HZ BOOKS
华章教育

计 算 机 科 学 丛 书

原书第4版

计算机系统

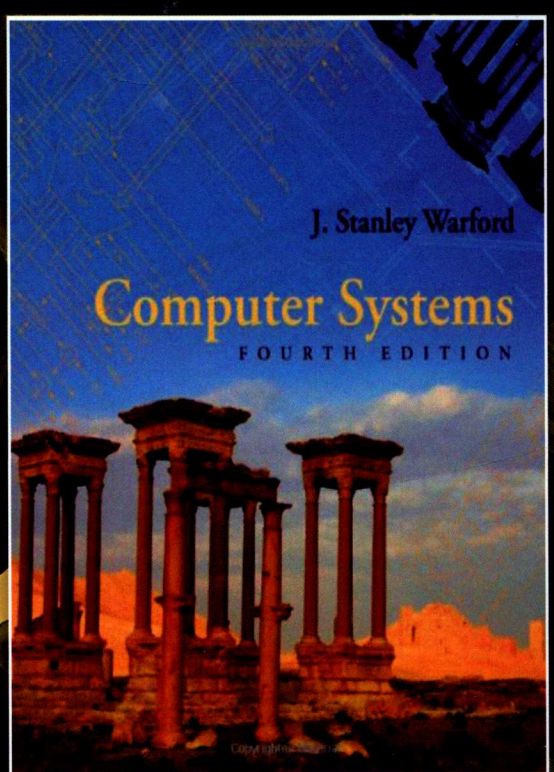
核心概念及软硬件实现

[美] J. 斯坦利·沃法德（J. Stanley Warford） 著
佩珀代因大学

龚奕利 译
武汉大学

Computer Systems

Fourth Edition



机械工业出版社
China Machine Press

计 算 机 科 学 丛 书

原书第4版

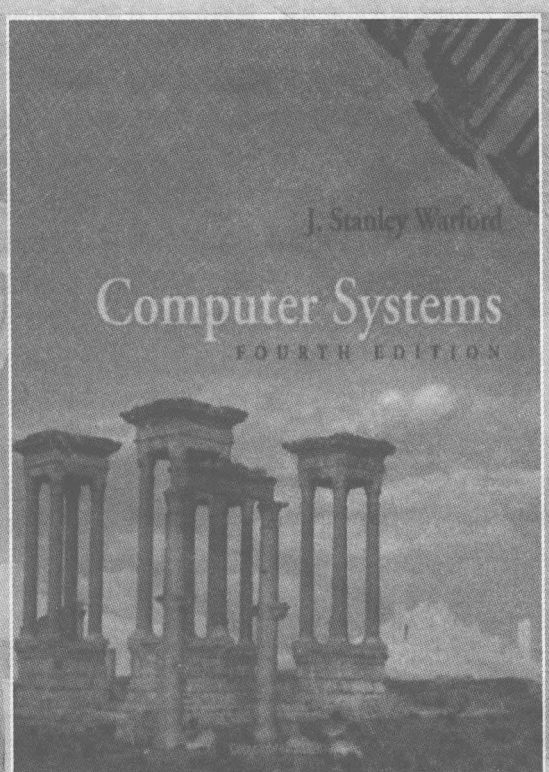
计算机系统

核心概念及软硬件实现

[美] J. 斯坦利·沃法德（J. Stanley Warford） 著 佩珀代因大学
龚奕利 译 武汉大学

Computer Systems

Fourth Edition



机械工业出版社
China Machine Press

图书在版编目（CIP）数据

计算机系统：核心概念及软硬件实现（原书第4版）/（美）沃法德（Warford, J. S.）著；龚奕利译．—北京：机械工业出版社，2015.7
（计算机科学丛书）
书名原文：Computer Systems, Fourth Edition

