



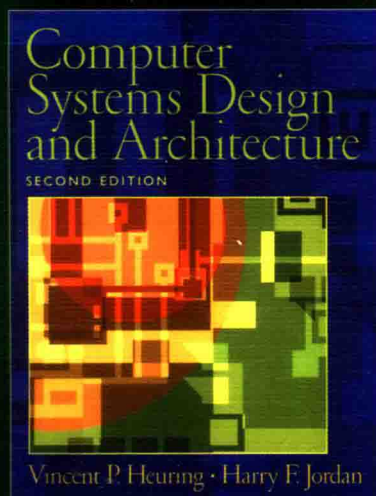
国外计算机科 学教材系列

权威作者 经典教材

计算机系统设计 与 结构

(第二版)

Computer Systems Design and Architecture
Second Edition



[美] Vincent P. Heuring 著
Harry F. Jordan

邹恒明 保蕾蕾 译

PEARSON
Prentice
Hall



电子工业出版社
Publishing House of Electronics Industry
<http://www.phei.com.cn>

国外计算机科学教材系列

计算机系统设计与结构

(第二版)

Computer Systems Design and Architecture
Second Edition

[美] Vincent P. Heuring 著
Harry F. Jordan

邹恒明 保蕾蕾 译

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本书从系统结构设计师、汇编程序员和逻辑设计师的角度介绍了计算机系统结构的设计。全书从计算机系统结构设计的综述入手,讲解了机器和机器语言之间的关系,引入了有代表性且容易理解的 SRC 模型和 RTN 结构功能描述语言,并讨论了相关的逻辑设计问题;接下来作者用实例说明了 CISC 和 RISC 的区别,深入剖析了指令集和硬件之间的接口关系,介绍了 CPU 流水线、多指令发射计算机、微代码控制单元的设计以及算术逻辑处理单元的设计;之后作者详细介绍了存储器的层次化结构设计,并且讨论了机器输入输出系统和外围设备;最后作者讨论了一些计算机网络互连方面的论题。

本书可作为高校计算机、电子等相关专业本科生和研究生微机原理、系统结构和计算机设计等方面课程的教材,对相关专业人士和研发人员也很有裨益。

Simplified Chinese edition Copyright © 2005 by Pearson Education, Inc and Publishing House of Electronics Industry.
Computer Systems Design and Architecture, Second Edition, ISBN: 0130484407 by Vincent P. Heuring and Harry F. Jordan. All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Prentice Hall.

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

本书中文简体字翻译版由电子工业出版社和 Pearson Education 培生教育出版亚洲有限公司合作出版。未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

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

版权贸易合同登记号 图字:01-2004-0955

图书在版编目(CIP)数据

计算机系统设计与结构(第二版)/(美)休林(Heuring, V. P.)等著;邹恒明等译.

北京:电子工业出版社,2005.3

(国外计算机科学教材系列)

书名原文:Computer Systems Design and Architecture, Second Edition

ISBN 7-121-00639-1

I.计... II.①休... ②邹... III.①电子计算机-系统设计-教材 ②计算机体系结构-教材 IV.TP30

中国版本图书馆CIP数据核字(2005)第010755号

责任编辑:贺瑞君

印刷:北京顺义兴华印刷厂

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编:100036

经销:各地新华书店

开本:787×1092 1/16 印张:29 字数:818千字

印次:2005年3月第1次印刷

定价:42.00元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换;若书店售缺,请与本社发行部联系。联系电话:(010)68279077。质量投诉请发邮件至 zls@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

出版说明

21世纪初的5至10年是我国国民经济和社会发展的关键时期,也是信息产业快速发展的关键时期。在我国加入WTO后的今天,培养一支适应国际化竞争的一流IT人才队伍是我国高等教育的重要任务之一。信息科学和技术方面人才的优劣与多寡,是我国面对国际竞争时成败的关键因素。

当前,正值我国高等教育特别是信息科学领域的教育调整、变革的重大时期,为使我国教育体制与国际化接轨,有条件的高等院校正在为某些信息学科和技术课程使用国外优秀教材和优秀原版教材,以使我国在计算机教学上尽快赶上国际先进水平。

电子工业出版社秉承多年来引进国外优秀图书的经验,翻译出版了“国外计算机科学教材系列”丛书,这套教材覆盖学科范围广、领域宽、层次多,既有本科专业课程教材,也有研究生课程教材,以适应不同院系、不同专业、不同层次的师生对教材的需求,广大师生可自由选择和自由组合使用。这些教材涉及的学科方向包括网络与通信、操作系统、计算机组织与结构、算法与数据结构、数据库与信息处理、编程语言、图形图像与多媒体、软件工程等。同时,我们也适当引进了一些优秀英文原版教材,本着翻译版本和英文原版并重的原则,对重点图书既提供英文原版又提供相应的翻译版本。

