国外优秀信息科学与技术系列教学用书

计算机系统结构基础

（影印版）

ESSENTIALS OF COMPUTER ARCHITECTURE

■ Douglas E. Comer

PEARSON
Prentice Hall

高等教育出版社
Higher Education Press

国外优秀信息科学与技术系列教学用书

计算机系统结构基础

（影印版）

ESSENTIALS OF COMPUTER ARCHITECTURE

Douglas E. Comer

高等教育出版社

图字:01 -2005 -0964 号

Essentials of Computer Architecture

Douglas E. Comer

原版 ISBN:0 - 13 - 149179 - 2

图书在版编目(CIP)数据

计算机系统结构基础=Essentials of Computer Architecture/(美)科默(Comer, D. E.).—影印本.
北京:高等教育出版社,2005.12
ISBN 7-04-017816-8

Ⅰ.计… Ⅱ.科… Ⅲ.计算机体系结构-高等学校-教材-英文 Ⅳ.TP303

中国版本图书馆 CIP 数据核字(2005)第 132966 号

出版发行	高等教育出版社	**购书热线**	010-58581118
社　　址	北京市西城区德外大街 4 号	**免费咨询**	800-810-0598
邮政编码	100011	**网　　址**	http://www.hep.edu.cn
总　　机	010-58581000		http://www.hep.com.cn
经　　销	北京蓝色畅想图书发行有限公司	**网上订购**	http://www.landraco.com
印　　刷	北京外文印刷厂		http://www.landraco.com.cn
开　　本	787×1092　1/16	**版　　次**	2005 年 12 月第 1 版
印　　张	25	**印　　次**	2005 年 12 月第 1 次印刷
字　　数	480 000	**定　　价**	38.00 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

物料号　17816-00

出 版 说 明

20 世纪末，以计算机和通信技术为代表的信息科学和技术对世界经济、科技、军事、教育和文化等产生了深刻影响。信息科学技术的迅速普及和应用，带动了世界范围信息产业的蓬勃发展，为许多国家带来了丰厚的回报。

进入 21 世纪，尤其随着我国加入 WTO，信息产业的国际竞争将更加激烈。我国信息产业虽然在 20 世纪末取得了迅猛发展，但与发达国家相比，甚至与印度、爱尔兰等国家相比，还有很大差距。国家信息化的发展速度和信息产业的国际竞争能力，最终都将取决于信息科学技术人才的质量和数量。引进国外信息科学与技术优秀教材，在有条件的学校推动开展英语授课或双语教学，是教育部为加快培养大批高质量的信息技术人才采取的一项重要举措。

为此，教育部要求由高等教育出版社首先开展信息科学与技术教材的引进试点工作。同时提出了两点要求，一是要高水平，二是要低价格。在高等教育出版社和信息科学技术引进教材专家组的努力下，经过比较短的时间，第一批由教育部高等教育司推荐的 20 多种引进教材已经陆续出版。这套教材出版后受到了广泛的好评，其中有不少是世界信息科学技术领域著名专家、教授的经典之作和反映信息科学技术最新进展的优秀作品，代表了目前世界信息科学技术教育的一流水平，而且价格也是最优惠的，与国内同类自编教材相当。这套教材基本覆盖了计算机科学与技术专业的课程体系，体现了权威性、系统性、先进性和经济性等特点。

在引进教材的同时，我们还应做好消化吸收，注意学习国外先进的教学思想和教学方法，提高自编教材的水平，使我们的教学和教材在内容体系上，在理论与实践的结合上，在培养学生的动手能力上能有较大的突破和创新。

希望这些教学用书的引进出版，对于提高我国高等学校信息科学技术的教学水平，缩小与国际先进水平的差距，加快培养一大批具有国际竞争力的高质量信息技术人才，起到积极的推动作用。同时也欢迎广大教师和专家们对教材引进工作提出宝贵的意见和建议。联系方式：hep.cs@263.net。

高等教育出版社

二〇〇四年十月

前　言

当我被派去帮忙拯救本科生的计算机系统结构这门课程时，便开始酝酿本书的编写了。计算机系统结构课程遭受了多年的忽视：虽然很多教授和访问学者都讲授过这门课程，但他们往往对数字硬件缺乏兴趣，或缺乏相关的知识背景，从而使这门课程已经退化成各种硬件体系结构相关知识的大杂烩。在一些学期里，学生们将一门课程的所有时间都花在学习布尔代数上，却毫不涉及实际的硬件；而在另外一些学期里，学生们只学习某一种汇编语言晦涩难懂的细节，却完全不了解其他的语言。

那么，计算机系统结构这门课程还值得保留吗？绝对值得保留！在众多计算机科学的教学计划中，计算机系统结构这门课程是学生们了解有关计算机结构的基本概念的唯一机会。对硬件的理解使学生能够编写效率更高、错误更少的程序。广义上讲，体系结构的基本知识有助于理解编程选择的因果关系，从而帮助程序员提高程序效率。此外，了解硬件的工作原理使程序员能够迅速找出程序出错的根源，从而改进编程过程。最后，毕业生们必须理解体系结构的基本概念，才能通过像 Intel 和 Microsoft 这样的公司的招聘测试。

