

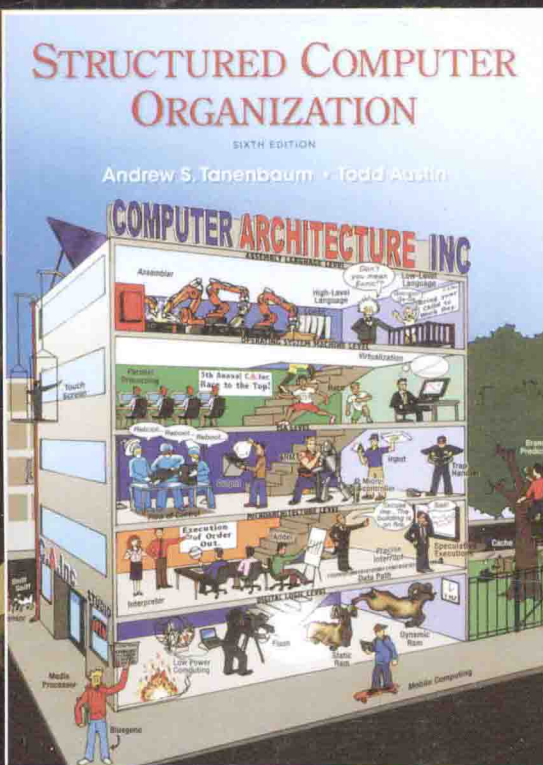
计算机组成 结构化方法

(荷) Andrew S. Tanenbaum

(美) Todd Austin

著 刘卫东 宋佳兴 译

Structured Computer Organization
Sixth Edition



机械工业出版社
China Machine Press

计 算 机 科 学 丛

原书第6版

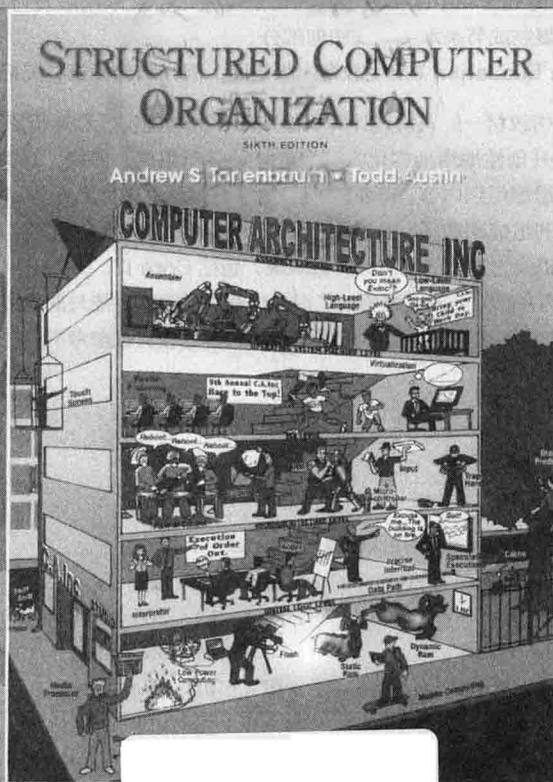
计算机组成 结构化方法

(荷) Andrew S. Tanenbaum

(美) Todd Austin

著 刘卫东 宋佳兴 译

Structured Computer Organization
Sixth Edition



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

计算机组成: 结构化方法 (原书第 6 版) / (荷) 塔嫩鲍姆 (Tanenbaum, A. S.) 等著; 刘卫东, 宋佳兴译. —北京: 机械工业出版社, 2014.7

(计算机科学丛书)

书名原文: Structured Computer Organization, Sixth Edition

ISBN 978-7-111-45380-2

I. 计… II. ①塔… ②刘… ③宋… III. 计算机组成原理 IV. TP301

中国版本图书馆 CIP 数据核字 (2014) 第 086966 号

本书版权登记号: 图字: 01-2012-6631

Authorized translation from the English language edition, entitled *Structured Computer Organization*, 6E, 9780132916523 by Andrew S. Tanenbaum, Todd Austin, published by Pearson Education, Inc., Copyright © 2013.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Chinese simplified language edition published by Pearson Education Asia Ltd., and China Machine Press Copyright © 2014.

