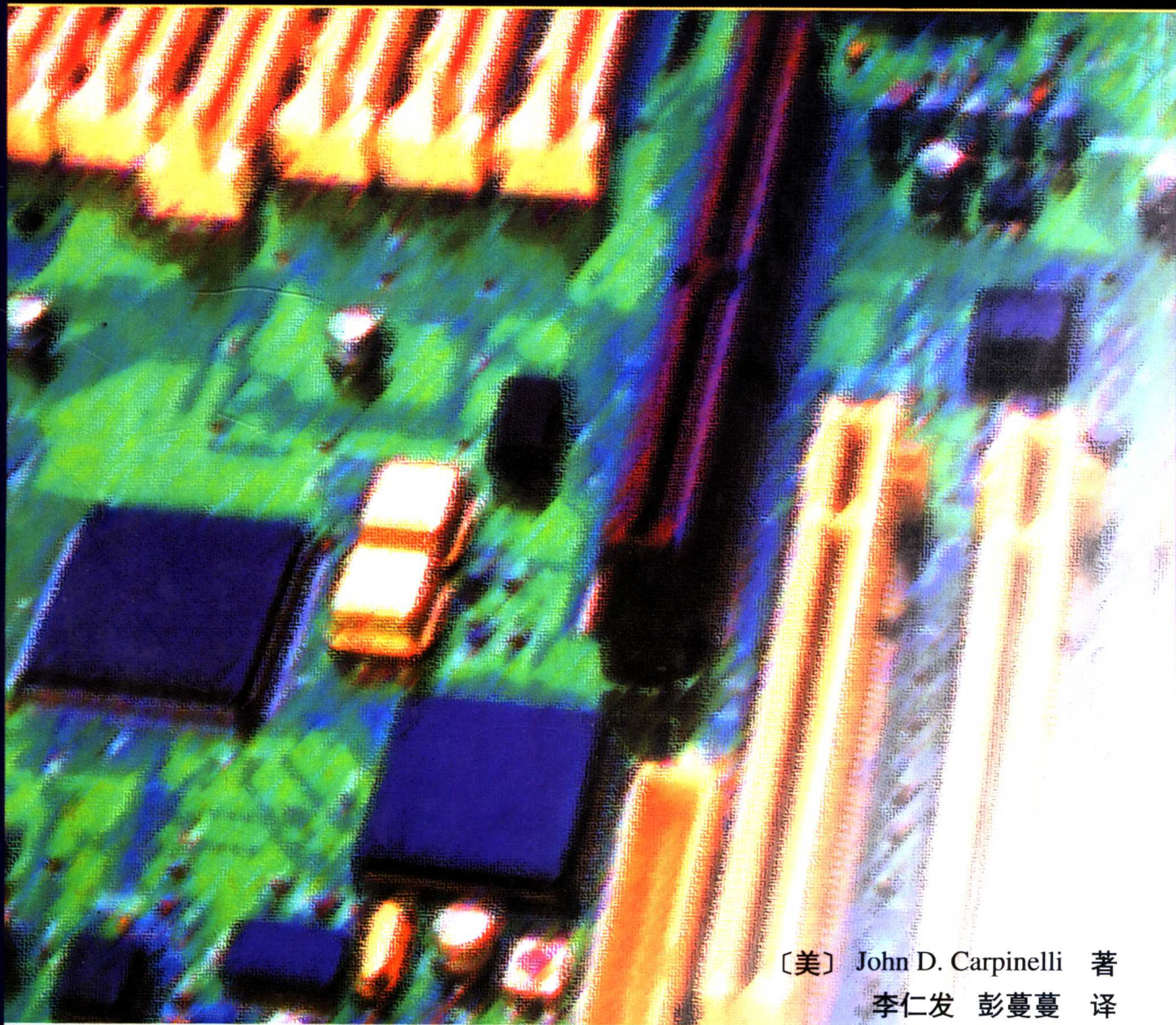


国外著名高等院校
信息科学与技术优秀教材



计算机系统组成与体系结构

Computer Systems Organization & Architecture



〔美〕 John D. Carpinelli 著
李仁发 彭蔓蔓 译

中文版

人民邮电出版社
POSTS & TELECOM PRESS

国外著名高等院校信息科学与技术优秀教材

计算机系统组成与体系结构

[美] John D. Carpinelli 著

李仁发 彭蔓蔓 译

人民邮电出版社

图书在版编目 (CIP) 数据

计算机系统组成与体系结构 / (美) 卡帕里 (Carpinelli, J. D.) 著; 李仁发, 彭蔓蔓译.
—北京: 人民邮电出版社, 2003.8

ISBN 7-115-11243-6

I. 计... II. ①卡... ②李... ③彭... III. 计算机体系结构 IV. TP303

中国版本图书馆 CIP 数据核字 (2003) 第 030023 号

版 权 声 明

Simplified Chinese Edition Copyright © 2003 by PEARSON EDUCATION NORTH ASIA LIMITED and POSTS & TELECOMMUNICATIONS PRESS.

Computer Systems Organization & Architecture by John D.Carpinelli

ISBN 0-201-61253-4

Copyright © 2001

All Rights Reserved.

Published by arrangement with Addison Wesley, Pearson Education, Inc.

This edition is authorized for sale only in People's Republic of China (Excluding the Special Administrative Region of the Hong Kong and Macao).

本书封面贴有 **Pearson Education** 出版集团激光防伪标签, 无标签者不得销售。

国外著名高等院校信息科学与技术优秀教材

计算机系统组成与体系结构

- ◆ 著 [美] John D.Carpinelli
译 李仁发 彭蔓蔓
责任编辑 陈 昇