在图书选题上,我们大都选择国外著名出版公司出版的高校教材,如Pearson Education培生教育出版集团、麦格劳-希尔教育出版集团、麻省理工学院出版社、剑桥大学出版社等。撰写教材的许多作者都是蜚声世界的教授、学者,如道格拉斯·科默(Douglas E. Comer)、威廉·斯托林斯(William Stallings)、哈维·戴特尔(Harvey M. Deitel)、尤利斯·布莱克(Uyless Black)等。

为确保教材的选题质量和翻译质量,我们约请了清华大学、北京大学、北京航空航天大学、复旦大学、上海交通大学、南京大学、浙江大学、哈尔滨工业大学、华中科技大学、西安交通大学、国防科学技术大学、解放军理工大学等著名高校的教授和骨干教师参与了本系列教材的选题、翻译和审校工作。他们中既有讲授同类教材的骨干教师、博士,也有积累了几十年教学经验的老教授和博士生导师。

在该系列教材的选题、翻译和编辑加工过程中,为提高教材质量,我们做了大量细致的工作,包括对所选教材进行全面论证;选择编辑时力求达到专业对口;对排版、印制质量进行严格把关。对于英文教材中出现的错误,我们通过与作者联络和网上下载勘误表等方式,逐一进行了修订。

此外,我们还将与国外著名出版公司合作,提供一些教材的教学支持资料,希望能为授课老师提供帮助。今后,我们将继续加强与各高校教师的密切联系,为广大师生引进更多的国外优秀教材和参考书,为我国计算机科学教学体系与国际教学体系的接轨做出努力。

电子工业出版社

教材出版委员会

- | | | |
|----|-----|---|
| 主任 | 杨芙清 | 北京大学教授
中国科学院院士
北京大学信息与工程学部主任
北京大学软件工程研究所所长 |
| 委员 | 王 珊 | 中国人民大学信息学院院长、教授 |
| | 胡道元 | 清华大学计算机科学与技术系教授
国际信息处理联合会通信系统中国代表 |
| | 钟玉琢 | 清华大学计算机科学与技术系教授
中国计算机学会多媒体专业委员会主任 |
| | 谢希仁 | 中国人民解放军理工大学教授
全军网络技术研究中心主任、博士生导师 |
| | 尤晋元 | 上海交通大学计算机科学与工程系教授
上海分布计算技术中心主任 |
| | 施伯乐 | 上海国际数据库研究中心主任、复旦大学教授
中国计算机学会常务理事、上海市计算机学会理事长 |
| | 邹 鹏 | 国防科学技术大学计算机学院教授、博士生导师
教育部计算机基础课程教学指导委员会副主任委员 |
| | 张昆藏 | 青岛大学信息工程学院教授 |

前 言

给教师的话

本书可作为大学三年级、四年级或研究生计算机设计课程的入门教材。在使用本书前,学生应该初步掌握一些高级程序设计语言,如C语言或Pascal语言,并学习过至少一学期的计算机逻辑设计。

计算机自诞生以来,已经从一个简单的独立单元演变为与高速局域网,甚至宽带网相连接的复杂系统。与这种迅猛发展同步,本书的内容也发生了巨大变化。早期的微处理器只有很原始的内部结构,而当今的微处理器已经含有多个流水线功能单元,并能提供对多处理器与多存储器的支持。有些以前很少谈到的话题如今已成为本教程讨论的主要内容。现在,入门的编译设计课程会讨论流水线处理器的优化问题,终端用户会考虑是否需要为PC机加入额外的Cache,技术服务人员也经常不得不向他们的用户讲解如何设置子网掩码。

计算机体系结构设计的中心课题是流水线处理器设计、内存结构和网络支持。为此,本书将用三章的篇幅来阐述这三个方面。为了更清晰地剖析计算机的工作原理,我们从三个角度来探讨计算机的体系结构:一是汇编与机器语言程序设计员的角度,二是数字逻辑设计师的角度,三是计算机系统架构师的角度。

在逻辑门层面,我们将通过剖析一个模型计算机的结构来阐述从机器指令集设计到中央处理器设计的全过程。在模型计算机的选择上,本书没有选择一个功能强大的商用计算机,而是选择了简化RISC计算机(SRC, Simple RISC Computer)。SRC是一个有32位地址、结构较为简单,但又有足够多功能的计算机模型。

为了精确,同时避免产生歧义,本书使用了形式描述语言来描述计算机的结构和功能。我们选择了ISP语言的一个变体,即寄存器转移标记(RTN, Register Transfer Notation)语言作为我们的形式语言。RTN简单易学,并且非常适合描述机器的行为与功能。

给学生的话

计算机诞生至今虽然只有50多年,但它在人类生活中产生的影响却是空前的。到底什么是计算机?计算机的工作原理是什么?它内部的构造是怎样的?它与外界又是如何沟通的?这些是本书将要回答的问题。一旦掌握了这些知识,计算机将变得不再神秘。学习计算机最好的办法是亲自动手设计一台计算机。因此,贯穿全书,我们将以一个计算机设计者的视角来讲述计算机的工作原理。