本书中文简体字版由 Pearson Education (培生教育出版集团) 授权机械工业出版社在中华人民共和国境内 (不包括中国台湾地区和香港、澳门特别行政区) 独家出版发行。未经出版者书面许可, 不得以任何方式抄袭、复制或节录本书中的任何部分。

本书封底贴有 Pearson Education (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

传统的计算机组成与结构教材一般只讲述计算机组成部件的功能实现和运行原理, 而缺乏对各组成部件之间关系的描述。本书开创性地用结构化方法来描述计算机组成, 围绕“计算机由层次结构组成, 每层完成规定的功能”这一思想组织内容, 详细介绍了数字逻辑层、微体系结构层、指令系统层、操作系统层和汇编语言层等各层的组成和实现, 并说明了低层是如何为上一层功能的实现提供支持的。

在第 6 版中, 示例计算机换为当前的主流计算机系统, 包括 Core i7、OMAP4430 和 ATmega168, 并根据计算机组成和体系结构方面的最新进展更新了很多内容, 使得本书能与时俱进, 保持经典性和时效性。本书可以作为计算机专业本科生学习计算机组成与结构课程的教材或参考书, 也可供其他相关专业人员参考。

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: 姚蕾 肖晓慧 刘立卿

责任校对: 殷虹

印刷: 三河市宏图印务有限公司

版次: 2014 年 8 月第 1 版第 1 次印刷

开本: 185mm × 260mm 1/16

印张: 35.75

书号: ISBN 978-7-111-45380-2

定价: 99.00 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: hzsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

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，不仅筹划了研究的范畴，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

2013年春节刚过，接到机械工业出版社华章分社的电话，希望我们能继续翻译《Structured Computer Organization》第6版。合作多年，盛情难却，踌躇一番后，也只好应承下来。

第6版和第5版，从出版时间上看，相差了差不多8年，而这段时间是计算机组成和系统结构领域飞速发展的时期。一系列新的概念兴起和新应用的普及，如云计算、物联网、移动计算等，使计算机应用更为广泛地深入到社会的各个方面，甚至成为推动社会发展的重要力量。反过来，这些新概念和新应用，也对计算机组成和结构提出了新的要求，引导着计算机组成和结构发展方向。

在此背景下，《Structured Computer Organization》第6版在保持原有的基本结构和主要内容时，高度关注计算机组成和结构领域的发展趋势，特别注意将计算机组成方面一些新的技术增加进来。在实例的选择上，Core i7、OMAP4430和ATmega168分别作为桌面计算、移动计算和嵌入式计算的代表，且在指令系统上又分别采用了x86、ARM和AVR指令系统，覆盖了当前流行计算模式的主要内容，体现了作者选材的独具匠心。而GPU、FLASH、FPGA和并行技术等引入，也使教材内容与时俱进，让读者能在掌握计算机组成的基本原理的同时，领略到该领域内一些新的发展方向 and 趋势。

本书一直保持了从层次化角度描述计算机硬软件系统完整体系结构的特点。在计算机专业教育中强调系统能力培养这一观点，也逐步得到我国教育界的重视。可以说，这本书是进行计算机系统能力培养的一本好教材。

本版前言和第1~4章由刘卫东翻译，第5~8章、附录A、附录B、附录C及索引由宋佳兴翻译。清华大学计算机科学与技术系王诚教授审阅了全书。

虽然我们特别注意改正了第5版译文中我们自己发现的和一些读者指出的翻译错误，也尽我们的能力纠正了第6版（英文版）本身的一些小的纰漏，但限于译者水平，译文中肯定还会有错误和不当之处，依然敬请读者不吝赐教。

刘卫东 宋佳兴

2014年4月14日

本书的前 5 个版本都是建立在计算机由层次结构组成、每层完成规定的功能这一思想上的。今天，这一基本思想依旧和第 1 版刚出版时一样正确，它依然是第 6 版的基础。和前 5 版一样，我们将详细讨论数字逻辑层、微体系结构层、指令系统层、操作系统层和汇编语言层。

尽管保留了基本的结构，但第 6 版还是包含了或大或小的许多变动，以跟上飞速更新的计算机产业的步伐。例如，本版的示例计算机均已改成当前的主流计算机系统，即 Intel 公司的 Core i7、德州仪器的 OMAP4430 和 Atmel 的 ATmega168。其中，Core i7 是广泛应用于笔记本、台式机和服务器的 CPU，而 OMAP4430 是一款基于 ARM 的主流 CPU，被很多智能手机和平板电脑采用。

许多人可能从未听说过 ATmega168 微控制器，但也许每天都会和它打交道。由于极其低廉的成本（几美分）、高附加值的软件和外部设备，以及数量众多的程序员，基于 AVR 的 ATmega168 广泛用于从定时收音机到微波炉的许多嵌入式系统中，世界上 ATmega168 的数量要比 Pentium 以及 Core i3、i5 和 i7 CPU 多出许多个数量级。同时，ATmega168 也是 Arduino 单片嵌入式系统中使用的处理器，Arduino 是一个基于开源代码的软硬件平台，最初由意大利一所大学设计，价格低廉，比在比萨店吃顿晚饭还便宜。

近年来，许多讲授本课程的教授多次询问关于汇编语言程序设计的内容，第 6 版把这些材料放在了本书的网站中（地址见后），这样做可以方便更新这些材料，以保持它们的时新性。我们选择的是 8088 汇编语言，主要原因在于它是当前流行的 Core i7 处理器使用的 IA32 指令集的先前版本，我们当然也可以选择 ARM 或者 AVR 指令集，甚至其他没人听说过的指令系统作为例子，但作为目的性很强的教学工具，由于大多数同学家里都有一台兼容 8088 的计算机，8088 显然是一个更好的选择。Core i7 的全集太复杂，不适宜让学生了解全部细节，8088 则要简单得多。