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
读者热线 010-67132705
北京汉魂图文设计有限公司制作
北京密云春雷印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 27
字数: 652 千字 2003 年 8 月第 1 版
印数: 1-5 000 册 2003 年 8 月北京第 1 次印刷

著作权合同登记 图字: 01-2002-6014 号

ISBN 7-115-11243-6/TP·3434

定价: 45.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

内容提要

本书详述了有关计算机及其子系统设计的基本概念及相关知识。全书由三大部分组成：第一部分是数字逻辑和有限状态机，介绍了布尔代数基础、数字部件、组合逻辑和顺序逻辑、可编程逻辑器件。有限状态机是全书的基础。第二部分是计算机组成和系统结构，内容包括指令集系统结构、计算机组成、寄存器传输语言、CPU 设计、控制部件设计、算术运算、存储器结构、I/O 结构。第三部分是高级专题，内容包括 RISC 计算机和并行处理。

本书内容适度、可读性好、实用性强，适合作为计算机工程、计算机科学、电子工程、信息系统等专业的计算机体系结构课程的教材。

出版说明

2001年，教育部印发了《关于“十五”期间普通高等教育教材建设与改革的意见》。该文件明确指出，“九五”期间原国家教委在“抓好重点教材，全面提高质量”方针指导下，调动了各方面的积极性，产生了一大批具有改革特色的新教材。然而随着科学技术的飞速发展，目前高校教材建设工作仍滞后于教学改革的实践，一些教材内容陈旧，不能满足按新的专业目录修订的教学计划和课程设置的需要。为此该文件明确强调，要加强国外教材的引进工作。当前，引进的重点是信息科学与技术 and 生物科学与技术两大学科的教材。要根据专业（课程）建设的需要，通过深入调查、专家论证，引进国外优秀教材。要注意引进教材的系统配套，加强对引进教材的宣传，促进引进教材的使用和推广。

邓小平同志早在1977年就明确指出：“要引进外国教材，吸收外国教材中有益的东西。”随着我国加入WTO，信息产业的国际竞争将日趋激烈，我们必须尽快培养出大批具有国际竞争能力的高水平信息技术人才。教材是一个很关键的问题，国外的一些优秀教材不但内容新，而且还提供了很多新的研究方法和思考方式。引进国外原版教材，可以促进我国教学水平的提高，提高学生的英语水平和学习能力，保证我们培养出的学生具有国际水准。