计算机也许是人类建造过的最复杂的系统。而任何复杂的系统都具有多面性。观察的角度不同,对计算机的认识也就会不同。这就像一栋大楼,我们既可以从它的整体建筑结构来探讨建筑设计对大楼功能的影响,也可以从楼的外形和尺寸来研究这些参数与大楼内部制热制冷系统的关系。同理,对计算机来说,我们既可以从它的整体结构和功能方面来进行讲述,也可以从一个汇编语言程序设计员的角度来对它进行研究,还可以从抽象逻辑结构或逻辑门的层次对计算机进行剖析。

所有这些不同角度的研究都是很重要的。本书将对从这三个方面进行的观察分别进行论述。我们认为,只有把握了这三个方面的认识,并在此基础上进行综合分析与研究,对计算机的理解才能有足够的广度和深度。无论一个学生将来的专业方向是计算机设计、计算机科学还是程序设计,这种广而深的理解都是十分重要的。

在学习本书前,学生应该对计算机有一些基本了解,例如使用过计算机,并且已经掌握了一些像C, Pascal或者FORTRAN之类的计算机程序设计语言。同时,学生对数字逻辑电路也应当有一定的了解。

第二版序

由于计算机技术发展过于迅速,有关计算机体系结构方面的书籍常常是墨迹未干便已过时。这种现象对计算机书籍的作者提出了很高的要求。作者必须不断地对自己的书进行修改、订正和再版,才能跟上时代发展的潮流。从一个方面看,这似乎是一个负担;但从另一个方面来说,这也是一个机会。因为我们能够不断将新的材料、新的工具或对旧议题的新思维引入修订和再版的书中。本书就是这样一个修改、订正与再版的产物。

本书将尽最大可能对计算机的工作原理进行详尽的剖析,揭开它神秘的面纱。但为了使本书篇幅合理,我们删节了一些辅助性质的内容。我们认为讲授根本的原理比介绍表面的情况更重要。学生只有准确把握了这些重要的计算机工作原理,才有可能成为未来的计算机设计者。

第二版的新特色:

工具方面,该版本引入了基于Java的汇编器和带有图形界面的仿真器,它们都可以用于SRC, MC68000和ARC(一个SPARC的子集)。这些仿真器可以在PC, Macintosh OSX,以及所有可以运行Java虚拟机的UNIX和Linux平台上运行。SRC机的VHDL和LogicWork模型也可以运行它们。

该版本每章都跟据当前的信息进行了更新,增加了许多新的例子和习题,还增加了一些经典例子。

本书的第1章是概论,主要是从三个不同的角度对计算机进行介绍。我们讲解了汇编与机器语言程序设计员所看到的计算机是什么模样,数字逻辑设计师所看到的计算机又是什么样子,以及从计算机系统设计师的角度观察计算机,又能得出什么结论。

第2章讲述计算机与机器语言的关系。着重讨论机器指令集对CPU结构设计的影响和指令集与用来描述机器结构与功能的形式语言之间的关系。描述机器的形式语言是我们研究计算机的工具,它能精确描述计算机硬件、机器指令及机器指令在硬件上的运行方式。形式描述语言有很多种,本书引入的形式语言叫寄存器转移标记(RTN, Register Transfer Notation)语言。RTN是一种简单且正好满足本教程要求的形式语言。我们通过对SRC模型机的描述来引入RTN。SRC是一个具有32位指令集结构(ISA, Instruction Set Architecture)、通用且与现今市场上的商用32位机类似的简化模型。本章结尾则对计算机逻辑电路进行了介绍,着重讲述如何将RTN描述的操作变成实实在在的计算机逻辑电路设计。

第3章讲述RISC与CISC两种流行的计算机体系结构。我们通过对代表RISC结构的SPARC处理器与代表CISC设计思想的摩托罗拉MC68000商用处理器的比较,阐述RISC与CISC结构的异同及其各自的优缺点,探讨这两种不同的设计对未来系统兼容性与机器设计方案的影响。

第4章是本书的重点。本章的目的是揭开计算机的神秘面纱。本章从逻辑门的层面对计算机硬件与机器指令之间的界面进行讲解。用RTN形式描述语言作为表述计算机操作的工具,我们首先设计一台只有一条总线的SRC机。对这台简单计算机的设计将从ISA寄存器开始,然后随着讲解

机器指令的读取、解码和执行过程,不断将新的寄存器加入到 RTN 的形式描述里。然后我们将注意力集中到“计算机的灵魂”——中央处理器和“计算机的心跳”——系统时钟与计时系统的设计上来。在剖析完只有一条总线的 SRC 机后,我们将讲述有两条和三条总线的计算机的设计,着重讨论硬件结构设计对机器指令执行速度的影响、硬件复位和异常处理。学习完本章之后,学生对计算机中央处理器的门级工作原理应该有相当清楚的理解。