另外，本版使用 Core i7 作为例子，讲解了不少该 CPU 的细节，Core i7 本身就能运行 8088 汇编语言程序。尽管如此，调试汇编语言代码依然比较困难，我们提供了一系列工具来帮助大家学习汇编语言编程，包括 8088 的汇编器、模拟程序和跟踪程序。这些工具可在 Windows、Unix 和 Linux 环境下运行，都可在本书的网站上下载。

这些年来，本书也变得越来越厚（第 1 版有 443 页，而本版有 769 页^①）。由于学科本身的发展和对它的了解的加深，这也是不可避免的。因此，当用作教材时，可能就无法在一门课（一个学期）中讲述完所有内容。一种可行的方法是讲述第 1、第 2 和第 3 章的全部内容，第 4 章的前 4 节，以及第 5 章的少量内容。其余的时间可根据教师和学生的兴趣介绍第 4 章剩余部分和第 6、第 7、第 8 章的部分内容。

下面逐章介绍一下各章纲要及对第 5 版的主要改动。第 1 章依然是对计算机体系结构发展的历史回顾，指出我们是如何走过来的，目前的位置和发展道路上的主要里程碑。当了解到 20 世纪 60 年代世界上最强大的计算机的成本高达数百万美元，而计算能力还不及

① 这里的页数均指原版书。——译者注

现在智能手机的 1% 时, 许多同学可能会十分惊讶。我们还要介绍广义上的计算机系列, 包括 FPGA、智能手机、平板电脑和游戏控制器。当然, 本版新的 3 种示例处理器 (Core i7、OMAP4430 和 ATmega168) 的体系结构也在本章中有简单说明。

第 2 章在处理方式方面, 增加了数据并行处理器也就是图形处理器 (Graphics Processing Unit, GPU) 的相关材料。存储技术领域引入了当前正趋于流行的基于 Flash 的存储设备。而 2.4 节中, 加入了对现代游戏控制器 (如 Wiimote、Kinect) 和智能手机、平板电脑上使用的触摸屏等的介绍。

第 3 章的许多地方都进行了修改。该章依然从晶体管开始论述, 这样做的好处是没有任何硬件基础的同学也能理解现代计算机的运行原理。新增加的内容主要是现场可编程门阵列 (Field-Programmable Gate Array, FPGA) 的相关内容, 现场可编程芯片价格降低到可以在教学中广泛使用, 使得真正的大规模门级设计可以引入课堂。对用作示例的三种新的处理器体系结构做了芯片级的介绍。

第 4 章讲解计算机是如何运行的。这章一直就颇受好评, 因此从第 5 版开始就没做大的改变。当然, 用作示例的 Core i7、OMAP4430 和 ATmega168 的微体系结构层的内容是新的。

第 5、6 章仅就新的示例体系结构所涉及的部分做了改写, 尤其是增加了对 ARM 和 AVR 指令系统的描述。第 6 章用 Windows 7 取代 Windows XP 作为例子。

第 7 章的内容是关于汇编语言编程的, 和前一版比基本没有变化。

第 8 章做了许多修改, 以反映在并行计算机领域各方面的新的研究动向。增加了 Core i7 多处理器体系结构的一些新的特征, 并详细介绍了 NVIDIA Fermi 的通用 GPU 体系结构。最后, 对 BlueGene 和 Red Storm 超级计算机的内容进行了更新, 以跟上这些巨型机最近的升级。

参考文献也做了修改。将推荐读物放在了网站上, 因此, 这一版中引出的仅仅是本书引用过的参考文献, 许多都是全新的。计算机组成是一个快速发展的领域。

附录 A 和附录 B 从上一版本开始就没有修改, 这些年以来, 二进制数和浮点数的表示没有什么变化。附录 C 是关于汇编语言程序设计的, 由阿姆斯特丹 Vrije 大学的 Evert Wattel 博士编写, Wattel 博士拥有多年使用这些工具进行教学的经验。我们十分感谢他写的这个附录, 主要内容没有做调整, 但将工具放到了网站上。

除汇编语言的工具之外, 本书网站还包含一个配合第 4 章使用的图形模拟器。它是由 Oberlin 学院的 Richard Salter 教授编制的, 可用于帮助同学们掌握第 4 章讨论的基本原理。十分感谢 Richard Salter 教授提供该软件。

包含这些工具以及其他内容的本书网站的网址是:

<http://www.pearsonhighered.com/tanenbaum>

从网页上点击本书的链接, 并从菜单项中选择你所找寻的页面。其中的学生资源包括:

- 汇编器 / 追踪器软件。
- 图形模拟器。

教师资源包括:

- 课程的 PowerPoint 幻灯片。
- 每章习题的解答。

教师资源需要通过密码访问。使用本书作为教材的教师可通过联系当地的 Pearson 教育

代表获得密码。

许多人读过本书的（部分）手稿，并提出了有益的建议或以其他方式对本书提供了帮助。我们要特别感谢 Anna Austin、Mark Austin、Livio Bertacco、Valeria Bertacco、Debapriya Chatterjee、Jason Clemons、Andrew DeOrio、Joseph Greathouse 和 Andrea Pellegrini。

下列人士审阅了手稿并提出了修改意见：Jason D. Bakos（University of South Carolina）、Bob Brown（Southern Polytechnic State University）、Andrew Chen（Minnesota State University, Moorhead）、J. Archer Harris（James Madison University）、Susan Krucke（James Madison University）、A. Yauvz Oruc（University of Maryland）、France Marsh（Jamestown Community College）和 Kris Schindler（University at Buffalo）。十分感谢他们。

还有几位帮助我们提供了新的习题。他们是：Byron A. Jeff（Clayton University）、Laura W. McFall（Depaul University）、Taghi M. Mostafavi（University of North Carolina at Charlotte）和 James Nystrom（Ferri State University）。我们再次感谢他们的帮助。

我们的编辑 Tracy Johnson 给了我们全方位的帮助，十分感谢他的耐心。Carole Snyder 在协调参与本书写作工作的各类人士方面提供了十分专业的帮助。Bob Englehardt 对本书的最后出品做了很好的贡献。

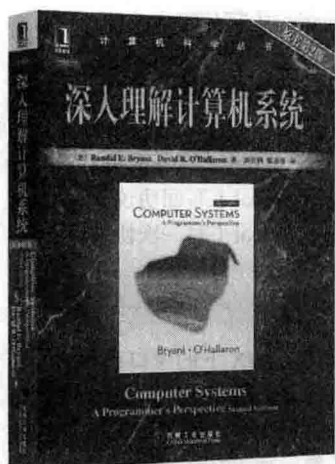
我（Andrew S. Tanenbaum）要再次感谢 Suzanne 的爱和耐心，她一直陪伴我完成了 21 本著作的写作。Barbara 和 Marvin 永远是开心果，他们现在也知道了教授是如何谋生的。Aron 属于新生的一代：上幼儿园之前就已经是资深的计算机用户了。Nathan 还没有达到这个程度，但他学会走路后，iPad 将会是他的下一个学习目标。

最后，我（Todd Austin）要趁此机会感谢我的岳母 Roberta，她为我写作本书提供了许多黄金时间。她在意大利 Bassano Del Grappa 的餐桌上刚好有足够的幽静、舒适和葡萄酒，供我完成这项重要的工作。

Andrew S. Tanenbaum

Todd Austin

推荐阅读



深入理解计算机系统（原书第2版）

作者: Randal E. Bryant 等 译者: 龚奕利 等 ISBN: 978-7-111-32133-0 定价: 99.00元



计算机系统：系统架构与操作系统的高度集成

作者: Umakishore Ramachandran 等 译者: 陈文光 等



计算机系统概论（原书第2版）

作者: Yale N. Patt 等 译者: 梁阿磊 等 ISBN: 978-7-111-21556-1 定价: 49.00元



计算机系统基础

作者: 袁春风 ISBN: 978-7-111-46477-8 定价: 49.00元

推荐阅读



算法导论（原书第3版）

作者：Thomas H. Cormen等 译者：殷建平 等 ISBN: 978-7-111-40701-0 定价：128.00元



机器学习

作者：Tom Mitchell 译者：曾华军 等 ISBN: 978-7-111-10993-7 定价：35.00元



数据挖掘：概念与技术（原书第3版）

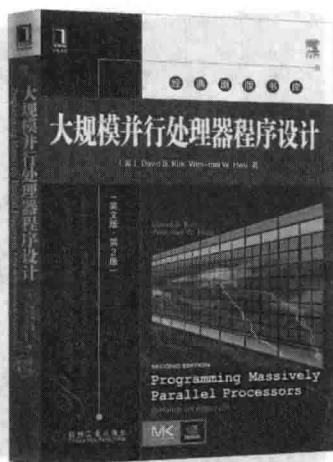
作者：Jiawei Han等 译者：范明 等 ISBN: 978-7-111-39140-1 定价：79.00元



数据挖掘：实用机器学习工具与技术（原书第3版）

作者：Ian H. Witten等 译者：李川 等 ISBN: 978-7-111-45381-9 定价：79.00元

推荐阅读



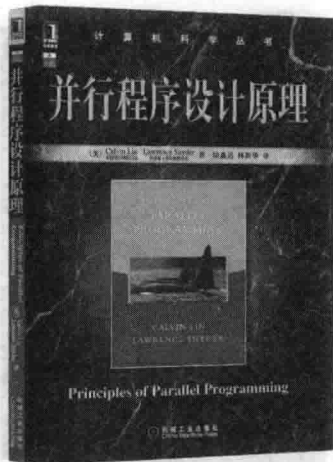
大规模并行处理器程序设计 (英文版·第2版)