为了贯彻中央“科教兴国”的方针，配合国内高等教育教材建设的需要，人民邮电出版社约请有关专家反复论证，与国外知名的教材出版公司合作，陆续引进一些信息科学与技术优秀教材。第一批教材针对计算机专业的主干核心课程，是国外著名高等院校所采用的教材，教材的作者都是在相关领域享有盛名的专家教授。这些教材内容新，反映了计算机科学技术的最新发展，对全面提高我国信息科学与技术的教学水平必将起到巨大的推动作用。

出版国外著名高等院校信息科学与技术优秀教材的工作将是一个长期的、坚持不懈的过程，我社网站（www.pptph.com.cn）上介绍了我们首批陆续推出的图书的详细情况，后续教材的引进和出版情况我们会及时在网上发布，敬请关注。希望广大教师和

学生将使用中的意见和建议及时反馈给我们，我们将根据您的反馈不断改进我们的工作，推出更多更好的引进版信息科学与技术教材。

人民邮电出版社

2002年4月

序 言

由 John D. Carpinelli 编著的《Computer Systems Organization & Architecture》一书由 Addison Wesley 出版社出版。本书的定位是要为计算机工程、计算机科学、电子工程、信息系统等专业的本科生提供一本内容适度、可读性好和实用性强的“计算机系统组成与体系结构”教材。

全书由三大部分组成。第一部分是数字逻辑和有限状态机，介绍了布尔代数基础、数字部件、组合逻辑和顺序逻辑、可编程逻辑器件。有限状态机是全书的基础。第二部分是计算机组成和系统结构，内容包括指令集系统结构、计算机组成、寄存器传输语言、CPU 设计、控制部件设计、算术运算、存储器结构、I/O 结构。第三部分是高级专题，内容包括 RISC 计算机和并行处理。

本书的主要特点如下：

1. 突出介绍设计方法。本书系统全面地介绍了计算机的设计方法，包括寄存器级设计，简单 CPU 设计、控制器设计、存储器设计和输入输出设计等。通过学习，读者能够掌握计算机的设计方法。

2. 实践性强。全书每章都有实践性很强的内容，例如：LED 设计、Java Applet，硬件描述语言、当今 CPU 中的映射策略，Itanium 处理器的 Cache 层次结构，i960 的 I/O 处理器，RS-422 串行标准，PowerPC750 RISC CPU 的寻址方式、RISC CPU 的寄存器窗口和寄存器命名。

3. 从历史观点介绍设计方法。全书每章都有历史回顾，便于读者了解设计方法，例如，数字电路实现和有限状态机、Intel 早期的微处理器、冯·诺依曼和哈佛结构、Intel 微处理器的存储器、协处理器。

4. 介绍了计算机实例和标准。例如，可编程逻辑器件、8085 微处理器指令集系统结构、基于 8085 的计算机、VHD-VHSIC 硬件描述语言、8085 微处理器的内部系统结构、Pentium 处理器、IEEE 754 浮点数标准、Pentium/Windows 个人计算中的存储器管理、串行通信标准、Itanium 微处理器等。

另外，Addison Wesley 公司网站上还提供了 CPU 模拟器软件和其他一些相关资源。

本书可作为计算机、电子工程等专业本科生的教材或参考书，也可作为相关专业研究生的教材和参考书。

清华大学计算机系教授

郑纬民

2001年11月18日

前 言

简介

本书是专门为计算机组成和体系结构课程编写的，适用于计算机工程、计算机科学、电子工程和信息系统等系。这门课是针对初中级水平或大学二年级学生而开设的。本书也可以用于计算机工程、计算机科学、电子工程和信息系统等专业研究生的过渡课程。

对于本书的学习不需要任何必备的基础知识。许多学生在用本书之前会修数字电路方面的课程，这虽然有用，但不是必需的，因为我为缺少这种基础的学生编写了两章有关数字电路和有限状态机的背景材料。学生也不必掌握任何一种具体的编程语言，但是应该能理解用伪代码编写的算法。