第 5 章讲述 CPU 的流水线结构、多指令发射计算机和微编码控制单元设计。几乎所有当代的计算机处理器均采用流水线结构。不管是编译系统设计员、机器语言程序设计员,还是计算机系统架构师,对流水线结构的清楚理解是这些专业人员必须具备的基本素质。我们首先介绍流水线结构的重要性,然后通过对 SRC 机的流水线结构进行分析来阐述流水线结构的设计过程。随后我们集中讲述机器指令的并行执行,并着重讨论两种并行情况。第一种并行情况是超标量运算,即一个 CPU 里含有多个能并行工作的功能单元。第二种情况是超长指令字 (VLIW, Very Long Instruction Word),即一条机器指令包含有多个可并行的处理步骤。作为例子,我们在本章介绍如何在 SRC 机上实现 VLIW 并行功能。本章的结尾讲述微编码设计技术。微编码在通用微处理器中用得不多,它更多地是在要求交货快或要求可自定义用途的专用计算机设计中发挥重要作用。其实,微编码作为一个有趣的计算机设计思路,颇值得我们花时间来了解。

第 6 章讲述计算机的算术逻辑单元 (ALU, Arithmetic and Logic Unit)。ALU 的设计对系统整体性能的影响巨大。而 ALU 的性能则主要是由在硬件上实现的算法及数据类型决定的。因此,我们的讨论也就从数据类型延伸到算法、再延伸到机器硬件。数据类型主要讨论整数和浮点数。算法则讨论顺序语句和逻辑语句在 ALU 中的执行,其中着重讲述如何使用 ALU 来执行条件及分支语句。本章结尾讲述浮点算术。

第 7 章讲述计算机的内存结构设计。从最简单的一位字长的 RAM 和 ROM 单元的设计为起点,我们讲述怎样将这些一位字长的小单元组成芯片,又怎样将芯片组成板卡,再如何将板卡组成模块。本章着重剖析计算机内存结构的基本性质、Cache 的设计及 Cache 与内存的相互关系。本章的内容还包括虚拟存储器。虚拟存储器主要研究在内存空间不够的情况下,如何将内存空间从内存扩展到磁盘上。

第 8 章对计算机的输入输出结构进行详细的讨论。本章从几种不同的总线构造讲起,仔细讲解总线信号和总线计时系统。然后集中阐述两种主要的通用 I/O 接口:串行与并行接口。在这之后,本章讨论机器中断与 I/O 系统的关系,及这两者与直接内存访问 (DMA, Direct Memory Access) 的关系。

第 9 章讲述 I/O 系统的另一端:外部设备。本章讨论的内容包括磁盘结构与功能、视频及其他显示设备、打印机、鼠标以及计算机与模拟设备的连接。本章将重点介绍这些外部设备的工作原理、它们与 CPU 之间的接口以及它们与外部世界的界面。在此基础上,再着重对外部设备的性能进行详细的分析。

第 10 章讲述计算机网络。从网络的结构和拓扑开始,我们讲述三种当代机器通信的方式。第一种方式是 RS-232 串行数据通信协议。这个通信协议用于两台计算机的点到点通信或一台计算机与一台终端的通信。在第二种方式的讨论里,我们对计算机局域网进行研究,着重从物理层与数据层来探讨以太网通信技术。讨论将包括以太网数据分组结构、高速以太网、USB 和 FireWire (火线,又叫 1394 接口)。第三种通信方式是 Internet。Internet 可能是世界上最大、也是最重要的计算机通信系统了。我们讨论的主题将集中在 TCP/IP 协议和 Internet 地址及其编址寻址上。本章也将对当前流行的 A、B 和 C 三种 Internet 编址系统进行讨论,着重论述这三种方式对 IP 地址空间的浪费,以

及如何使用 CIDR、DHCP 和 NAT 方法来减少地址浪费。本章的结尾将简略讨论一下 Internet 的应用和未来。

教学支持材料

本书备有一份习题解答手册，其中含有本书几乎所有习题的答案。本书的电子讲义则含有约 600 幅幻灯片。这些幻灯片有两种格式：一种是 Acrobat 格式，另一种是 PowerPoint 格式。这些幻灯片涵盖了本书的主要观点和全部图表。本教材的软件支持工具包括基于 Java 的 SRC、MC68000 和 SPARC 的汇编器与模拟器。这些软件工具能在 PC、Macintosh OSX 平台，以及 UNIX 和 Linux 系统上运行。电子讲义与软件支持工具均可以从 <ftp://schof.colorado.edu/pub/CSDA> 网站上下载，其他材料的提供方式请参见书后的教学支持说明。

目 录

第1章 通用计算机	1
1.1 通用计算机	1
1.2 用户眼中的计算机	2
1.3 机器语言与汇编语言程序员眼中的计算机	3
1.4 计算机架构师眼中的计算机	10
1.5 逻辑设计师眼中的计算机	14
1.6 历史回顾	16
1.7 研究现状与发展趋势	19
1.8 本书的讲解方式	20
小结	20
参考文献	21
习题	22
第2章 机器, 机器语言和数字逻辑	24
2.1 计算机及机器指令的分类	24
2.2 计算机指令集	26
2.3 简化 RISC 计算机的非形式化描述	39
2.4 使用寄存器转移标记语言对 SRC 机进行形式描述	45
2.5 使用 RTN 语言对寻址模式进行描述	53
2.6 寄存器转移与逻辑电路: 从行为到硬件	55
小结	64
参考文献	65
习题	66
第3章 几种真实机器	69
3.1 计算机功能与性能	69
3.2 精简指令集计算机与复杂指令集计算机的比较	73
3.3 CISC 处理器: 摩托罗拉 MC68000	76
3.4 一种 RISC 计算机体系结构: SPARC 机	97
小结	110
参考文献	111
习题	111
第4章 处理器设计	114
4.1 处理器设计过程	114
4.2 1 总线 SRC 计算机的微观结构	115