ISBN 978-7-111-50783-3

I. 计… II. ① 沃… ② 龚… III. 计算机系统 IV. TP30

中国版本图书馆CIP数据核字（2015）第150961号

本书版权登记号：图字：01-2013-5990

本书基于虚构的计算机Pep/8，清晰、详细、循序渐进地介绍了计算机组成、汇编语言和计算机体系结构中的核心思想，围绕7个抽象层次组织内容，详细介绍了计算机系统的应用层、高级语言层、汇编层、操作系统层、指令集架构层、微代码层和逻辑门层。本书有完整的程序示例，理论和实践相结合，宽度和深度相结合，提供了对普适的冯·诺依曼机器架构的深入理解。

本书可作为高等院校计算机科学专业本科生的教材，也可作为相关专业人员学习计算机基础知识的参考书。

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：盛思源 曲 熠 责任校对：董纪丽
印 刷：三河市宏图印务有限公司 版 次：2015年7月第1版第1次印刷
开 本：185mm×260mm 1/16 印 张：31
书 号：ISBN 978-7-111-50783-3 定 价：79.00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

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，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

华章科技图书出版中心

It has been several years since the publication of the Fourth Edition of *Computer Systems*.

During that time computer technology has continued to advance. The natural question for this translation is, Why is *Computer Systems* still current after these years since its English publication? There are two reasons why it is current and will remain current for many years.

First, *Computer Systems* is still current because it teaches the fundamentals of computer systems that have been constant for decades. All computer systems consist of software and hardware, and all commercial hardware systems are based on the ubiquitous von Neumann cycle.

On the software side, all high-order languages manipulate data with iterative and recursive algorithms. All languages must be translated to lower level machine language to be executed or interpreted. On the hardware side, all physical machines are combinational and sequential circuits built from logic gates. No matter what the technology, the design principles behind these software and hardware systems do not change.

Second, *Computer Systems* is still current because it teaches the above design principles with the Pep/8 virtual machine. The Pep/8 machine will always be current because it is not subject to obsolescence, but rather is built to teach the fundamentals of computer systems that do not change. Specifically, it presents a computer system as seven levels of abstraction:

- Applications
- High-order languages
- Assembly
- Operating system
- Instruction set architecture
- Microcode
- Logic gate

The book illustrates these levels of abstraction using C++ as the high-order language and Pep/8 assembly language and machine language at the lower levels. The strength of this approach is that the central concepts of computer science are taught without getting entangled in the many irrelevant details that often accompany courses that focus on current technology rather than computing fundamentals. Students who learn the fundamentals are better equipped to master whatever new technology they will encounter in their future computing careers than they would be if they studied only the technology that is current when they are students.

本书第 4 版完成已经有几年的时间了，这期间计算机技术持续发展。要出版中文版，一个很自然的问题就是：为什么英文版出版多年之后本书的内容仍然没有过时呢？它现在没有过时，今后许多年也不会过时，原因有两点。

首先，本书讲授计算机系统的基础知识，这些基础知识数十年都没有改变过。所有的计算机系统都是由软件和硬件组成的，所有的商用硬件系统都是基于普适的冯·诺依曼周期。

从软件方面来说，所有的高级语言都以迭代或递归算法来处理数据。所有的语言都必须翻译成更低级的机器语言才能执行或解释。从硬件方面来说，所有的物理机器都是由逻辑门构成的组合或时序电路。无论技术如何发展，这些软件和硬件系统背后的设计原理没有改变。

其次，本书以 Pep/8 虚拟机为例来讲授上述设计原理。Pep/8 机器不会过时，因为它不受时间限制，设计它的初衷就是讲授计算机系统中不会变化的基础知识。具体来说，它展现的是计算机系统在 7 个抽象层次上的样子：

- 应用层
- 高级语言层
- 汇编层
- 操作系统层
- 指令集架构层
- 微代码层
- 逻辑门层

本书在描述各个抽象层次时使用 C++ 作为高级语言，在较低层上使用 Pep/8 汇编语言和机器语言。这种方法的好处就是讲授计算机科学的核心概念，而又不会纠结于无关的细节，很多把重点放在当前技术而不是计算基础的课程通常都会有这种问题。比起只学习当前技术的学生，学习了基础知识的学生在以后的工作中能更好地掌握遇到的新技术。