我的方法

过去，我曾用不同的教科书教过本书所包含的内容，但没有哪本教科书是完美的。有些书一开始就很难，使学生在掌握基本概念之前就陷入到不重要的细节中，而其他书则缺乏主题间的连贯性。我写本书是为了提醒大家注意这些问题并提高学生的理解力，我希望能够成功地解决出现在其他教科书中的这些问题。

在写书的过程中，我思考的东西远比本书所包括的内容要多得多，我考虑过用各种不同的方法来介绍这些内容和各种设计方法。可喜的是，在本书中所使用的方法初步证明对我的学生是很有用的。

我喜欢在课程中讲解很多设计工作，通过做设计学生不仅能够掌握知识，而且能更好地理解事情为什么会以它们已有的方式发展。认证委员会，特别是工程方面的，需要在课程中有设计工作。本书包括了一个很重要的设计环节，它从相对简单的设计开始，逐步过渡到较为复杂的设计。

学生好像更喜欢自顶向下的系统设计方法，然而，研究表明在教学上采用自底向上的方法，学生更容易理解一些。为了对这两种方法进行折衷，我首先采用自顶向下的方法描述系统，这使学生对设计过程有一个总的了解，使他们知道哪一部分会组合成

最终的设计；然后，我采用自底向上的方法来设计系统，先设计系统的各个部分，然后再把它们集成在一起形成想要的系统。我觉得自底向上的方法可以为学较早地建立已学的概念打下良好的基础。

我喜欢用简单的例子来阐述概念，这使我不用太多的其他细节就能把概念解释清楚，通常细节太多会使表述难懂。例如，本书用一个只有 4 条指令的非常简单 CPU 来讲解 CPU 的设计，在学生掌握了这个设计之后，本书才接着介绍一个较复杂的、有 16 条指令的相对简单 CPU，该 CPU 使用了非常简单 CPU 的设计技术，同时又引进了一些更先进的技术。相对简单 CPU 又用作为书中其他设计的例子。

本书把相对简单 CPU 作为一条主线，首先说明了该 CPU 的指令集结构，然后设计了一个使用该 CPU 的简单计算机，接着又用硬布线控制和微代码控制两种方法介绍了该 CPU 的设计，最后还用该 CPU 阐述了其他的主题，例如，在 CPU 内如何处理中断。这种连贯性把学生必须学习的有关材料数量减少到最小，可以使他们重点掌握所讲解的概念。

对于 CPU 设计的问题，尤其是 CPU 的控制单元的设计，我采用了有限状态机的方法。通过对读取、译码和执行指令中必要的操作进行顺序设计，我相信这会让学生更好的理解 CPU 是如何处理指令的，这也提供了一种把设计分成两部分的好方法：说明操作并实现操作。

最后，我认为当学生能把概念与现实世界的系统联系起来的时候，他们就能更好的理解所学到的知识，因此，在每一章中，我都提供了实例，既有以往的又有现在的，以满足学生的需要。

涉及的范围

本书涉及的范围较广，从基本的数字电路到并行处理。首先让我们来看看每一章的主题，然后我们再来描述贯穿全书的几条主线。这本书分为三个部分，每一部分描述如下：

数字逻辑和有限状态机

第一部分是数字逻辑和有限状态机，包括前两章。第 1 章是数字逻辑基础，它用组合逻辑和时序逻辑介绍了布尔代数和数字元件的基本知识，还介绍了贯穿本书的设计中要用到的一些更为复杂的元件，另外还包括一个有关可编程逻辑设备的实例。学生可以在重现该书的所有设计中使用这些材料。

第 2 章介绍有限状态机，这一章的内容通常不应出现在计算机组成和体系结构教材中，但是在本书中我觉得有必要这样做。微处理机的控制单元就是一个有限状态机，学习有限状态机的基础知识使学生更容易理解微处理机是怎样工作的，以及它们为什么按这种方式工作。在第 6 章和第 7 章中，有限状态机被广泛用于 CPU 的设计中。

计算机组成和体系结构

本书的第二部分是计算机组成和体系结构，它包含了八章。第 3 章讲的是微处理器的指令集结构(ISA)，当开始设计微处理器的时候，首先要明确它必须实现的任务，然后再设计指令、内部寄存器和完成这些任务所需的其他 ISA 元件，ISA 是计算机设计过程中的第一步。

