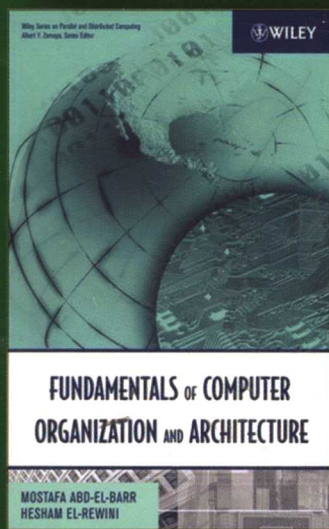


计算机组成与 体系结构

Fundamentals of Computer Organization and Architecture



[科] Mostafa Abd-El-Barr 著
[美] Hesham El-Rewini

陆鑫达 翁楚良 林新华 译



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

计算机组成与体系结构

Fundamentals of
Computer Organization and Architecture

[科] Mostafa Abd-El-Barr 著

[美] Hesham El-Rewini

陆鑫达 翁楚良 林新华 译

电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书主要论述计算机组成和计算机体系结构中的基础问题, 主要内容包括: 计算机系统导论, 指令集体系结构和设计, 汇编语言程序设计, 计算机算术, 处理部件设计, 存储系统设计, 输入/输出系统设计及组成结构, 流水线设计技术, RISC 计算机, 以及多处理器导论。

本书取材合理, 内容精练, 结构清晰, 可作为计算机专业及电气工程专业高年级本科生的教材, 也可作为相关工程师和程序设计人员的参考书。

Mostafa Abd-El-Barr, Hesham El-Rewini: **Fundamentals of Computer Organization and Architecture.**

ISBN 0-471-46741-3

Copyright © 2005, John Wiley & Sons, Inc.

All Rights Reserved. Authorized translation from the English language edition published by Wiley Publishing, Inc.

No part of this book may be reproduced in any form without the written permission of Wiley Publishing, Inc.

Simplified Chinese translation edition Copyright © 2005 by Publishing House of Electronics Industry.

本书中文简体字翻译版由 Wiley Publishing, Inc 授予电子工业出版社。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。

版权贸易合同登记号 图字: 01-2004-5757

图书在版编目 (CIP) 数据

计算机组成与体系结构 / (科) 巴尔 (Barr, M. A.) 等著; 陆鑫达等译.

北京: 电子工业出版社, 2005.12

(国外计算机科学教材系列)

书名原文: Fundamentals of Computer Organization and Architecture

ISBN 7-121-01955-8

I. 计... II. ①巴... ②陆... III. ①计算机体系结构 - 教材 IV. TP303

中国版本图书馆 CIP 数据核字 (2005) 第 135153 号

责任编辑: 贺瑞君 特约编辑: 吴 颖

印 刷: 北京市顺义兴华印刷厂

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: 12.25 字数: 345 千字

印 次: 2005 年 12 月第 1 次印刷

定 价: 23.00 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 zlt@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 |

译 者 序

本书是将“计算机组成”和“计算机体系结构”内容组合在一起的一本教材。这样做的最大优点是,可避免过去分开编写“计算机组成”和“计算机体系结构”这两门教材时必不可少的内容重复和交叉,从而导致两者不一致的缺陷,应该说这是本书的特色之一。此外,本书在指令集体系结构和设计这一章之后,专门叙述了有关汇编语言程序设计的内容,使读者对汇编语言程序设计的基本概念和具体的编程方法有了初步的掌握,这是本书的另一特色。本书的内容包括:指令集体系结构和设计,汇编语言程序设计,计算机算术,处理部件设计,存储系统设计,输入/输出设计和组成结构,流水线技术,RISC 计算机,以及多处理机系统导论。对计算机发展的历史背景也做了较全面和公正的论述。本书中对关键概念的叙述,往往通过实例来加以说明,尽可能地展示了这些概念的实际应用。可以说这是本书的又一特色。此外,本书的每一章都有小结、适量的习题,以及相应的参考文献。

本书各章节组织的逻辑性较强,叙述浅显易懂,从而能使读者较快地掌握计算机专业的关键知识点。本书适合作为计算机科学、计算机工程和电气工程本科生的计算机专业课教材,对在职的工程师和程序员也是一本很有价值的参考书。