J. Stanley Warford

译者序

Computer Systems，Fourth Edition

本书第 1 版出版于 2010 年。在不断感叹计算机技术发展太快的时候，2010 年的内容会不会太“老”太“旧”了呢？答案显然是否定的，尤其在读完了整本书之后。本书展示了计算机系统的 7 个抽象层次：应用层、高级语言层、汇编层、操作系统层、指令集架构层、微代码层和逻辑门层。本书的特色之一就是着眼于计算机软件和硬件系统背后的设计原理，而这些原理数年来都未曾改变过。而且，去除那些眼花缭乱的新技术的表象，能够更好地看清和理解系统的本质。

本书的另一个特色是建立了一个虚构的计算机系统 Pep/8，借助于这个示例系统，能够让读者 / 学生更具体地了解计算机系统中的各个组成部分，进而了解核心概念，而不是通过抽象的描述学习抽象的概念；同时又不必拘泥于现实系统的实现细节。本书覆盖广泛，但又重点突出，强调了硬件及其相关软件的实现。本书文字简洁明了，是非常合适的计算机系统入门教材。

与所有计算机书籍的翻译一样，翻译过程中充满了艰难的术语选择，因为我们越来越习惯于在日常技术工作中使用英文术语，有时候使用它的中文翻译反而显得有些陌生和别扭。比如 token，中文一般译作“语言符号”，但是实际上它有终结符字符串的含义，如果翻译成中文，很容易失去这些意义。所以我选择保留英文术语，相信不会影响读者阅读或增加阅读难度。

在此感谢王文杰帮助我一起讨论翻译中遇到的问题，还要感谢机械工业出版社华章公司的编辑们给了我很多理解和支持，使得本书得以完成。

在翻译过程中，我尽量做到认真仔细，但还是难以避免出现错误和不尽如人意的地方。在此欢迎广大读者批评指正，我也会把勘误表及时在网上更新，便于大家阅读。

龚奕利

2015 年 5 月于安娜堡

本书清晰、详细、循序渐进地展示了计算机组成、汇编语言和计算机体系结构中的核心思想。本书的很大一部分是建立在一个虚构的计算机 Pep/8 基础上的，用它来讲解经典的冯·诺依曼机器的基本概念。这种方法的好处是能够讲解计算机科学的核心概念，而又不必拘泥于此类课程中常见的许多不相关的细节。这种方法还能鼓励学生思考计算机科学底层的原理。本书的范围也比较广泛，重点强调了与硬件及其相关软件的处理有关而少有提及的计算机科学主题。

内容摘要

计算机运行在一些抽象层上，在高级抽象层上编程只是一部分。基于图 1 的层次结构，本书展示了计算机系统的一个统一的概念。

图 1 典型计算机系统的层次结构

对应于图 1 的 7 层，本书也分为 7 个部分：

App7 层　应用层
HOL6 层　高级语言层
ISA3 层　指令集架构层
Asmb5 层　汇编层
OS4 层　操作系统层
LG1 层　逻辑门层
Mc2 层　微代码层

本书主要是按照从上到下、从最高层到最低层的顺序来书写。ISA3 层在 Asmb5 层之前以及 LG1 层在 Mc2 层之前讲解是出于教学的目的。在这两种情况下，暂时用相反的从下至上的方法来讲解更自然，有了低层的构造模块就很容易完成上层的构建。

App7 层　App7 层是单独一章，介绍了应用程序。本章展示了抽象层次的概念，建立本书剩下部分的框架。还介绍了一些关系数据库的概念，作为典型计算机应用的例子。同时，还假设学生对文字编辑器或文字处理器有一定的经验。

HOL6 层　HOL6 层也是一章，复习了 C++ 编程语言。本章假设学生具有某种命令语言的经验，不一定是 C++，可以是 Java 或 C。书中避免了 C++ 的高级特性，包括面向对象的概念。如果有必要，教师可以把 C++ 例子翻译成其他 HOL6 层的语言。

本章着重介绍了 C++ 内存模型，包括全局变量和局部变量、函数参数以及动态分配的变量。也介绍了递归的问题，因为它依赖于运行时栈上的内存分配机制。还相当详细地解释了函数调用的内存分配过程，因为本书后面还会在较低抽象层次上分析这个机制。