4.3 数据通道实现	120
4.4 1 总线 SRC 机的逻辑设计	121
4.5 计算机控制单元	132
4.6 2 总线和 3 总线处理器设计	139
4.7 机器复位	144
4.8 机器异常	146
小结	153
参考文献	154
习题	155
第 5 章 处理器设计高级议题	157
5.1 流水线结构	157
5.2 流水线冲突	176
5.3 指令级并行	185
5.4 微编码	194
小结	206
参考文献	207
习题	207
第 6 章 计算机算法与算术单元	211
6.1 数字系统与基数转换	211
6.2 定点算术	220
6.3 算术单元 ALU 设计的半数值情形	238
6.4 浮点算术	244
小结	251
参考文献	251
习题	252
第 7 章 内存系统设计	256
7.1 导言: 内存系统的组成单元	256
7.2 RAM 结构: 逻辑设计师的视角	260
7.3 内存电路板和模块	275
7.4 双层内存架构	287
7.5 Cache	292
7.6 虚拟内存	300
7.7 计算机的存储子系统	309
小结	311
参考文献	311
习题	312
第 8 章 输入和输出	314
8.1 I/O 子系统	314

8.2 可编程 I/O	317
8.3 I/O 中断	326
8.4 直接内存访问	333
8.5 I/O 数据格式转换和错误控制	336
小结	342
参考文献	342
习题	343
第 9 章 外部设备	345
9.1 磁盘驱动器	346
9.2 改善磁盘系统的性能和可靠性	354
9.3 其他海量存储设备	355
9.4 视频显示设备	356
9.5 打印机	363
9.6 输入设备	365
9.7 与模拟世界之间的接口	365
小结	369
参考文献	370
习题	370
第 10 章 通信, 网络和 Internet	372
10.1 计算机与计算机之间的数据通信	372
10.2 串行数据通信协议	380
10.3 局域网	386
10.4 现代串行总线: USB 和火线	389
10.5 Internet	391
小结	401
参考文献	401
习题	402
附录 A SRC 机的 RTN 描述	404
附录 B 汇编与汇编器	412
附录 C 部分习题及解答	419
附录 D 简单 RISC 计算机	431
附录 E SRC 汇编语言约定	433
索引	434

第1章 通用计算机

本书的目的是剖析计算机的结构和工作原理。我们将从数字逻辑层面、机器语言设计层面和系统结构层面对计算机的结构进行透彻讲解,并对各层面之间的相互联系进行阐述,以使学生在对多个层面进行综合分析的基础上把握计算机设计与结构的精髓。

本书第1章主要介绍观察计算机的三个视角或层面。我们分别讲解从逻辑设计师的视角、机器语言程序设计者的视角和从系统结构设计者的视角所看到的计算机的异同。每一个视角对应计算机结构的一个不同层次。在每一个层次上,我们将讲解在这个层次上所考虑的计算机组件、这个层次所提供的功能及承担的责任,以及工作在该层的人员所能使用的工具。为了增加对计算机发展过程的理解,同时纪念在计算机结构与设计上做出卓越贡献的先驱,本章的末尾对计算机发展的历史做了一个简短的回顾,还介绍了计算机技术领域的现状和未来趋势。

1.1 通用计算机

计算机是数字计算机的简称。数字计算机通常也叫通用计算机。计算机科学理论对通用计算机的定义是:如果提供足够的内存容量,通用机与图灵机是等价的(图灵机是根据20世纪40年代英国计算机科学家Alan Turing的名字命名的)。泛泛地讲,图灵机能够在给以足够时间的情况下计算出所有可以计算的函数。这里的关键是足够的内存与足够的时间。在过去的50多年里,计算机的内存已从仅仅1024位(bit)增加到了约 10^{12} 位;计算机的运算速度也从每毫秒执行一条机器指令增长到每毫秒执行100 000条机器指令。即便如此,今天的计算机并不能解决所有的计算问题。例如,实时三维仿真图像处理、天气预报和复杂物理系统的模拟就是当代计算机也很难计算的问题。也许有一天,计算机会发展到能够在合理的时间内解决所有人类感兴趣的问题。但这一天离我们似乎还很远。

当今的计算机种类繁多,对不同的应用领域,我们通常会选择不同种类的计算机。目前计算机的种类主要有:用来控制小型家电和玩具的电子控制器、在家里和办公室使用的台式电脑及各种用于大型复杂计算与处理的大型机和超级计算机。