作者: David B. Kirk 等 ISBN: 978-7-111-41629-6 定价: 79.00元



并程序计设计导论

作者: Peter S. Pacheco ISBN: 978-7-111-39284-2 定价: 49.00元



并程序计设计原理

作者: Larry Snyder 等 ISBN: 978-7-111-27075-1 定价: 45.00元



多处理器编程的艺术 (修订版)

作者: Maurice Herlihy 等 ISBN: 978-7-111-41858-0 定价: 69.00元

目 录

Structured Computer Organization, Sixth Edition

出版者的话

译者序

前言

第 1 章 概述	1
1.1 结构化计算机组成	1
1.1.1 语言、层次和虚拟机	1
1.1.2 现代多层次计算机	3
1.1.3 多层次计算机的演化	5
1.2 计算机体系结构的里程碑	8
1.2.1 第零代——机械计算机 (1642—1945)	8
1.2.2 第一代——电子管计算机 (1945—1955)	10
1.2.3 第二代——晶体管计算机 (1955—1965)	12
1.2.4 第三代——集成电路计算机 (1965—1980)	14
1.2.5 第四代——超大规模集成电路 计算机 (1980 年至今)	15
1.2.6 第五代——低功耗和无所 不在的计算机	17
1.3 计算机家族	18
1.3.1 技术和经济推动	18
1.3.2 计算机扫视	20
1.3.3 一次性计算机	21
1.3.4 微型控制器	22
1.3.5 移动计算机和游戏计算机	23
1.3.6 个人计算机	24
1.3.7 服务器	25
1.3.8 大型主机	26
1.4 系列计算机举例	26
1.4.1 x86 体系结构简介	27
1.4.2 ARM 体系结构简介	31
1.4.3 AVR 体系结构简介	32
1.5 公制计量单位	33

1.6 本书概览	34
习题	35
第 2 章 计算机系统组成	38
2.1 处理器	38
2.1.1 CPU 组成	39
2.1.2 指令执行	40
2.1.3 RISC 和 CISC	42
2.1.4 现代计算机设计原则	43
2.1.5 指令级并行	44
2.1.6 处理器级并行	47
2.2 主存储器	50
2.2.1 存储位	50
2.2.2 内存编址	51
2.2.3 字节顺序	52
2.2.4 纠错码	53
2.2.5 高速缓存	56
2.2.6 内存封装及其类型	58
2.3 辅助存储器	59
2.3.1 层次存储结构	59
2.3.2 磁盘	60
2.3.3 IDE 盘	62
2.3.4 SCSI 盘	63
2.3.5 RAID 盘	64
2.3.6 固态硬盘	67
2.3.7 只读光盘	68
2.3.8 可刻光盘	71
2.3.9 可擦写光盘	73
2.3.10 DVD	73
2.3.11 Blu-Ray	74
2.4 输入/输出设备	75
2.4.1 总线	75
2.4.2 终端	78
2.4.3 鼠标	81
2.4.4 游戏控制器	83
2.4.5 打印机	84
2.4.6 电信设备	88