第 4 章讲的是计算机组成，它考查了计算机中处理器是怎样和存储器及 I/O 设备相连的，还讨论了物理存储器的构造。本章内容提供了一个坚实的基础，本书的剩余部分都是建立在这个基础之上的。这一章还介绍了两种简单的计算机结构，第一种基于相对简单 CPU，它在本章中起着指导性的辅助作用；另一种是一个实际的设计，基于 Intel 8085 微处理器。

第 5 章讨论了寄存器传送语言 RTL, RTL 贯穿全书, 用来设计微处理机和计算机系统的其他部件。本章提供了 RTL 的基本语法和一些典型的设计, 这为学生学习本书包含的系统设计、甚至本书以外的系统设计提供所需知识。这一章还介绍了 VHDL, 它是一种在数字电路设计中广泛使用的硬件描述语言。

接下来的五章内容详细地讨论了计算机系统中不同部件的设计工作。第 6 章介绍了用硬连线控制的 CPU 设计, 首先从头介绍了一种非常简单 CPU 的设计, 这是为了既阐明标准 CPU 设计中的一些概念, 又避免让学生陷入太多的细节里面。然后提供了一个稍微复杂但仍相对简单的 CPU 设计。相对简单 CPU 的设计用到了很多非常简单 CPU 的设计技术, 同时也引进了更高级的技术。

CPU 的控制单元有两种设计方法, 一种是第 6 章介绍的硬连线控制, 另一种是第 7 章讨论的微序列控制单元的设计, 当今这种控制方式被用于很多的高级微处理器中。第 7 章首先介绍了非常简单 CPU 的微序列控制器的设计, 在该章的前一部分还介绍了微序列控制器的基础知识, 但没有涉及到太多无关的细节。然后又介绍了相对简单 CPU 的微序列控制器的设计, 这种微序列控制器还是建立在非常简单 CPU 的基础之上, 同时又引进了更为先进的设计技术。第 7 章还讨论了一个微代码 CPU (Pentium 微处理器) 的内部结构。

第 8 章讲解了计算机的运算方法。它介绍了不同的数值格式以及对这些格式的数据进行算术操作的算法, 还描述了实现这些算法的硬件, 同时还讨论了实现算术操作的专用硬件和用于所有现代处理器中的 IEEE 754 浮点标准。

第 9 章讲述了计算机系统存储器组成, 介绍了存储器的层次结构, 并详细讨论了 Cache 和虚拟内存。为了阐明它们在典型计算机系统配置, 该章介绍了运行在 Windows NT 环境下带有奔腾微处理器的计算机的存储器层次结构。

第 10 章讲述了输入/输出组织, 它介绍了 I/O 的基本功能和改进 I/O 数据传输的一些方法。虽然这一章也涉及到中断, 但是有关中断的部分内容在保持教材内容连贯性的前提下可以较早地介绍。

高级专题

第三部分包含最后两章, 介绍的是高级专题。第 11 章讲述了精简指令集计算, 或者叫 RISC 处理, 它介绍了 RISC 的基本原理和主要特征。和第 10 章的中断一样, 第 11 章中有关指令流水线的内容也可以在保持本书内容连贯性的基础上较早地介绍。这一章还介绍了 Intel Itanium 的微处理器。

最后是第 12 章, 即并行处理简介, 该章讨论了基本结构和多处理器系统的拓扑结构。它还介绍了多处理器之间的通信和存储器组成, 并且提供了通用功能的并行算法。

特点

为了使学生更容易理解, 本书在编写上有几大特点, 部分特征如下:

- 实践视角。这些视角帮助学生理解系统为什么设计成它们现有的样子, 并让学生明白这些概念是怎样应用到实际系统中去的。这些视角的例子包括为什么将 LED 显示器设计成低电平有效、在 Itanium 微处理器中 Cache 存储器是怎样组织的等等。
- 历史视角。这些视角描述了从过去到现在出现过的元件、系统或事件, 它们可以帮助学生理解计算机设计的发展史和一些重要的事件。这些视角的例子包括 Intel 早期