1.1.1 定义与约定

在对计算机进行描述时,我们往往不得不使用一些极大和极小的数字。掌握这些数字的用法及其所代表的意义是学习这本书的基础。计算机上经常用到的大数主要有千(K, kilo),兆(M, mega),吉(G, giga)和太(T, tera)。在商业与工程领域,我们都知道1 K等于 10^3 ,1 M等于 10^6 ;1 G等于 10^9 ;1 T则等于 10^{12} 。因为我们使用的是十进制,这些数字均以 10^3 递进。但计算机使用二进制,直接用十进制的数来描述计算机有时不是很方便。我们便将K, M, G, T这些计量单位转换成了与其十进制值最接近的二进制值。表1.1列出了这些计量单位在十进制与二进制里所代表的数值。区分这些计量标准何时代表十进制、何时代表二进制应该是一件很容易的事。例如,二进制通常用来描述计算机内存容量,而十进制则用来描述计算机系统时钟频率。本书用到的小数则有:毫(m, milli, 10^{-3});微(μ , micro, 10^{-6});纳(n, nano, 10^{-9});和皮(p, pico, 10^{-12})。这些小数在计算机领域里通常被用来描述时钟周期和系统延迟。这些数值表示法和单位在全书中频繁出现,读者应该掌握它们。

表 1.1 10 的幂和 2 的幂的表示法

术语	十进制表示	以2为底表示
K (kilo)	10^3	$2^{10} = 1024$
M (mega)	10^6	$2^{20} = 1\,048\,576$
G (giga)	10^9	$2^{30} = 1\,073\,741\,824$
T (tera)	10^{12}	$2^{40} = 1\,099\,511\,627\,776$

在计算机术语里, 一个字节 (byte) 代表 8 个位。一个字 (word) 则代表一组被一起处理的位。一个字可以是 8 位、16 位、32 位或任何其他计算机能同时处理的位数。在本书里, 字的定义须根据当时的上下文而定。为描述方便, 我们常常对一个字里的位进行标记。我们既可以从左到右标, 也可以从右到左标。但不管哪种标记方式, 我们称最左边的位为最高有效位 (MSB, Most Significant bit), 称最右边的位为最低有效位 (LSB, Least Significant Bit)。

在计量单位里, 秒缩写为 s, 位缩写为 b, 字缩写为 w, 纳秒缩写为 ns, 兆字节缩写为 MB, 微秒则缩写为 μs 。例如, 每秒吉位缩写为 Gb/s。秒的倒数称为赫兹, 缩写为 Hz。100 ns 的时钟周期相当于 10 MHz 的时钟频率。

在有可能造成误解的地方, 我们将在二进制数后加一个 B, 如 0110B, 在十六进制前面加 0x 来区分, 如 0xA2B。

经典例子: 托尔斯泰的小说《战争与和平》有 696 页。一张 CD-ROM 可以存储大约 600 MB 的数据。一个字符要用一个字节来存储。一张 CD-ROM 能容纳下这本小说的内容吗? 请说明如何计算。

答案: 一张 CD-ROM 能容纳下这本小说的内容。这本书需要 $5\,000 \times 696 = 3\,480\,000$ 字节或 3.32 MB 的存储容量。(注意 1 000 000 和 1 Mega 的区别)

1.1.2 计算机面面观

与所有复杂系统一样, 对计算机的研究也可以从多个方面来进行。后面将定义并讲解观察计算机的四个视角: 即终端用户视角、机器语言程序员视角、计算机系统架构师的视角和计算机逻辑设计师的视角。在这里, 用户是计算机的最终使用者, 机器语言程序员关心的是计算机在机器层面上的行为, 系统架构师关心的是整个计算机系统的设计和性能, 逻辑设计师则将注意力集中于计算机的数字逻辑层面。

当然, 我们还可以定义更多的观察层面或视角。但对于研究计算机的工作原理来说, 上面的四种观察视角已经足够了。在当今的计算机研发队伍里, 可能有人善于集成电路设计, 有人善于使用逻辑综合分析工具, 有人对应用层面的性能分析在行, 有人精于成本分析, 有人则负责工程制造, 还有人负责外观。在设计的一个个环节, 任何一个人都有可能从一个或多个视角来看待计算机。

1.2 用户眼中的计算机

这里的用户指的是利用计算机来进行有用工作的人, 他可能在使用文本处理器, 也可能在运行 Excel, 还有可能在进行高级语言编程。不管是哪种情况, 用户是看不到计算机内部结构的。用户看到的只不过是操作系统和应用软件提供的界面。而程序设计人员看到的计算机则是由操作系统和程序设计环境提供的界面。用户对计算机的关心仅限于程序运行的速度、内存容量和外部设备的性能。或者说, 用户对计算机的关心只限于性能方面, 参见图 1.1。