ISA3 层　ISA3 层是指令集架构层，包括两章，描述了一个用于说明计算机概念的虚构的 Pep/8 计算机。Pep/8 是经典的冯·诺依曼机器。CPU 包含一个累加器、一个变址寄存器、一个程序计数器、一个栈指针和一个指令寄存器。有 8 种寻址方式：立即数、直接、间接、栈相对、栈相对间接、变址、栈变址和栈变址间接。在模拟的只读存储器（ROM）中，Pep/8 操作系统能从学生的文本文件中装入和执行十六进制格式的程序。学生可以在 Pep/8

模拟器上运行小程序，学习不会改变内存值的 ROM 存储指令。

学生能学习到位层的信息表示和计算机组成的知识。因为本书的中心主题是层次之间的关系，所以有关 Pep/8 的章节展示了 ASCII 表示（ISA3 层）和类型为 `char` 的 C++ 变量（HOL6 层）之间的关系。还展示了补码表示（ISA3 层）和类型为 `int` 的 C++ 变量（HOL6）之间的关系。

Asmb5 层 Asmb5 层是汇编层，书中介绍了汇编器的概念（汇编器是汇编层和机器层之间的翻译器），还介绍了 Asmb5 层的符号和符号表。

这里是统一的方法派上用场的地方。第 5 章和第 6 章中的编译器是高级语言到汇编语言的翻译器。前面，学生学习了一种具体的 HOL6 层语言 C++ 和一种具体的冯·诺依曼机器 Pep/8。接下来的章节将继续介绍层次之间的关系，讲述下面这样一些对应关系：（a）HOL6 层的赋值语句和 Asmb5 层的装入 / 存储指令；（b）HOL6 层的循环和 `if` 语句与 Asmb5 层的分支指令；（c）HOL6 层的数组和 Asmb5 层的变址寻址；（d）HOL6 层的过程调用和 Asmb5 层的运行时栈；（e）HOL6 层的函数和过程参数与 Asmb5 层的栈相对寻址；（f）HOL6 层的 `switch` 语句和 Asmb5 层的转移表；（g）HOL6 层的指针和 Asmb5 层的地址。

统一方法之美就在于可以在较低层次上实现 C++ 章节中的例子。例如，第 2 章的递归例子中描述的运行时栈直接对应于 Pep/8 主存中的硬件栈。学生可以用手动方式直接在两层之间翻译，以便更好地理解编译的过程。

这种方法为讨论计算机科学中的核心问题提供了一种很自然的环境。例如，本书介绍了 HOL6 层的结构化编程，可以和 Asmb5 层的非结构化编程的可能性进行对比。书中讨论了 goto 争议、结构化编程 / 效率之间的折中，给出了两个层次上语言的实际例子。

第 7 章向学生介绍了计算机科学理论。既然学生对如何把高级语言翻译成汇编语言已经有了感性的认识，那么我们就提出所有计算中最基本的问题：什么是能够被自动化的？这里介绍理论是很自然的，因为学生现在知道了编译器（自动化翻译器）必须做什么。他们通过识别 C++ 和 Pep/8 汇编语言的语言符号来学习语法分析和有限状态机——确定性的和非确定性的。本章包括两种小语言之间的自动翻译器，说明了词法分析、语法分析和代码生成。词法分析器是有限状态机的实现。还有什么比这样更自然的介绍理论的方法呢？

OS4 层 OS4 层讲述操作系统，分为两章。第 8 章讲述进程管理，包括两节，一节讲装载器，一节讲陷阱处理程序，说明了 Pep/8 操作系统的概念。有 5 条指令具有产生软件陷阱的未实现操作码。操作系统把用户正在运行的进程的进程控制块存储在系统栈上，中断服务例程解释该指令。通过具体实现一个挂起进程来强化操作系统中运行和等待进程的经典状态转移图。本章结尾描述了并发进程和死锁。第 9 章描述存储管理，包括主存和磁盘存储器。