2.4.7 数码相机	94	3.8 小结	169
2.4.8 字符编码	95	习题	169
2.5 小结	99	第4章 微体系结构层	173
习题	99	4.1 微体系结构举例	173
第3章 数字逻辑层	103	4.1.1 数据通路	174
3.1 门和布尔代数	103	4.1.2 微指令	178
3.1.1 门	103	4.1.3 微指令控制: Mic-1	180
3.1.2 布尔代数	105	4.2 指令系统举例: IJVM	183
3.1.3 布尔函数的实现	107	4.2.1 栈	183
3.1.4 等价电路	108	4.2.2 IJVM 内存模型	185
3.2 基本数字逻辑电路	110	4.2.3 IJVM 指令集	186
3.2.1 集成电路	111	4.2.4 将 Java 编译为 IJVM	189
3.2.2 组合逻辑电路	111	4.3 实现举例	190
3.2.3 算术电路	114	4.3.1 微指令和符号	190
3.2.4 时钟	118	4.3.2 用 Mic-1 实现 IJVM	193
3.3 内存	119	4.4 微体系结构层设计	201
3.3.1 锁存器	119	4.4.1 速度与价格	202
3.3.2 触发器	121	4.4.2 缩短指令执行路径长度	203
3.3.3 寄存器	122	4.4.3 带预取的设计: Mic-2	208
3.3.4 内存组成	123	4.4.4 流水线设计: Mic-3	211
3.3.5 内存芯片	125	4.4.5 七段流水线设计: Mic-4	215
3.3.6 RAM 和 ROM	128	4.5 提高性能	217
3.4 CPU 芯片和总线	130	4.5.1 高速缓存	218
3.4.1 CPU 芯片	130	4.5.2 分支预测	222
3.4.2 计算机总线	132	4.5.3 乱序执行和寄存器重命名	226
3.4.3 总线宽度	134	4.5.4 推测执行	230
3.4.4 总线时钟	135	4.6 微体系结构层举例	232
3.4.5 总线仲裁	138	4.6.1 Core i7 CPU 的微体系结构	232
3.4.6 总线操作	141	4.6.2 OMAP4430 CPU 的微体系结构	236
3.5 CPU 芯片举例	143	4.6.3 ATmega168 微控制器的微体系结构	240
3.5.1 Intel Core i7	143	4.7 Core i7、OMAP4430 和 ATmega168 三种 CPU 的比较	241
3.5.2 德州仪器的 OMAP4430 片上系统	147	4.8 小结	242
3.5.3 Atmel 的 ATmega168 微控制器	150	习题	243
3.6 总线举例	152	第5章 指令系统层	246
3.6.1 PCI 总线	152	5.1 指令系统层概述	247
3.6.2 PCI Express	159	5.1.1 指令系统层的性质	247
3.6.3 通用串行总线 USB	162	5.1.2 存储模式	249
3.7 接口电路	165	5.1.3 寄存器	250
3.7.1 输入/输出接口	165		
3.7.2 地址译码	166		

5.1.4	指令	251	5.5.6	循环控制指令	282
5.1.5	Core i7 指令系统层概述	251	5.5.7	输入 / 输出指令	283
5.1.6	OMAP4430 ARM 指令系统层概述	253	5.5.8	Core i7 指令系统	285
5.1.7	ATmega168 AVR 指令系统层概述	255	5.5.9	OMAP4430 ARM CPU 指令系统	287
5.2	数据类型	256	5.5.10	ATmega168 AVR 指令系统	289
5.2.1	数值数据类型	257	5.5.11	指令集比较	291
5.2.2	非数值数据类型	257	5.6	控制流	291
5.2.3	Core i7 的数据类型	258	5.6.1	顺序控制流和转移	291
5.2.4	OMAP4430 ARM CPU 的数据类型	258	5.6.2	过程	292
5.2.5	ATmega168 AVR CPU 的数据类型	259	5.6.3	协同过程	295
5.3	指令格式	259	5.6.4	陷阱	297
5.3.1	指令格式设计准则	260	5.6.5	中断	297
5.3.2	扩展操作码	261	5.7	详细举例：汉诺塔	300
5.3.3	Core i7 指令格式	263	5.7.1	Core i7 汇编语言实现的汉诺塔	300
5.3.4	OMAP4430 ARM CPU 指令格式	264	5.7.2	OMAP4430 ARM 汇编语言实现的汉诺塔	302
5.3.5	ATmega168 AVR 指令格式	266	5.8	IA-64 体系结构和 Itanium 2	303
5.4	寻址	267	5.8.1	IA-32 的问题	303
5.4.1	寻址方式	267	5.8.2	IA-64 模型：显式并行指令计算	304
5.4.2	立即寻址	267	5.8.3	减少内存访问	305
5.4.3	直接寻址	267	5.8.4	指令调度	305
5.4.4	寄存器寻址	267	5.8.5	减少条件转移：判定	307
5.4.5	寄存器间接寻址	267	5.8.6	推测加载	308
5.4.6	变址寻址	269	5.9	小结	309
5.4.7	基址变址寻址	270	习题	310	
5.4.8	栈寻址	270	第 6 章	操作系统层	314
5.4.9	转移指令的寻址方式	272	6.1	虚拟内存	314
5.4.10	操作码和寻址方式的关系	273	6.1.1	内存分页	315
5.4.11	Core i7 的寻址方式	274	6.1.2	内存分页的实现	316
5.4.12	OMAP4430 ARM CPU 的寻址方式	276	6.1.3	请求调页和工作集模型	319
5.4.13	ATmega168 AVR 的寻址方式	276	6.1.4	页置换策略	320
5.4.14	寻址方式讨论	276	6.1.5	页大小和碎片	321
5.5	指令类型	277	6.1.6	分段	322
5.5.1	数据移动指令	277	6.1.7	分段的实现	324
5.5.2	双操作数指令	278	6.1.8	Core i7 的虚拟内存	326
5.5.3	单操作数指令	279	6.1.9	OMAP4430 ARM CPU 的虚拟内存	329
5.5.4	比较和条件转移指令	280	6.1.10	虚拟内存和高速缓存	331
5.5.5	过程调用指令	281	6.2	硬件虚拟化	331
			6.3	操作系统层 I/O 指令	333