本书的翻译工作由陆鑫达教授主持,并翻译了目录、前言、第 1~3 章、第 11 章以及索引。翁楚良讲师翻译了第 4~7 章,林新华助教翻译了第 8~10 章。译稿全文由陆鑫达教授进行了统一和校对。

由于翻译时间较紧,译文中若有错误或不妥之处,敬请广大读者不吝指正。

最后,要感谢电子工业出版社的有关策划和编辑人员,正是他们的大力支持才使本书得以顺利出版。

前 言

本书主要面向计算机工程、计算机科学和电气工程等专业的学生,适合作为一学期的“计算机组成和汇编语言”课程及一学期的“计算机体系结构”课程的教材,并且本书假定准备学习“计算机组成”或“计算机体系结构”的学生已经学习过“数字逻辑设计基础”课程和“高级计算机语言导论”等课程。

本书作者长期从事计算机组成和计算机体系结构课程的教学工作,本书正是他们 15 年教学经验的结晶。书中所用到的大部分材料已在大学本科课堂中使用过。本书着眼于两个基本的计算机视角。第一个是程序员的视角,而第二个是计算机整体结构和功能的视角。前一个视角着眼于通常在低年级计算机组成和汇编语言课程中所教授的内容,而第二个视角则涉及了通常在高年级计算机体系结构课程中所教授的内容。下面,我们将逐章对本书中的内容进行介绍。这样做的目的在于向教师、学生以及实践工程师、科学工作者提供足够多的信息,以便帮助他们选择其中合适的一章或若干章节进行阅读或研究。

第 1 章是后续各章的基础。在这一章中,我们的论述将从对计算机系统发展简短历史的回顾开始,其目的在于让读者了解影响计算机性能的因素,并希望能预测计算机的未来发展。此外还将引入有关通用和专用机的一般性问题。计算机系统可以通过处于若干抽象层间的接口加以定义,其中每一层将为它的上一层提供功能支持。处于应用程序和高级语言之间的接口称为语言体系结构。指令集体系结构定义了基本指令集和运行时间及 I/O 控制之间的接口。另一个有关计算机体系结构的不同定义建立在四个基本概念之上,它们是结构、组成、实现和性能。结构定义了各个硬件部件之间的互连;组成定义了各个部分之间的动态相互影响和管理;实现定义了硬件部分的详细设计,而性能说明了计算机系统的行为。第 1 章中还包括了体系结构的发展和风格。第 1 章的最后一部分将专门讨论不同 CPU 性能的评测。

第 2 章和第 3 章将介绍指令集体系结构和汇编语言编程。第 2 章涉及指令集体系结构与设计中的基本原理。我们将从向存储器存储信息和从存储器中检索信息这一问题开始,接着讨论若干不同的寻址方式。此外还将用一定篇幅解释指令的执行和排序。这一章在编写完成简单编程任务的示例代码段时将展示所提及的寻址方式的应用,以及一些指令特征。在第 2 章内容的基础上,第 3 章将考虑有关汇编语言的编程问题。我们将介绍由程序员所看到的假想机以及用来表示机器模型中不同指令的助记符和语法。此后将讨论汇编程序的执行和 X86 Intel CISC 系列汇编语言示例。

第 4 章和第 5 章涉及算术电路的设计和分析,以及中央处理器(CPU)的设计。第 4 章将向读者介绍有关算术操作以及用于支持计算机中计算的电路的基本问题。这里首先将介绍数的表示、基数转换和整数算术,特别要介绍用来完成整数加法、减法、乘法和除法的一些算法以及硬件方法。对于浮点算术,我们将介绍浮点表示、浮点操作以及浮点硬件的设计。第 5 章所涉及的主要问题是有关 CPU 组成和设计的。CPU 的主要功能是执行存储在计算机存储器中的一组指令。一个简单的 CPU 由一组寄存器、算术逻辑部件(ALU)和控制部件(CU)组成。其中首先介绍了取指令的读取-执行周期以及 CPU 寄存器组设计的基本原理。这些基本原理已应用于实际机器的设计,如 80x86 和 MIPS 系列,此外还详细讨论了典型 CPU 数据通路和控制器的设计。