LG1 层 LG1 层用两章来介绍组合电路和时序电路。从布尔代数的定理开始，第 10 章重点介绍计算机科学的数学基础的重要性，展示布尔代数和逻辑门之间的关系，然后介绍一些常见的 SSI 和 MSI 逻辑设备，包括 Pep/8 ALU 的完整的逻辑设计。第 11 章通过介绍时序电路的状态转移图，描述有限状态机的基本概念。最后描述常见的计算机子系统，比如双向总线、内存芯片和双端口存储器体。

Mc2 层 第 12 章描述 Pep/8 CPU 的微程序设计控制区，给出了一些示例指令和寻址方式的控制序列，还提供了有关其他指令和寻址方式的大量练习。本章还介绍了装入 / 存储体系结构的概念，对比了 MIPS 的 RISC 机器和 Pep/8 CISC 机器。最后描述了高速缓存、流水线、动态分支预测和超标量机器，介绍了一些性能问题。

在课程中的使用

本书覆盖的内容广泛，教师在设计课程时可以省略一些内容。第 1 ～ 5 章可以看作核心，第 6 ～ 12 章可以有所取舍。

本书第 1 ～ 5 章必须顺序讲授，第 6 章和第 7 章可以按任意顺序讲授。我通常会省略第 6 章而直接讲第 7 章，开始一个大的软件项目——为 Pep/8 汇编语言的一个子集写一个汇编器，这样学生在一学期中有足够的时间完成它。第 11 章显然依赖于第 10 章，但是它们都不依赖于第 9 章，所以第 9 章可以省略。图 2 是一个章节依赖图，图中总结了可以省略哪些章节。

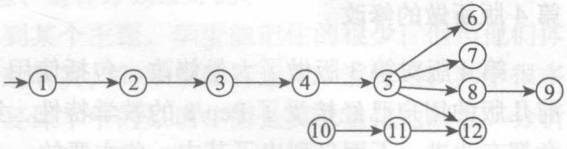


图 2　章节依赖图

辅助资料

下面列出的辅助资料可以从出版社网站获得：

http://www.jbpub.com/catalog/9780763771447/

Pep/8 汇编器和模拟器　Pep/8 机器在 MS Windows、Mac OS X 和 UNIX/Linux 系统上都有。汇编器的特性包括：

- 集成的文字编辑器；
- 在源代码中发现错误的地方插入红色字体的错误消息；
- 对学生友好的、十六进制格式的机器语言目标代码；
- 能够直接以机器语言编写代码，跳过汇编器；
- 能够重定义触发同步陷阱的未实现操作码的助记符。

模拟器的特性包括：

- 模拟的 ROM，装入指令不能修改；
- 在模拟的 ROM 中烧入了一个小的操作系统，包括一个装载器和一个陷阱处理程序；
- 一个集成的调试器，允许设置断点、单步执行、CPU 跟踪和内存跟踪；
- 用户定义的、从无限循环恢复的语句执行计数的上限；
- 能够通过为未实现操作码设计新的陷阱处理程序来修改操作系统。

Pep/8 CPU 模拟器　CPU 模拟器，有 MS Windows、Mac OS X 和 UNIX/Linux 系统版本，可以用在计算机组成课程中。CPU 模拟器的特性包括：

- 颜色编码的展示通路，根据控制信号跟踪复用器的数据流；
- 操作的单周期模式，用 GUI 输入每个控制信号，立即展示信号的效果；
- 操作的多周期模式，学生可以在集成的文字编辑器中编写 Mc2 微代码序列并执行它们以便实现 ISA3 指令。

课程课件　每章有 50 ～ 125 页的课程幻灯片，有 Keynote 和 PDF 格式。幻灯片包括课本中所有的图和总结信息，通常以标号的形式给出。不过其中没有太多的例子，给教师展示示例和教师指导讨论留出了空间。

考试题目　提供有一组考试题目，包括参考信息，例如 ASCII 表、指令集表等，供考试和自学之用。这些对用本书作为教材的教师开放。

数字电路实验　有一组 6 个数字电路实验，能够让学生在物理实验电路板上亲身体验。这些实验说明了第 10 章和第 11 章的组合和时序设备，使用许多本书中没讲到的电路。学生