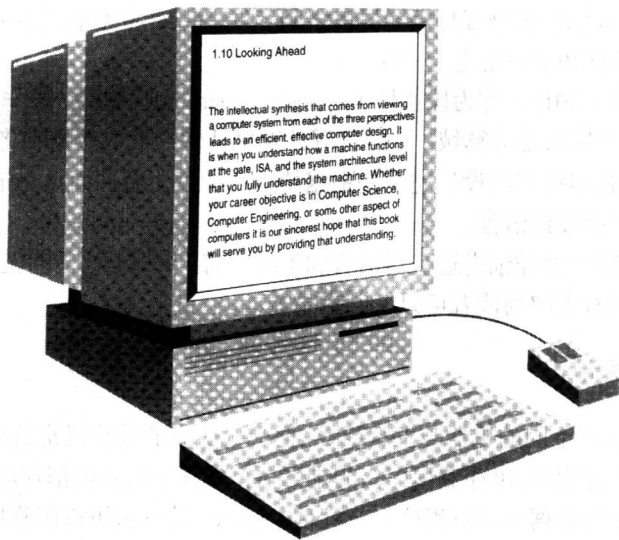


图 1.1 用户眼中的计算机

1.3 机器语言与汇编语言程序员眼中的计算机

机器语言与汇编语言程序员关心的是与机器（计算机）打交道的软件。为了能够对这个表述进行讲解，我们首先给出一些基本定义。

- **机器指令：**以 1 和 0 两个数值组成的可以被直接执行的序列^①。
- **机器语言：**所有机器指令的总和，也称为机器的自然语言。
- **汇编语言：**机器语言的对应助记符集。使用字母与数字符号便于程序员记忆与使用。
- **汇编器：**一个计算机程序。这个程序的功能是将汇编语言编写的程序翻译^②成机器语言程序。

汇编语言与机器语言之间存在一一对应关系。从汇编语言到机器语言的翻译是一个一一对应的机械转换过程。因此，可以说汇编语言程序员实际上是在使用机器的自然语言来进行程序设计的。

表 1.2 列出了几条摩托罗拉 MC68000 汇编语言编写的语句和它们所对应的机器语言指令。我们将这些指令分解成了不同的字段，每一字段描述指令的一部分。

表 1.2 两条 MC68000 指令

MC68000汇编语言	机器语言
MOVE.W D4, D5	0011 101 000 000 100
ADDI.W #9, D2	00000110 01 000 010 0000 0000 0000 1001

第 1 条指令 MOVE.W D4, D5 指示计算机将第 4 个寄存器（100）中的 16 位字的内容复制（0011）到第 5 个寄存器里（101，寄存器是能存储一组位的一种存储单元）。

表中的第 2 条指令 ADDI.W 指示计算机将一个 16 位长的整数(00000110)加到第 2 个寄存器存储的数值上，并将结果保存在第 2 个寄存器里。在这条指令里，这个 16 位长的整数是数值 9(0000 0000 0000 1001)。

① 此条为译者所加。——译者注
② 这种翻译是一对多的映射关系。

从表中可以看出,虽然汇编语言与机器语言之间存在一一对应关系,但一条指令的对应组成部分在该指令中所处的位置并不一定完全一样(参见第3章)。

指令的第一个字段(0011)称为操作域(opcode, operation code)。操作域告诉计算机要进行的操作是什么。后面4个字段是运算域。它们告诉计算机从何处读取数据和将结果存在何处。这里要注意:每一种汇编和机器语言的设计完全是根据特定的计算机特性而量身定制的,因此汇编和机器语言不是跨平台的程序设计语言。

我们将在第2章和第3章详细讲述机器语言和汇编语言。附录B含有对汇编器和汇编过程的详细介绍,对汇编过程没有了解的读者可以参阅附录B。

1.3.1 可存储程序

不同类的计算机有着不同的机器与汇编语言,但当代所有的计算机都具有一个共同的特点,那就是它们都只运行可存储程序。可存储程序指的是可执行的程序与数据都存放在计算机的内存或磁盘上,计算机可以像处理数据一样对程序进行处理。例如:计算机可以将程序从磁盘上调入内存,或从内存存回到磁盘上。

可存储程序这个概念已经深入到今天的计算机设计和应用中,没有人会觉得这有什么了不起的。但在50年前,计算机刚刚发明的时候,提起程序设计,人们会想到插线板、打孔纸带或其他的机械存储方法,装载程序则是通过接入插线板(plug board)或用机械的办法读入打孔纸带。可存储程序这个概念是由20世纪40年代中期的Eckert和Mauchley等提出来的。这个概念一经提出就获得了普遍支持。目前,可存储程序这个概念已成为几乎所有通用计算机的设计基础。

存储在计算机中的程序是一条一条地执行的。计算机从内存读取一条指令,把它临时存储在指令寄存器中,中央处理器执行指令寄存器里的指令,然后计算机读取下一条指令,重复上述过程。这个过程称为读取-执行-读取循环。计算机的基本操作就是这样一个读取-执行-读取-执行的大循环。这个循环一直延续到程序的结束或机器运行被终止(当代的计算机则有可能同时执行多条指令,我们将在第3章和第5章对其进行讲解)。