拯救系统结构这门课程的步骤之一就是寻找一本合适的教材。我们发现，现有的教材大致可以分为两类：一类面向将来会从事硬件设计的工程类低年级学生；另一类则面向计算机科学系的学生，并试图囊括编译器、操作系统和网络协议（至少一种）等诸多内容。这些教材都不适合作为单独的入门级计算机系统结构课程的教材。我们需要这样一本教材：（1）重点介绍概念而非工程细节（因为我们的学生并不会去深究硬件设计）；（2）从程序员的角度来阐述系统结构这一主题，并强调其对程序员的重要意义；（3）不应涵盖多门课程的内容。当我们无法找到这样的教材时，唯一的办法就是编写一本。

本书主要分为五个部分。第 I 部分讲述数字逻辑、逻辑门和数据表示的基础知识，重点是数据表示这一章，这是因为二进制补码运算和整数值域的概念是编程的基础。第Ⅱ、第Ⅲ和第Ⅳ部分分别介绍体系结构的三个基本组件：处理器、存储器和 I/O 系统。各章都为学生提供了充足的背景知识，以便理解系统操作机制及其对于程序员的意义。最后，第Ⅴ部分讨论了一些高级论题，例如并行处理、流水线技术以及性能。

附录主要介绍此课程的一个重要组成部分：实验，学生们可以通过实验进行学习。虽然大多数实验问题都要求通过编程解决，但在开始的几个星期内，学生们应该在实验室里将一些门电路焊接在电路板上。这些器材并不贵（平均每个学生花费不到 15 美元，用于购买固定设备；每个学生花费不到 20 美元就可以买到一套属于自己的芯片）。

我们已经建立了本书的辅助网站：

http://www.eca.cs.purdue.edu

Rajesh Subraman 已经同意维护该网站，其中包括了一系列由作者与 Rajesh 分别编写的课堂教学材料。我们还邀请了其他教师贡献出自己的教学材料。

本书和实验练习已经在普度大学采用，学生们给予了极高的评价。我们收到了许多关于本书及课程的感谢留言。许多学生在实验室第一次接触到了硬件，并且表现出极大的热情。

我要感谢许多对此书的出版做出贡献的人。Bernd Wolfinger 对此书进行了广泛的审阅，并对本书的主题和方向提出了诸多重要的建议。Dan Ardelean、James Cernak 和 Tim Korb 对许多章节进行了详细的评论。Dave Capka 审阅了前面几章的内容。Rajesh Subraman 采用本书进行教学并对书中内容提出了自己的建议。普度大学计算机科学系 250 班的下列学生分别发现了手稿中的一个或多个错误：Nitin Alreja、Alex Cox、David Ehrmann、Roger Maurice Elion、Andrew Lee、Stan Luban、Andrew L. Soderstrom 和 Brandon Wuest。

最后，我要感谢我的妻子 Chris，她耐心而仔细的编辑工作以及颇有见地的建议，为本书增色不少。

Douglas E. Comer
2004 年 6 月

关 于 作 者

Douglas Comer 博士在计算机系统领域有着广泛的知识背景，而且一直从事着软件和硬件的教学与研究工作。Comer 在软件方面的研究工作横跨计算机系统的各个领域，包括编译器和操作系统。他构建过一个完整的操作系统，包括进程管理器、内存管理器以及串行接口和并行接口的设备驱动程序。Comer 也为常规计算机和网络处理器编写过网络协议软件和网络设备驱动程序。他开发的操作系统 Xinu 和 TCP/IP 协议栈已经被应用于商业产品中。

Comer 的硬件工作经验包括设计离散元件、逻辑门电路以及基本的硅工艺经验。他已经编写了几本畅销的关于网络处理器体系结构的教材。在贝尔实验室，Comer 研究过 VLSI 设计并设计了一种 VLSI 芯片。

Comer 是普度大学计算机科学系的著名教授，他致力于计算机系统结构、操作系统、网络和 Internet 方面的开发、教学和研究。Comer 建立了几个创新实验室，使得学生可以建立和测试各种系统，例如操作系统和 IP 路由器；Comer 的所有课程都包括实验。他还不断地在世界各地的大学、工业界和会议上进行咨询和演讲。

除了编写一系列享誉世界的计算机操作系统、网络、TCP/IP 和计算机技术等方面的书籍外，Comer 还兼任 *Software: Practice and Experience* 杂志的主编。他是 ACM 成员、普度大学教学学会成员，获得过无数嘉奖，其中包括 USENIX 终身成就奖（USENIX Lifetime Achievement Award）。

关于其本人的更多信息可查阅网站：

www.cs.purdue.edu/people/comer

关于 Comer 著作的信息可查阅网站：

www.comerbooks.com

Preface

This book began when I was assigned to help salvage an undergraduate computer organization course. The course had suffered years of neglect: it had been taught by a series of professors, mostly visitors, who had little or no interest or background in digital hardware, and the curriculum had deteriorated to a potpourri of topics that were only loosely related to hardware architectures. In some semesters, students spent the entire class studying Boolean Algebra, without even the slightest connection to actual hardware. In others, students learned the arcane details of one particular assembly language, without a notion of alternatives.