可以自学实际的电子设计和实现概念，这些超出了本书的讲述范围，它们可以按照书中讨论的主题顺序，从组合电路开始，然后是时序电路和 ALU。

答案手册 附录中有部分练习的答案。剩下练习的答案对用本书作为教材的教师开放。出于安全原因，答案直接从出版社获取。相关信息请联系 Jones 和 Bartlett Publishers Representative，电话 1-800-832-0034。

第 4 版所做的修改

第 4 版对第 3 版做了大量修改，包括使用了 Pep/8，它是对 Pep/7 架构的彻底重新设计。前几版的用户已经接受了 Pep/8 的教学特性，第 4 版中还是保留了 Pep/8 架构。本版的每一章都有改进，下面只列出了其中一些主要的：

- *改进了 C++ 回顾*——扩展了第 3 版中引入的 C++ 内存模型，更系统地从头开始描述。内存分配图更现实，与主函数一致，显示了对主函数返回地址和返回值的分配。重命名了所有以前命名为 i 的变量。当程序翻译成 Pep/8 汇编语言后容易有误会，Pep/8 汇编语言用字母 i 表示立即数寻址。
- *改进了字符编码的内容*——讲述了 Unicode 字符集，代替了 EBCDIC。
- *跟踪标签*——Pep/8 汇编器和模拟器包括一个新的符号跟踪特性，当用户单步跟踪程序时，能够实时显示全局变量和运行时栈。使用这个新特性要求程序员在某些汇编语言语句的注释字段放置跟踪标签，翻译器将忽略这些标签，但是调试器将使用它们。跟踪标签的一个巨大好处是迫使程序员做好文件说明。要使用调试器，学生必须在注释字段精确说明哪些变量要在运行时栈上分配以及分配的顺序。汇编器验证分配的字节数是否与变量列表要求的字节数匹配。跟踪标签的文件说明优点很大，现在这个版本中描述了跟踪标签语法，书中和答案手册中的每个汇编语言程序中都包括了跟踪标签。
- *改进了语言翻译的内容*——前面版本中，第 7 章讲述的语言翻译原理假设学生没有面向对象知识。本版假设学生已经学习过基本的面向对象设计原理，展示的语法分析程序使用了对象组合、继承和多态调度以及 UML 图。
- *新的项目问题*——这一版本有两个项目问题，一个是新的，在第 6 章中，要求写一个 Pep/8 机器模拟器；另一个是改进的，在第 7 章中，要求写一个 Pep/8 汇编器。这两个项目要求开发上千行代码的程序，它们都有多个部分，每一个都往应用程序中增加了更多的功能。项目的目的有两个：1）让学生获得编写较复杂程序的经验；2）增强对这门课程问题域中计算机系统概念的理解。
- *改进了 RAID 的内容*——这个版本介绍了更广泛的 RAID 磁盘系统内容，扩展了 RAID 等级 01 和 10 的区别，增加了新的图和新的量化分析习题。
- *改进了 MIPS 的内容*——扩展了 MIPS 的内容，更系统地比较了 CISC 架构的 Pep/8 和 RISC 架构的 MIPS。新的 MIPS 章节用新的指令集表描述了 5 种寻址方式。MIPS 机器的数据区的图包括了伪直接寻址方式所要求的数据通路和复用器。明确命名的控制信号使用与 Pep/8 控制信号相同的语法，提供了 MIPS 指令实现更简洁而详细的描述。