微处理器的发展史和奔腾微处理器如何得名等。

- 实例。每一章都以一个真实元件、系统或通用标准来结尾，这使学生能够更好的理解书中所介绍的概念，这些小节解释了几种微处理器的内部结构，范围从 25 年前的 8085 到最新的 Itanium。同时还介绍了每一章谈论的主题是怎样在实际系统中实现的，比如 Pentium/Windows 计算机中的 cache 和虚拟存储器管理。最后，这些部分还讨论了计算机设计中使用的标准，如 IEEE 754 浮点标准和通用串行总线标准。

全书的主线

教师使用这本书的主线取决于班上学生的知识背景。如果学生没有数字逻辑方面的基础知识，他们应该从本书的开始学起，他们需要学习第 1 章、第 2 章、第 3 章和第 4 章。然后可以学习第 5 章的全部内容或者只学习有关 RTL 的部分，接着学习第 6 章的全部或一部分。还可以学习第 7 章和第 8 章，或者跳过这两章，这不影响知识的连贯性。大多数情况下要学习第 9 章和第 10 章。时间允许的话，教师还可以讲讲第 11 章或第 12 章，或者在这些章节中有选择地学习一些专题。

如果学生已经修过数字设计方面的课程，则可以跳过第 1 章，甚至第 2 章。有些数字设计课程不包括有限状态机，或者只是粗略地介绍了一下，如果这样的话，在学习本书的时候就要学第 2 章或者让学生自学或作参考资料。从这儿起，该课程通常要包括第 3 章到第 7 章。第 8 章是可以选讲的，取决于教师的喜好。接着这门课程还要包括第 9 章、第 10 章和第 11 章，也可以选择第 12 章的全部或部分内容。

不管采用哪条主线，教师可能希望做些修改。有的教师可能希望用其他的材料来替换实例的部分内容。例如，第 5 章介绍了 VHDL，但如果只是将这本书作为使用 Verilog 语言实验的辅助教材或预备知识，那么介绍一些 Verilog 语言来代替 VHDL 会是恰当的。教师应该注意到有些实例是建立在前面例子的基础之上的，第 3 章、第 4 章和第 6 章在各自的实例中都使用了 8085 微处理器，而第 7 章介绍的奔腾微处理器用在了第 9 章讨论的实际计算机中。

双重主题适用于本书的好几个地方，尤其是第 10 章的中断和第 11 章的指令流水线。我把内容都完整的写了下来，以便教师可以不打断教材的连贯性。教师可以在第 6.4 节后介绍这些主题，也可以按本书编写的顺序来介绍它们。

补充材料

使用这本书的读者可以得到各种各样的补充材料，有些材料是任何人都可以得到的，而有些只提供给教师。下列的补充材料都能在本书的相关网站得到，网址是 <http://www.awl.com/carpinelli>。

所有人都可得到的材料

- 相对简单 CPU 的模拟器。这个模拟器让学生输入一个程序，对它进行汇编，然后对它的执行进行仿真。该模拟器模拟了 CPU 内的数据流程，使学生更好地理解 CPU 是怎样工作的。该模拟器被编写成一个 Java 小程序，可以运行在任何计算平台支持 Java 的页面浏览器上。其他模拟器也会经常上传到本书的相关网站上。
- 图片。本书的所有图片都已经转换成图像文件。教师可以把这些文件下载下来在课堂上使用。

致谢

本书只列出了一个作者，但是有很多人对本书的编写及最后的成书做出过贡献。可能贡献最大的是在过去 15 年间上过计算机组成与体系结构这门课的那些学生们，在解释书中的主题时，在问题和方法的取舍上，我的学生给了我很多反馈意见，在编写本书的时候我尽可能多地采纳了这些反馈意见，因为我相信他们的观点。我尤其要感谢在前几个学期帮我测试某些章节最初版本的那些学生。