6.3.1 文件	333	7.4.4 动态链接	390
6.3.2 操作系统层 I/O 指令的实现	335	7.5 小结	393
6.3.3 目录管理指令	337	习题	393
6.4 用于并行处理的操作系统层指令	338	第 8 章 并行计算机体系结构	396
6.4.1 进程创建	339	8.1 片内并行	397
6.4.2 竞争条件	339	8.1.1 指令级并行	397
6.4.3 使用信号量的进程同步	342	8.1.2 片内多线程	402
6.5 操作系统实例	345	8.1.3 单片多处理器	406
6.5.1 简介	345	8.2 协处理器	410
6.5.2 虚拟内存实例	350	8.2.1 网络处理器	411
6.5.3 操作系统层 I/O 举例	352	8.2.2 图形处理器	416
6.5.4 进程管理实例	361	8.2.3 加密处理器	418
6.6 小结	365	8.3 共享内存的多处理器	418
习题	366	8.3.1 多处理器与多计算机	418
第 7 章 汇编语言层	371	8.3.2 内存语义	424
7.1 汇编语言简介	371	8.3.3 UMA 对称多处理器体系结构	426
7.1.1 什么是汇编语言	372	8.3.4 NUMA 多处理器系统	432
7.1.2 为什么使用汇编语言	372	8.3.5 COMA 多处理器系统	439
7.1.3 汇编语言语句的格式	373	8.4 消息传递的多计算机	440
7.1.4 伪指令	374	8.4.1 互连网络	441
7.2 宏	376	8.4.2 MPP——大规模并行处理器	443
7.2.1 宏定义、调用和扩展	376	8.4.3 集群计算	450
7.2.2 带参数的宏	377	8.4.4 多计算机的通信软件	454
7.2.3 高级特性	378	8.4.5 调度	456
7.2.4 汇编器中宏处理的实现	378	8.4.6 应用层的共享内存	457
7.3 汇编过程	379	8.4.7 性能	461
7.3.1 两趟汇编的汇编器	379	8.5 网格计算	465
7.3.2 第一趟扫描	379	8.6 小结	467
7.3.3 第二趟扫描	382	习题	468
7.3.4 符号表	384	参考文献	471
7.4 链接和加载	385	附录 A 二进制数	479
7.4.1 链接器的处理过程	386	附录 B 浮点数	487
7.4.2 目标模块的结构	388	附录 C 汇编语言程序设计	493
7.4.3 绑定时间和动态重定位	389	索引	534

概 述

数字计算机是通过执行人们给出的指令来完成工作的机器。描述如何完成一个确定任务的指令序列称为**程序 (program)**。每台计算机的电路都只能识别和直接执行有限的简单指令，所有程序都必须在执行前转换成这些指令。这些基本的指令几乎都不会比下面的指令复杂：

两个数相加。

检查某数是否为零。

将一些数据从计算机内存的某些单元复制到另外的单元中。

计算机的这些原始指令共同组成了一种可供人和计算机进行交流的语言，我们称其为**机器语言 (machine language)**。设计一种新的计算机时，人们必须首先决定它的机器语言中包含哪些指令。通常，原始指令应尽量简单，兼顾考虑计算机的使用要求和性能要求，以降低实现电路的成本和复杂度。正因为大多数机器语言如此简单，使用起来才显得十分困难和乏味。

通过对计算机的这些简单描述，我们可将计算机结构化为一系列抽象机，每台抽象机都建立在其下层抽象机的基础上。这样，计算机的复杂性就在可控范围内，计算机系统的设计也可在有组织和系统的状态下进行。我们把这种方法称为**结构化计算机组成 (structured computer organization)**，并以此命名本书。下一节我们将解释它的含义，然后回顾一下计算机发展历史和这当中一些有影响的机型。

1

1.1 结构化计算机组成

正如前面提到的，在方便人们使用和方便计算机实现之间存在着巨大的差距。人可能要做 X，而计算机只会做 Y。这就有问题了。本书的目的就是解释如何解决这个问题。

1.1.1 语言、层次和虚拟机

这个问题可从两个途径解决，两者都需要设计一个比内置的机器指令更方便人们使用的新的指令集合。这样，新的指令集合也构成了一种语言，我们称为 L1，对应地把机器中内置的机器语言指令组成的语言叫 L0。两种途径的不同之处在于采取什么办法让只能执行用 L0 写的程序的计算机执行用 L1 写的程序。

一种途径是在执行用 L1 写的程序之前生成一个等价的 L0 指令序列来替换它，生成的程序全部由 L0 指令组成。计算机执行等效的 L0 程序来代替原来的 L1 程序，这种技术叫做**翻译 (translation)**。

另一种途径是用 L0 写一个程序，将 L1 的程序作为输入数据，按顺序检查它的每条指令，然后直接执行等效的 L0 指令序列计算出结果。它不需要事先生成一个 L0 语言的新程序。我们把这种方法称为**解释 (interpretation)**，把完成这个过程的 L0 程序称为**解释器 (interpreter)**。