Is a computer organization course worth saving? Absolutely! In many Computer Science programs, the computer organization course is the only time students are exposed to fundamental concepts that explain the structure of the computer they are programming. Understanding the hardware makes it possible to construct programs that are more efficient and less prone to errors. In a broad sense, a basic knowledge of architecture helps programmers improve program efficiency by understanding the consequences of programming choices. Knowing how the hardware works can also improve the programming process by allowing programmers to pinpoint the source of bugs quickly. Finally, graduates need to understand basic architectural concepts to pass job application tests given by firms like Intel and Microsoft.

One of the steps in salvaging our architecture course consisted in looking at textbooks. We discovered the texts could be divided into roughly two types: texts aimed at beginning engineering students who would go on to design hardware, and texts written for CS students that attempt to include topics from compilers, operating systems, and (in at least one case) a complete explanation of how Internet protocols operate. Neither approach seemed appropriate for a single, introductory course on the subject. We wanted a book that (1) focused on the concepts rather than engineering details (because our students are not focused on hardware design); (2) explained the subject from a programmer's point of view, and emphasized consequences for programmers; and (3) did not try to cover several courses' worth of material. When no text was found, it seemed that the only solution was to create one.

The text is divided into five parts. Part 1 covers the basics of digital logic, gates, and data representation. We emphasize the representation chapter because notions of two's-compliment arithmetic and ranges of integer values are essential in programming. Parts 2, 3, and 4 cover the three essential areas of architecture: processors, memories, and I/O systems. In each case, the chapters give students enough background to under-

stand how the mechanisms operate and the consequences for programmers. Finally, Part 5 covers advanced topics like parallelism, pipelining, and performance.

An Appendix describes an important aspect of the course: a hands-on lab where students can learn by doing. Although most lab problems focus on programming, students should spend the first few weeks in lab wiring a few gates on a breadboard. The equipment is inexpensive (we spent less than fifteen dollars per student on permanent equipment; students purchase their own set of chips for under twenty dollars).

We have set up a web site to accompany the book at:

http://www.eca.cs.purdue.edu

Rajesh Subraman has agreed to manage the site, which contains a set of class presentation materials created by the author as well as a set created by Rajesh. We invite other instructors to contribute their materials.

The text and lab exercises have been used at Purdue; students have been extremely positive about both. We received notes of thanks for the text and course. For many students, the lab is their first experience with hardware, and they are enthusiastic.

My thanks to the many individuals who contributed to the book. Bernd Wolfinger provided extensive reviews and made several important suggestions about topics and direction. Dan Ardelean, James Cernak, and Tim Korb gave detailed comments on many chapters. Dave Capka reviewed early chapters. Rajesh Subraman taught from the book and provided his thoughts about the content. In the CS 250 class at Purdue, the following students each identified one or more typos in the manuscript: Nitin Alreja, Alex Cox, David Ehrmann, Roger Maurice Elion, Andrew Lee, Stan Luban, Andrew L. Soderstrom, and Brandon Wuest.

Finally, I thank my wife, Chris, for her patient and careful editing and valuable suggestions that improve and polish each book.

Douglas E. Comer

June, 2004

About The Author

Dr. Douglas Comer has an extensive background in computer systems, and has worked with both hardware and software. Comer's work on software spans most aspects of systems, including compilers and operating systems. He created a complete operating system, including a process manager, a memory manager, and device drivers for both serial and parallel interfaces. Comer has also implemented network protocol software and network device drivers for conventional computers and network processors. Both his operating system, Xinu, and TCP/IP protocol stack have been used in commercial products.

Comer's experience with hardware includes work with discrete components, building circuits from logic gates, and experience with basic silicon technology. He has written popular textbooks on network processor architectures, and at Bell Laboratories, Comer studied VLSI design and fabricated a VLSI chip.

Comer is a Distinguished Professor of Computer Science at Purdue University, where he develops and teaches courses and does research on computer organization, operating systems, networks, and Internets. Comer has created innovative laboratories in which students can build and measure systems such as operating systems and IP routers; all of Comer's courses include hands-on lab work. He continues to consult and lecture at universities, industries, and conferences around the world.

In addition to writing a series of internationally acclaimed technical books on computer operating systems, networks, TCP/IP, and computer technologies, Comer serves as the editor-in-chief of the journal *Software — Practice and Experience.* He is a Fellow of the ACM, a Fellow of the Purdue Teaching Academy, and a recipient of numerous awards, including a Usenix Lifetime Achievement award.

Additional information can be found at:

www.cs.purdue.edu/people/comer

and information about Comer's books can be found at:

www.comerbooks.com

Contents