局部存储器	464	B. 1.1	什么时候使用汇编语言	493
A. 5.8	常量存储器	464	B. 1.2	汇编语言的缺点	493
A. 5.9	纹理存储器	464	B. 2	汇编器	494
A. 5.10	表面	465	B. 2.1	目标文件的格式	495
A. 5.11	load/store 访问	465	B. 2.2	附加工具	496
A. 5.12	ROP	465	B. 3	链接器	498
A. 6	浮点算术	465	B. 4	加载	499
A. 6.1	支持的格式	465	B. 5	内存的使用	499
A. 6.2	基本算术	465	B. 6	过程调用规范	500
A. 6.3	专用算术	466	B. 6.1	过程调用	502
A. 6.4	性能	467	B. 6.2	过程调用举例	503
A. 6.5	双精度	467	B. 6.3	另外一个过程调用的例子	505
A. 7	资料: NVIDIA GeForce 8800	468	B. 7	异常和中断	507
A. 7.1	流处理器阵列 (SPA)	468	B. 8	输入和输出	509
A. 7.2	纹理/处理器簇 (TPC)	469	B. 9	SPIM	511
A. 7.3	流多处理器 (SM)	470	B. 10	MIPS R2000 汇编语言	513
A. 7.4	指令集	471	B. 10.1	寻址方式	514
A. 7.5	流处理器 (SP)	471	B. 10.2	汇编语法	515
A. 7.6	特殊功能单元 (SFU)	471	B. 10.3	MIPS 指令编码	515
A. 7.7	光栅化	471	B. 10.4	指令格式	516
A. 7.8	光栅操作处理器 (ROP) 和 存储系统	471	B. 10.5	常数操作指令	520
A. 7.9	可扩展性	472	B. 10.6	比较指令	520
A. 7.10	性能	472	B. 10.7	分支指令	521
A. 7.11	密集线性代数性能	472	B. 10.8	跳转指令	523
A. 7.12	FFT 性能	473	B. 10.9	陷阱指令	523
A. 7.13	排序性能	474	B. 10.10	取数指令	525
A. 8	资料: 将应用映射到 GPU	474	B. 10.11	保存指令	526
A. 8.1	稀疏矩阵	475	B. 10.12	数据传送指令	527
A. 8.2	在共享存储器中进行缓存	477	B. 10.13	浮点运算指令	528
A. 8.3	扫描和归约	478	B. 10.14	异常和中断指令	532
A. 8.4	基数排序	480	B. 11	小结	533
A. 8.5	GPU 上的 N-Body 应用	482	B. 12	参考文献	533
A. 9	谬误与陷阱	486	B. 13	练习题	533