第6章和第7章专门讨论存储器系统设计。一个典型的存储器层次结构从一个小而贵但相当快速的、称为高速缓存存储器(cache)的部件开始。层次结构中在cache之下的是较大、稍便宜但相对较慢的主存储器部件。高速缓存存储器和主存储器由固体半导体材料构成。在它们之下的是一个通常由磁盘(硬盘)和磁带组成的磁存储器,该磁存储器很大,但较便宜且速度很慢。第6章首先分析了影响成功构成一台计算机的存储器层次结构的因素,这一章的其余部分则叙述了高速缓冲存储器的设计和分析。有关主存储器和虚拟存储器的设计和分析则在第7章中论述。第7章还将对不同只读存储器的实现进行讨论。

在任何现代计算机系统中,输入/输出(I/O)都起着关键的作用。对I/O操作、部件和接口基本原理的清晰理解和正确评价是极其重要的。第8章的重点是研究I/O的设计和组成。我们将涉及有关可编程和中断驱动I/O的基本问题。此外将对80x86和MC9328MX1/MXL AITC实际机器中的中断体系结构进行解释。然后我们将讨论直接存储器访问(DMA)、总线(同步和异步)及其仲裁方法。第8章的内容将以对I/O接口的讨论结束。

提高一个处理器的指令执行速率存在两种基本技术,即增加时钟速率,以便减少指令的执行时间,或是增加可同时执行的指令数。流水线技术和指令级并行是后一技术的例子。流水线技术是第9章讨论的重点。它的基本思想是使处理器同时处理多条指令。这可以通过将一条指令的执行分解到若干子部件(级)上来完成,每一个子部件完成所需操作的一部分,即取指令、指令译码、取操作数和存结果。我们将介绍流水线处理器的性能衡量,讨论造成指令流水冲突的主要问题及某些可能的解决方法。此外,我们还将描述算术流水线的概念,以及在设计这种流水线时将会遇到的问题。我们将通过介绍ARM 1026EJ-S和UltraSPARC-III这两个流水线处理器来结束这一章。

第10章专门介绍精简指令集计算机(RISC)。这类机器代表了计算机体系结构的重要转变。RISC强调合理分配资源来增强体系结构,从而使得执行最常用和最耗时的指令变得最为有效。基于RISC的机器可以由一些共同属性描述,比如简单而精练的指令集、固定的指令格式、一个机器周期一条指令、流水线的指令读取和执行部件、数量充足的通用寄存器(或优化的编译器代码生成),载入/存储式的存储器操作和硬连线的控制器设计。这一章中,我们的讨论从RISC体系结构的发展开始。然后讨论RISC发展中的一个基本概念:重叠寄存器窗口。最后介绍了应用基本RISC原理制造的机器,如BerkeleyRISC、Stanford MIPS、Compaq Alpha和SUN UltraSparc。

在讨论了多处理器设计和分析的基本问题并指出单流机器的主要限制之后,第11章介绍多处理器系统的基本概念。在多处理器系统中,一些(两个或多个)处理器以某种方式相连,从而允许它们同时参与一个单一任务。使用多处理器的优点是只需简单连接许多较小的现有计算机就可以创建出功能强大的计算机。此外,由许多单处理器系统组成多处理机系统比构造一台高性能的单处理器系统更为经济。我们介绍了许多连接多个处理器时所使用的拓扑结构、计算机体系结构的不同分类方案和基于拓扑的互连网络的分类。对MIMD(多指令多数据)的两种存储器组成方案也做了介绍,即共享存储器和消息传递。这一章以关于多处理器的分析和性能指标的内容结束。感兴趣的读者可以参阅《先进计算机体系结构与并行处理》^①一书。

从以上对本书各章的介绍中我们可以清楚地看到本书的各章在很大程度上是相互独立、自成体系的。我们相信这样的方式有助于教师针对课程目的有选择性地讲授部分章节。然而,经验告诉我们,第1章到第5章以及第8章的内容适用于“计算机工程”及“电气工程”学生的“计算

① 该书与本书同期出版,电子工业出版社——编者注。

机组成和汇编语言”课程,而第 1 章、第 6 章、第 7 章、第 9 章到第 11 章适用于“计算机体系结构”课程。实践工程师和科学工作者可以通过本书各章节的预览有选择性地参阅单独的一章或某些章节。例如,如果想了解更多存储器设计的内容,可以参考第 6 章和第 7 章的内容。