虽然我设计了相对简单 CPU 模拟器的基本功能，但是实现这些功能的是我的四位学生：Aamish Kapadia、Ray Bobrowski、Leo Hendriks 和 Benedicto Catalan。他们不仅参与了简单代码的编写，而且还实现了模拟器的一些功能并且对此提出了建议。由于他们的努力，该模拟器有很多功能并且用户界面友好。

我在新泽西技术学院任教的同事也给了我非常大的帮助。Sol Rosenstark 写过好几本书，并从一开始就给我提供了很宝贵的建议，从而使得本书的编写也容易了很多。Tony Lambiase、Ken Sohn 和 Joe Strano 经常在一起讨论本书的内容、讲授和编写风格、以及其他一些有关的事情，我们的校外午餐为写作班子提供了很多实例。

我画了书中大部分的图片，但不是全部。很感谢 Intel 公司允许我复印它们的一些图片，感谢 Michelle Evans 安排了这些复印工作。

Addison Wesley 的出版组在这个项目一开始就很正确，我的编辑 Susan Hartman Sullivan 和助理编辑 Lisa Kalner 给予了我完成这本书的指导和信心。作为第一次编写教科书的作者，我相信我需要得到比经验丰富的作者更多的帮助，没有他们我不可能完成这本书（谢谢 iguana 和 Lisa）。也谢谢 Addison Wesley 出版社的 Pat Mahtani、Michael Hirsch、Mary Boucher、Jennifer Pelland、Regina Hagen 和 Joyce Cosentino，以及 Pre-Press 公司的 Marilyn Lloyd。

这本书的评审者对我的帮助也很大，他们的批评和建议在完善手稿、纠正错误和清晰陈述方面起了非常重要的作用。这些评审者列出如下：

- Murray R. Berkowitz (Towson University)
- Jose G. Delgado-Frias (University of Virginia)
- Ata Elahi (Southern Connecticut State University)
- Hassan Farhat (University of Nebraska)
- John C. Kelly, Jr. (North Carolina A&T State University)
- Rabi Mahapatra (Texas A&M University)
- Robert McIlhenny (California State University, Northridge)
- Paliath Narendran (University at Albany-SUNY)
- Yusuf Ozturk (San Diego State University)
- Ralph L. Place (Ball State University)
- John B. Zuckerman (Texas A&M University)

最后，我要感谢我的家人和朋友，他们在我编写这本书的过程中一直支持和鼓励我。谢谢我的妻子 Mary 给予我的爱、理解和永远的支持。同时也感谢我的女儿 Maria，在本书的写作过程中，她的到来是一件最令人愉快的事情。

目 录

第一部分 数字逻辑与有限状态机

第 1 章 数字逻辑基础	3
1.1 布尔代数	3
1.1.1 基本函数	4
1.1.2 布尔函数的使用	5
1.2 基本的组合逻辑	9
1.3 更复杂的组合元件	11
1.3.1 多路选择器	11
1.3.2 译码器	13
1.3.3 编码器	14
1.3.4 比较器	16
1.3.5 加法器和减法器	18
1.3.6 存储器	20
1.4 组合电路设计	21
1.4.1 BCD 码的 7 段译码器	22
1.4.2 数据排序器	24
1.5 基本时序元件	25
1.6 更复杂的时序元件	28
1.6.1 计数器	28
1.6.2 移位寄存器	30
1.7 实例：可编程逻辑设备	31
1.8 总结	33
1.9 习题	34
第 2 章 介绍有限状态机	37
2.1 状态图和状态表	38
2.2 Mealy 机和 Moore 机	41
2.3 设计状态图	41
2.3.1 模 6 计数器	42
2.3.2 串检查器	43
2.3.3 收费站控制器	44

2.4 从状态图到实现	48
2.4.1 状态赋值	49
2.4.2 Mealy 机和 Moore 机的实现	50
2.4.3 产生次态	51
2.4.4 产生系统输出	55
2.4.5 一种可替代的设计	58
2.4.6 八状态串检查器	59
2.5 实例: 实际考虑	61
2.5.1 未使用状态	61
2.5.2 异步设计	63
2.5.3 状态机转换	66
2.6 总结	67
2.7 习题	68