翻译和解释其实是类似的。两种方法中 L1 的指令最终都是通过执行等效的 L0 指令序列来实现的。区别在于，翻译时整个 L1 程序都先转换为 L0 程序，然后 L1 程序就被抛弃，新

的 L0 程序被装入计算机内存中执行。执行过程中，运行的都是新生成的 L0 程序，控制计算机的也是 L0 程序。

而解释时，每条 L1 指令被检查和解码之后将立即执行，不生成翻译后的程序。这里，控制计算机的是解释器。对它来说，L1 程序仅仅只是数据。这两种方法，以及后来它们的综合，都得到了广泛的应用。

比起理解翻译和解释这两个概念，想象存在一种假想的以 L1 为机器语言的计算机或虚拟机也许更简单一些。让我们把这种虚拟机定义为 M1（相应地，把原来的以 L0 为机器语言的虚拟机定义为 M0），如果这种计算机可以以足够低廉的成本得到，那就根本不需要 L0 这种语言或者是执行 L0 语言程序的机器了。人们可以简单地用 L1 写程序并让计算机直接执行。即使因为使用 L1 为语言的虚拟机太贵或太复杂而不能由电子电路构成，大家还是可以写 L1 语言的程序。这些程序可以用能直接被现有计算机执行的 L0 语言程序翻译或解释。换句话说，大家完全可以像虚拟机真正存在一样用它们的语言写程序。

为使翻译或解释现实可行，两种语言 L0 和 L1 的差别不能“太”大。这条限制经常意味着，虽然 L1 比 L0 好一些，但对于多数应用来说还不理想。这也许会导致对提出 L1 的最初目的——减轻程序员不得不用一种更适合计算机的语言来描述算法的负担——有些失望。然而，不应该是绝望。

显然，解决问题的办法是发明一种比 L1 更面向人且少面向机器的指令集来取代它，这个指令集形成的语言，我们可以称为 L2（对应的虚拟机称为 M2）。人们可以像用 L2 作为机器语言的虚拟机真正存在一样用 L2 写程序，然后翻译成 L1 或用 L1 写成的解释器来执行。

这种发明一系列语言，每种都比前一种更方便人们使用的步骤可以无限地继续下去，直到最后找到一种合适的语言。每种语言都以前一种为基础，我们可以把使用这种技术的计算机看成一系列层（layer 或者 level），如图 1-1 所示，一层在另一层之上。最底部的语言或层最简单，而最上面的语言或层最复杂。

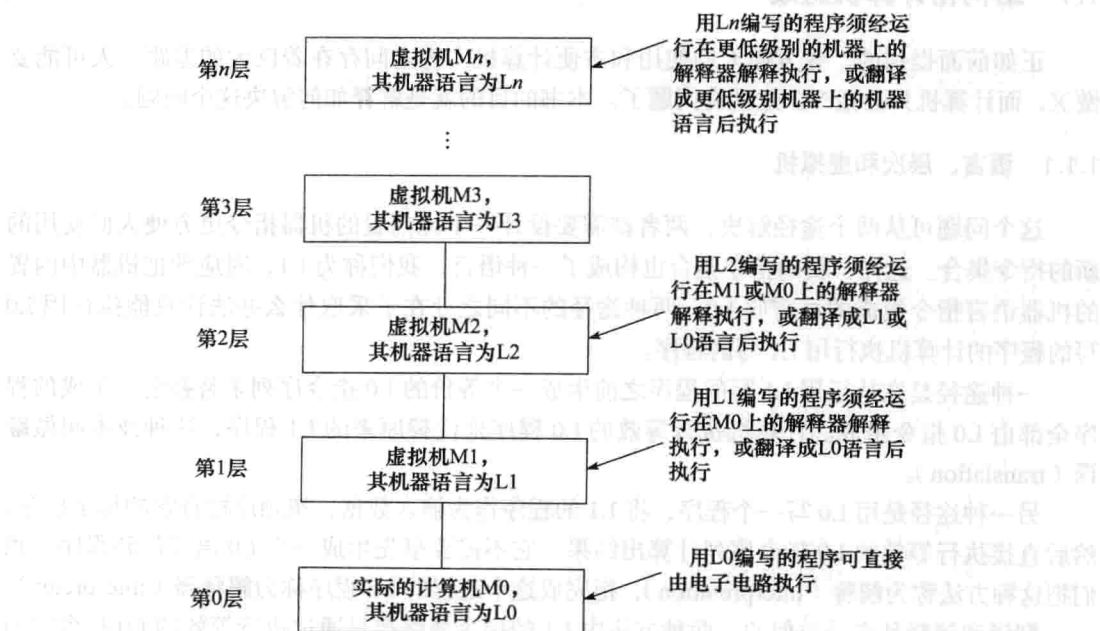


图 1-1 多层计算机