独特的特性

本书有几个独特的特性，使之有别于其他计算机系统、汇编语言和计算机组成的书。

- *概念的方法*——许多教科书试图跟上领域的变化，包括最新的技术发展。例如，最新外围设备的通信协议规范。这类书通常通篇是“设备如何工作”的描述性解释。本书避开了这类资料，而只选择基础的计算概念，掌握了这些就有了理解当前和未来技术的基础。例如，以数字电路设计问题来说，让学生掌握空间 / 时间折中的概念比简单地阅读通用描述更重要。再举一个例子，通过学习如何在 ISA 指令的微代码实现中合并周期来掌握硬件并行的概念，这样才是最好的。
- *强调问题解决*——如果只听说或者读到某个主题，学生能记住的很少；但当他们体验到时，才会记住很多。本书强调问题解决，每章后面有近 400 道练习，其中很多有多个部分。这些练习不会让学生重复课本中的原话，而是要求量化地解答、分析或者设计系统某个抽象层次上的一个程序或电子电路。
- *一致的机器模型*——Pep/8 机器是一个小型的 CISC 计算机，是描述系统所有层次的载体。学生可以清晰地看到抽象层次之间的关系，因为他们要在所有的层次上为这个机器编程或者设计电子电路。例如，当在 LG1 层设计 ALU 组件时，他们知道 ALU 在 ISA3 层的实现中应该在哪个位置。通过像编译器那样把 C++ 程序翻译成汇编语言，他们学到优化编译器和非优化编译器之间的差别。在不同层次上都使用同样的机器模型做工作在效率上有很大的优势，因为模型从上至下都是一致的。不过本书也讲述了 MIPS 机器，对比了 RISC 设计原理和微程序设计的 CISC 设计。
- *完整的程序示例*——许多计算机组成和汇编语言的书会受到代码片段综合征的影响。Pep/8 的内存模型、寻址方式和输入 / 输出特性使得学生能写出完整的程序，容易执行和测试，而不只是代码片段。真实的机器，特别是 RISC 机器，有复杂的函数调用协议，涉及寄存器分配、寄存器溢出和内存对齐限制之类的问题。Pep/8 是少数几种教学机之一（有可能是唯一一个），允许学生书写具有输入 / 输出的完整程序，可以使用全局变量和局部变量、全局数组和局部数组、传值调用和传引用调用、数组参数、使用转移表的开关语句、递归、使用指针的链式结构和堆。写完整程序的作业进一步实现了通过动手来学习的目标，而不是通过读代码片段来学习。
- *理论和实践的结合*——有些读者注意到了，讲述语言翻译原理的第 7 章在计算机系统书中不常见。这种现象可悲地说明了计算机科学课程体系甚至计算机科学领域中理论和实践之间的鸿沟。因为本书讲述了 HOL6 层的 C++ 语言、Asmb5 层的汇编语言和 ISA3 层的机器语言，而且都有一个目标，即理解层次之间的关系，一个更好的问题是：“为什么不能包括讲述语言翻译原理的一章呢？”本书尽可能地加入理论以支持实践。例如，介绍布尔代数作为一个公理系统，配合练习来证明定理。
- *宽度和深度*——第 1 ～ 6 章中的内容对计算机系统或汇编语言编程的书来说是很典型的，第 8 ～ 12 章对计算机组成的书来说是很典型的。在一本书中包括这么广泛的内容是很独特的，而且它还在一个完整机器的各个抽象层次上使用一个一致的机器模型。数字电路 LG1 层内容的深度也是很特别的，它使得 CPU 的组成部分不再神秘。例如，本书描述了 Pep/8 CPU 的复用器、加法器、ALU、寄存器、内存子系统和双向总线的实现。学生学习逻辑门层的实现，没有概念上的空洞，而如果只是泛泛地描述，就只能选择相信而不能完全理解。

本书回答了这个问题：“汇编语言编程和计算机组成在计算机科学课程体系中的位置是什么？”它提供了对无处不在的冯 · 诺依曼机器架构的深入理解。本书的目标是提供本领域

内所有主要知识域的综合概述，包括软件和硬件的结合，理论和实践的结合。

计算机课程体系 2001

ACM 和 IEEE 计算机学会建立了计算机科学的“课程体系 2001”指导原则。该指导原则给出了知识体的分类和具体核心。本书适用于体系结构和组成（Architecture and Organization，AR）类别，几乎包含 AR 知识体中所有的核心主题。初期报告中的 AR 核心域以及本书中覆盖这些域的章节如下所示：