第二部分 计算机组成与体系结构

第 3 章 指令集结构	75
3.1 程序设计语言的级别	76
3.1.1 语言种类	76
3.1.2 编译和汇编程序	76
3.2 汇编语言指令	79
3.2.1 指令类型	80
3.2.2 数据类型	81
3.2.3 寻址方式	82
3.2.4 指令格式	84
3.3 指令集结构设计	86
3.4 相对简单的指令集结构	87
3.5 实例: 8085 微处理器指令集结构	92
3.5.1 8085 微处理器的寄存器组	92
3.5.2 8085 微处理器指令集	93
3.5.3 一个简单的 8085 程序	97
3.5.4 分析 8085 指令集结构	98
3.6 总结	99
3.7 习题	99
第 4 章 介绍计算机组成	102
4.1 基本的计算机组成	102
4.1.1 系统总线	103
4.1.2 指令周期	104

4.2 CPU 组成	106
4.3 存储器子系统组成和接口	107
4.3.1 存储器的种类	107
4.3.2 芯片内部组成	109
4.3.3 存储器子系统配置	110
4.3.4 多字节数据组成	114
4.3.5 基本功能的拓展	115
4.4 I/O 子系统组成和接口	115
4.5 相对简单计算机	117
4.6 实例：一台基于 8085 的计算机	120
4.7 总结	123
4.8 习题	124
第 5 章 寄存器传送语言	126
5.1 微操作和寄存器传送语言	127
5.2 用 RTL 描述数字系统	132
5.2.1 数字元件	132
5.2.2 简单系统的描述与实现	133
5.3 更复杂的数字系统和 RTL	136
5.3.1 模 6 计数器	137
5.3.2 收费站控制器	139
5.4 实例：VHDL—VHSIC 硬件描述语言	143
5.4.1 VHDL 语法	144
5.4.2 高层抽象的 VHDL 设计	146
5.4.3 低层抽象的 VHDL 设计	149
5.5 总结	151
5.6 习题	151
第 6 章 CPU 设计	154
6.1 CPU 的设计规范	154
6.2 非常简单 CPU 的设计与实现	155
6.2.1 非常简单 CPU 的设计规范	155
6.2.2 从存储器中取指令	156
6.2.3 指令译码	157
6.2.4 指令执行	158
6.2.5 建立所需的数据通路	160
6.2.6 非常简单 ALU 的设计	163
6.2.7 用硬连线控制设计控制单元	164
6.2.8 设计验证	168

6.3 相对简单 CPU 的设计和实现	169
6.3.1 相对简单 CPU 的规范	169
6.3.2 取指令和指令译码	171
6.3.3 执行指令	172
6.3.4 创建数据通路	176
6.3.5 相对简单 ALU 的设计	179
6.3.6 用硬连线控制设计控制单元	181
6.3.7 设计验证	183
6.4 简单 CPU 的缺点	183
6.4.1 更多的内部寄存器和高速缓存	183
6.4.2 CPU 内部的多总线	184
6.4.3 指令流水线式处理	185
6.4.4 更大的指令集	186
6.4.5 子程序和中断	187
6.5 实例: 8085 微处理器的内部结构	187
6.6 总结	189
6.7 习题	189
第 7 章 微序列控制单元设计	194
7.1 微序列控制器设计基础	194
7.1.1 微序列控制器的操作	194
7.1.2 微指令格式	196
7.2 非常简单微序列控制器的设计和实现	197
7.2.1 基本布局	197
7.2.2 生成正确序列并设计映象逻辑	198
7.2.3 用水平微代码生成微操作	199
7.2.4 用垂直微代码生成微操作	201
7.2.5 从微代码直接产生控制信号	205
7.3 相对简单微序列控制器的设计和实现	206
7.3.1 修改状态图	206
7.3.2 设计顺序硬件和微代码	207
7.3.3 用水平微代码完成设计	210
7.4 减少微指令数	212
7.4.1 微子程序	212
7.4.2 微代码跳转	215
7.5 微程序控制和硬连线控制的比较	216
7.5.1 指令集的复杂度	217
7.5.2 修改的容易度	217
7.5.3 时钟速度	217