要想让读取-执行-读取-执行的大循环成为可能,则计算机必须知道从哪里读取下一条指令。而这个信息存储在一个叫程序计数器的寄存器里。这个寄存器有时又被叫做指令指针。一般情况下,本书将使用为更多人接受的程序计数器这个术语,除非在讨论某一特定计算机时,该计算机的制造商使用指令指针这个术语。

图1.2描述了指令的读取过程。该图也给出了内存与寄存器的大小。例如:左上方通用CPU寄存器为32位,程序计数器也是32位,而内存单元大小则为16位。由于MC68000可以按字节读取内存,而且一次可以读取1、2或者4个字节,这使得计算机的地址计算变得复杂了。在本图显示的例子中,机器指令为16位长,占两个字节。由于程序计数器中的地址以字节为单位,所以每次读取指令后,程序计数器必须加2才能正确地指向下一条指令的位置。

图中所示程序计数器的内容为4000,也就是下一条将被执行的指令所在的内存地址为4000。计算机的控制单元首先将存在地址4000中的指令读取出来并存放到计算机的指令寄存器里,然后进行解码和执行。从图中可以看出,被读取的指令就是我们在表1.2列出的MOVE.W指令。在读取指令的时候,计算机控制单元同时将程序计数器加2,使其指向地址4002。当计算机执行完上一条指令后,将读取地址为4002的指令,然后重复上述过程。

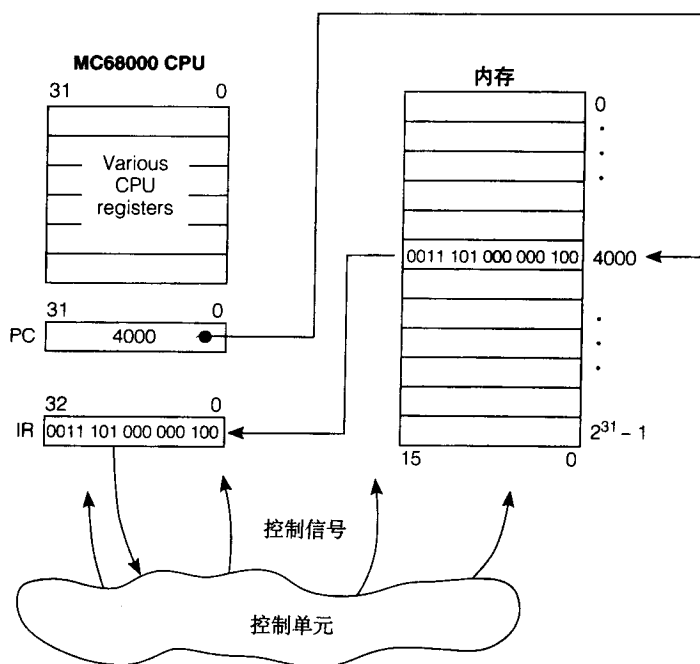


图 1.2 指令的读取 - 执行过程

1.3.2 程序设计员的模型：指令集结构

计算机能执行的所有指令的集合称为指令集。从程序员的视角来看,计算机是由指令集和这些机器指令所控制的资源组成的。这个指令集加上可被其控制的资源,有时被称为指令集结构 (ISA, Instruction Set Architecture)。ISA 包括指令集、内存和所有可编程的寄存器。但 ISA 不包括那些用于临时存储功能的寄存器。

微处理器的三代划分

微处理器从问世以来已经历了几代。图 1.3 从程序员的视角对三代微处理器进行了比较。图中这三代微处理器分别是比较老的 20 世纪 70 年代的摩托罗拉 M6800、20 世纪 70 年代后期的英特尔 8086 及其 2000 年推出的后继者 Pentium 4 型微处理器。这里同时进行对比的还有 DEC 在 20 世纪 80 年代开发的 VAX11 和 1999 年问世的 PowerPC G4, G4 是由 IBM、摩托罗拉和苹果公司组成的工业集团共同开发的杰作。图中寄存器右边的符号是其名字和编号,上面显示了位的排列。奔腾机的寄存器画在方框里。如果寄存器或内存的中间画有一条线,则说明该机可以对不足一个字的寄存器或内存空间进行操作。请注意 PPC601 (PowerPC601) 的位排列顺序与别的机器刚好相反。这是因为该处理器在设计时将最高有效位编为 0,而别的机器则将最低有效位定为 0。从图中我们可以看到,伴随着技术进步,尤其是用来构建微处理器的超大规模集成电路的迅速发展,处理器的寄存器容量、内存空间及指令数量也都由少增多。

但是,从 VAX11 到 PPC601,指令集包含的指令数却减少了。这是因为 PPC601 采用了一种新的计算机体系结构设计思想。这个设计思想就是所谓的精简指令集计算机 (RISC, Reduced Instruction Set Computer)。这个新思想认为用更少的指令完成更多的任务效率更高。这种“少”就是“多”的概念起源于 John Cocke 在 20 世纪 70 年代设计的 IBM 801 机。但 RISC 这个术语及这方面的研究和发展的基本是由加州大学伯克利分校 (University of California at Berkeley) 的 David Patterson