致谢

我们要向所有协助完成此书的人表示由衷的感谢。他们是 Saskatchewan, SMU; KFUPM 大学“计算机组成和计算机结构”课程的学生; Kuwait 大学使用了本书一些章节的草稿并提出了宝贵的反馈意见和建议,对他们表示感谢。我们的同事 Donald Evan, Fatih Kocan, Peter Seidel, Mitch Thornton, A. Naseer, Habib Ammari 和 Hakki CanKaya 提出了积极的意见和很好的建议,使得本书在形式和表述方面得到了明显的改善。在 John Wiley 公司的安排下我们还接受了许多不知名书评家的建议和指正。此外,还要特别感谢这套丛书的编辑 Albert Y. Zomaya 和 John Wiley 公司的 Val Moliere, Kirsten Rohstedt 和 Christine Punzo,是他们使这本书最终成型。当然,此书还存在许多错误和不足之处,欢迎读者批评指正。最后,我们要感谢我们的家人,感谢他们在本书编写过程中给予我们的支持和鼓励。

Mostafa Abd-El-Barr
Hesham El-Rewini

目 录

第 1 章 计算机系统导论	1
1.1 历史背景	1
1.2 体系结构的发展和风格	3
1.3 技术进展	3
1.4 性能衡量	4
1.5 小结	8
习题	8
参考文献和推荐读物	10
参考网站	10
第 2 章 指令集体系结构和设计	11
2.1 存储单元和操作	11
2.2 寻址方式	13
2.3 指令类型	19
2.4 程序设计举例	22
2.5 小结	24
习题	24
参考文献和推荐读物	25
第 3 章 汇编语言程序设计	26
3.1 一台简单机器	26
3.2 指令助记符和句法	28
3.3 汇编器指令和命令	30
3.4 程序汇编和执行	31
3.5 举例:x86 系列	34
3.6 小结	40
习题	40
参考文献和推荐读物	41
第 4 章 计算机算术运算	42
4.1 数字系统	42
4.2 整数运算	44
4.3 浮点运算	53
4.4 小结	56
习题	56
参考文献和推荐读物	57

第 5 章 处理单元设计	58
5.1 CPU 基础	58
5.2 寄存器组	59
5.3 数据通路	62
5.4 CPU 指令周期	64
5.5 控制器	66
5.6 小结	73
习题	74
参考文献和推荐读物	75
第 6 章 存储系统设计 I	76
6.1 基本概念	76
6.2 高速缓冲存储器	77
6.3 小结	93
习题	93
参考文献和推荐读物	94
参考网站	95
第 7 章 存储系统设计 II	96
7.1 主存	96
7.2 虚拟存储器	101
7.3 只读存储器	111
7.4 小结	112
习题	113
参考文献和推荐读物	114
参考网站	114
第 8 章 输入/输出的设计与组成	115
8.1 基本概念	115
8.2 编程 I/O	117
8.3 中断驱动 I/O	119
8.4 直接存储器访问	125
8.5 总线	127
8.6 I/O 接口	129
8.7 小结	130
习题	131
参考文献和推荐读物	131
参考网站	132
第 9 章 流水线设计技术	133
9.1 一般概念	133
9.2 指令流水线	134
9.3 流水线处理器实例	144

9.4 指令级并行	148
9.5 算术流水线	149
9.6 小结	152
习题	152
参考文献和推荐读物	153
参考网站	154
第 10 章 精简指令集计算机	155
10.1 RISC/CISC 进化周期	155
10.2 RISC 设计原则	155
10.3 重叠寄存器窗口	157
10.4 RISC 对 CISC	157
10.5 先驱 RISC 机器	159
10.6 高级 RISC 机器实例	162
10.7 小结	166
习题	166
参考文献和推荐读物	167
参考网站	167
第 11 章 多处理机系统导论	168
11.1 引言	168
11.2 计算机体系结构的分类	169
11.3 SIMD 方式	174
11.4 MIMD 方式	176
11.5 互连网络	179
11.6 分析和性能指标	181
11.7 小结	181
习题	181
参考文献和推荐读物	182

第 1 章 计算机系统导论

计算机工业的技术进展是两支主力军做出一系列巨大努力的结果。这两支主力军就是以大学研究中心为代表的学术界和以计算机公司为代表的工业界。但是,公平地讲,当今计算机工业的技术进展应归功于大学研究中心的始创。为了了解当前计算机工业中的技术进展,人们不得不对计算机的历史和它的发展进行追溯。对历史回顾旨在了解影响今天的计算机性能的因素,并希望能预测计算机的未来发展。我们日常生活中所使用的绝大多数计算机都是通用机。这些机器在构造时并不考虑专门的应用,而是考虑能完成许多不同应用所需的性能。这些通用机不同于为特殊用途服务(制造)的专用机。在 1.1 节中,我们将对计算机诞生和发展的历史背景进行简短的叙述。

