<<计算机体系结构>>

图书基本信息

书名：<<计算机体系结构>>

13位ISBN编号：9787111364580

10位ISBN编号：7111364589

出版时间：2012-1

出版时间：机械工业出版社

作者：John L. Hennessy,David A. Patterson

<<计算机体系结构>>

内容概要

《计算机体系结构：量化研究方法（英文版·第5版）》堪称计算机系统结构学科的“圣经”，是计算机设计领域学生和实践者的必读经典。
本书系统地介绍了计算机系统的设计基础、存储器层次结构设计、指令级并行及其开发、数据级并行、GPU体系结构、线程级并行和仓库级计算机等。
现今计算机界处于变革之中：移动客户端和云计算正在成为驱动程序设计和硬件创新的主流范型。
因此在这个最新版中，作者考虑到这个巨大的变化，重点关注了新的平台（个人移动设备和仓库级计算机）和新的体系结构（多核和GPU），不仅介绍了移动计算和云计算等新内容，还讨论了成本、性能、功耗、可靠性等设计要素。
每章都有两个真实例子，一个来源于手机，另一个来源于数据中心，以反映计算机界正在发生的革命性变革。
本书内容丰富，既介绍了当今计算机体系结构的最新研究成果，也引述了许多计算机系统设计开发方面的实践经验。
另外，各章结尾还附有大量的习题和参考文献。
本书既可以作为高等院校计算机专业高年级本科生和研究生学习“计算机体系结构”课程的教材或参考书，也可供与计算机相关的专业人士学习参考。

作者简介

John L. Hennessy
斯坦福大学校长，IEEE和ACM会士，美国国家工程研究院院士及美国科学艺术研究院院士。Hennessy教授因为在RISC技术方面做出了突出贡献而荣获2001年的Eckert-Mauchly奖章，他也是2001年Seymour
Cray计算机工程奖得主，并且和本书另外一位作者David A. Patterson分享了2000年John von Neumann奖。

David A. Patterson
加州大学伯克利分校计算机科学系主任、教授，美国国家工程研究院院士，IEEE和ACM会士，曾因成功的启发式教育方法被IEEE授予James
H. Mulligan，Jr.教育奖章。

<<计算机体系结构>>

书籍目录

ForewordPrefaceAcknowledgmentsChapter 1 Fundamentals of Quantitative Design and Analysis1.1 Introduction1.2 Classes of Computers1.3 Defining Computer Architecture1.4 Trends in Technology1.5 Trends in Power and Energy in Integrated Circuits1.6 Trends in Cost1.7 Dependability1.8 Measuring, Reporting, and Summarizing Performance1.9 Quantitative Principles of Computer Design1.10 Putting It All Together: Performance, Price, and Power1.11 Fallacies and Pitfalls1.12 Concluding Remarks1.13 Historical Perspectives and References Case Studies and Exercises by Diana FranklinChapter 2 Memory Hierarchy Design2.1 Introduction2.2 Ten Advanced Optimizations of Cache Performance2.3 Memory Technology and Optimizations2.4 Protection: Virtual Memory and Virtual Machines2.5 Crosscutting Issues: The Design of Memory Hierarchies2.6 Putting It All Together: Memory Hierachies in the ARM Cortex-AS and Intel Core i72.7 Fallacies and Pitfalls2.8 Concluding Remarks: Looking Ahead2.9 Historical Perspective and References Case Studies and Exercises by Norman P. Jouppi, Naveen Muralimanohar, and Sheng LiChapter 3 nstruction-Level Parallelism and Its Exploitation3.1 Instruction-Level Parallelism: Concepts and Challenges3.2 Basic Compiler Techniques for Exposing ILP3.3 Reducing Branch Costs with Advanced Branch Prediction3.4 Overcoming Data Hazards with Dynamic Scheduling3.5 Dynamic Scheduling: Examples and the Algorithm3.6 Hardware-Based Speculation3.7 Exploiting ILP Using Multiple Issue and Static Scheduling3.8 Exploiting ILP Using Dynamic Scheduling, Multiple Issue, and Speculation3.9 Advanced Techniques for Instruction Delivery and Speculation3.10 Studies of the Limitations oflLP3.11 Cross-Cutting Issues: ILP Approaches and the Memory System3.12 Multithreading: Exploiting Thread-Level Parallelism to Improve Uniprocessor Throughput3.13 Putting It All Together: The Intel Core i7 and ARM Cortex-AS3.14 Fallacies and Pitfalls3.15 Concluding Remarks: What's Ahead?3.16 Historical Perspective and References Case Studies and Exercises by Jason D. Bakos and Robert R ColwellChapter4 Data-Level Parallelism in Vector, SIMD, and GPU Architectures4.1 Introduction4.2 Vector Architecture4.3 SIMD Instruction Set Extensions for Multimedia4.4 Graphics Processing Units4.5 Detecting and Enhancing Loop-Level Parallelism4.6 Crosscutting Issues4.7 Putting It All Together: Mobile versus Server GPUS and Tesla versus Core i74.8 Fallacies and Pitfalls4.9 Concluding Remarks4.10 Historical Perspective and References Case Study and Exercises by Jason D. BakosChapter 5 Thread-Level Parallelism5.1 Introduction5.2 Centralized Shared-Memory Architectures5.3 Performance of Symmetric Shared-Memory Multiprocessors……Chapter6 Warehouse-Scale Computers to Exploit Request-Level and Data-Level ParallelismAppendix A Instruction Set PrinciplesAppendix B Review of Memory HierarchyAppendix C Pipelining: Basic and Intermediate Concepts

章节摘录

版权页：插图：The pressure of both a fast clock cycle and power limitations encourages limited size for first-level caches. Similarly, use of lower levels of associativity can reduce both hit time and power, although such trade-offs are more complex than those involving size.The critical timing path in a cache hit is the three-step process of addressing the tag memory using the index portion of the address, comparing the read tag value to the address, and setting the multiplexor to choose the correct data item if the cache is set associative. Direct-mapped caches can overlap the tag check with the transmission of the data, effectively reducing hit time. Furthermore, lower levels of associativity will usually reduce power because fewer cache lines must be accessed.Although the total amount of on-chip cache has increased dramatically with new generations of microprocessors, due to the clock rate impact arising from a larger L1 cache, the size of the L1 caches has recently increased either slightly or not at all. In many recent processors, designers have opted for more associativity rather than larger caches. An additional consideration in choosing the associativity is the possibility of eliminating address aliases; we discuss this shortly.One approach to determining the impact on hit time and power consumption in advance of building a chip is to use CAD tools. CACTI is a program to estimate the access time and energy consumption of alternative cache structures on CMOS microprocessors within 10% of more detailed CAD tools. For a given minimum feature size, CACTI estimates the hit time of caches as cache size varies, associativity, number of read/write ports, and more complex parameters. Figure 2.3 shows the estimated impact on hit time as cache size and associativity are varied.

媒体关注与评论

“ 本书之所以成为永恒的经典，是因为它的每一次再版都不仅仅是更新补充，而是一次全面的修订，对这个激动人心且快速变化领域给出了最及时的信息和最独到的解读。
对于我来说,即使已有二十多年的从业经历，再次阅读本书仍自觉学无止境,感佩于两位卓越大师的渊博学识和深厚功底。
”——Luiz Andre Barroso，Google公司

<<计算机体系结构>>

编辑推荐

<<计算机体系结构>>

版权说明