AR1. 数字逻辑和电子系统，第 10、11、12 章。

AR2. 数据的机器级表示，第 3 章。

AR3. 汇编层机器组成，第 4、5、6 章。

AR4. 内存系统组成和体系结构，第 9、11 章。

AR5. 接口与通信，第 8、9 章。

AR6. 功能组成，第 11、12 章。

AR7. 多处理和其他体系结构，第 8 章。

致谢

Pep/1 有 16 条指令、一个累加器和一种寻址方式。Pep/2 增加了变址寻址。John Vannoy 用 ALGOL W 语言写了 2 个模拟器。Pep/3 有 32 条指令，用 Pascal 语言编写，是学生软件项目，由 Steve Dimse、Russ Hughes、Kazuo Ishikawa、Nancy Brunet 和 Yvonne Smith 完成。Harold Stone 在早期审阅中提出许多对 Pep/3 架构的改进意见，后来被加到 Pep/4 中，并延续到后续的机器中。Pep/4 有特殊的栈指令，模拟 ROM 和软件陷阱。Pep/5 有更正交的设计，允许任何指令使用任何寻址方式。John Rooker 写了 Pep/4 系统和早期的 Pep/5 版本。Gerry St. Romain 实现了一个 MacOS 版本和一个 MS-DOS 版本。Pep/6 简化了变址寻址方式，也包括了一组完整的条件分支指令。John Webb 用 BlackBox 开发系统编写了跟踪功能。Pep/7 把安装的内存从 4 MB 增加到了 32 MB。Pep/8 把寻址方式的数量从 4 增加到 8，安装的内存增加到 64MB。Pep/8 汇编器和模拟器的 GUI 版本由一组学生用 Qt 开发系统和 C++ 实现和维护，小组成员包括 Deacon Bradley、Jeff Cook、Nathan Counts、Stuartt Fox、Dave Grue、Justin Haight、Paul Harvey、Hermi Heimgartner、Matt Highfield、Trent Kyono、Malcolm Lipscomb、Brady Lockhart、Adrian Lomas、Ryan Okelberry、Thomas Rampelberg、Mike Spandrio、Jack Thomason、Daniel Walton、Di Wang、Peter Warford 和 Matt Wells。Ryan Okelberry 也参与编写了 Pep/8 CPU 模拟器。Luciano d'Ilori 编写了汇编器的命令行版本。

Tanenbaum 的《 Structured Computer Organization 》比其他任何一本书都更大地影响了本书的编写。本书扩展了 Tanenbaum 书的层次结构，在上面增加了高级语言层和应用层。

下面这些书稿审阅者和前一版本的用户极大地影响了本版本的终稿，他们是：Wayne P. Bailey、Jim Bilitski、Fadi Deek、William Decker、Peter Drexel、Gerald S. Eisman、Victoria Evans、David Garnick、Ephraim P. Glinert、Dave Hanscom、Michael Hennessy、Michael Johnson、Andrew Malton、Robert Martin、Richard H. Mercer、Randy Molmen、John Motil、Peter Ng、Bernard Nudel、Carolyn Oberlink、Wolfgang Pelz、James F. Peters III、James C. Pleasant、Eleanor Quinlan、Glenn A. Richard、David Rosser、Gerry St. Romain、Harold S.

Stone、J. Peter Weston 和 Norman E. Wright。Joe Piasentin 提供了艺术咨询。有两个人极大地影响了 Pep/8 的设计，一位是 Myers Foreman，有关指令集的很多想法都来自于他；另一位是 Douglas Harms，他提出很多改进意见，其中之一是 MOVSPA 指令，使得可以用传引用方式传递局部变量。

Jones and Bartlett Publishers 的责任编辑 Tim Anderson、产品主管 Amy Rose 和编辑助理 Melissa Potter 提供了宝贵的支持，很高兴与他们一起工作。Kristin Parker 设计的吸引人的封面正符合本书的风格。

我很幸运在一所致力于在本科教学中追求卓越的学校。佩珀代因（Pepperdine）大学的 Ken Perrin，提供了富有创造性的环境和专业的支持，正是在这种环境中，本书得到孕育。我的妻子 Ann 给予我无尽的支持，我要为本书占用的时间向她道歉，并送上我由衷的感谢。

Stan Warford
Malibu，California

目 录

Computer Systems, Fourth Edition