计算机系统已通过若在若干抽象层上的接口加以定义,其中每一层将为它的上一层提供支持。这些层包括应用程序、高级语言和机器指令集。根据系统不同级间的接口,就可以定义许多计算机体系结构。处于应用程序和高级语言之间的接口称为语言体系结构,而指令集体系结构则定义了基本指令集与运行时间及 I/O 控制之间的接口。计算机体系结构的另一个不同定义建立在四个基本概念之上,它们是结构、组成、实现和性能。结构定义了各个硬件部件之间的互连,组成定义了各个部件之间的动态相互影响和对它们的管理,实现定义了硬件部件的详细设计,而性能说明了计算机系统的行为。有关体系结构的发展和风格将在 1.2 节中叙述。

1.3 节中将论述许多技术的发展情况。本章的最后将详细讨论 CPU 性能的衡量。

1.1 历史背景

在这一节中,我们将提供一个计算机工业发展的历史背景。我们在一开始就要指出建造计算机并不是源出一处。有充分理由相信建造第一台计算机的尝试存在于地球上不同的地方,我们也坚定地相信建造一台计算机需要合作。因此,我们只提及机器和它的所在地,而不把它们与某个研究者的名字联系在一起。

较为公平地说,曾经建造过的第一台程序控制(机械)计算机是 Z1(1938 年)。接着在 1939 年出现了第一台定点算术操作程序控制计算机 Z2。在大学中,第一次有记载的建造计算机的尝试可以追溯到 20 世纪 40 年代早期的美国依阿华州立大学。该大学的研究者们建造了一台小型的专用电子计算机,但这台机器从未真正运行过。几乎在同时,即 1941 年,在德国有报告说完成了功能完全可编程的专用机器 Z3 的设计。有迹象表明由于缺少资金,该设计无法得以实现。历史还记载在进行这两个建造专用电子计算机的尝试时,来自世界上不同地区的研究者们通过对正在从事该项研究工作的实验室和研究所的访问获得了第一手经验。可以设想正是这种访问和思想的交流,使得访问者在返回他们自己的实验室后开始了类似的项目研究。

就通用机而言,记载表明宾夕法尼亚大学于 1944 年主持建造了 ENIAC(电子数字积分器和计算器)机器。这是第一台用真空管建造的、实际运行的通用机。在第二次世界大战中,这台机器主要用来帮助计算火炮轨迹表。借动手动的开关设置和电缆的插接可对该机器进行编程。按照今天的标准这台机器是很慢的,它只有很有限的存储容量和原始的可编程性。该大学还提出了一个改进的 ENIAC 版本,称为 EDVAC(电子离散变量自动计算机),该机器试图改进程序的输入方法,启发了开发存储程序式计算机的探索。直到 1952 年,EDVAC 的研究项目才完成。受 ENIAC 实现思想的启发,普林斯顿高级研究所(IAS)的研究者们建造了 IAS 机器(1946 年),它的速度比 ENIAC 快了约 10 倍。

1946 年,当 EDVAC 研究项目尚在进行中时,剑桥大学启动了一个类似的研究项目。该项目试图建造一台存储程序计算机,称为电子延迟存储自动计算器(EDSAC)。1949 年,EDSAC 成为了世界上第一台完整的、存储程序的全运行计算机。由 EDSAC 带来的副产品是在哈佛建造了一系列的机器。该系列机包括 MARK I, II, III 和 IV。后两台机器引入了分离的指令存储器和数据存储器。后人对这些机器赋予了“哈佛体系结构”(Harvard Architecture)的概念,以表明使用了分离的存储器。注意,“哈佛体系结构”今天则用来描述具有分离的指令高速缓存和数据高速缓存。

1951 年,第一台通用的商用计算机——UNIVAC(通用自动计算机)I 面市了,它是对 1949 年建造的 BINAC(二进制自动计算机)的改进。IBM 公司在 1952 年发布了它的第一台计算机 IBM 701。但在 20 世纪 50 年代早期,计算机工业发展遭遇减速。此后,在 1964 年,IBM 公发布了 IBM 360 系列机。该系列机包括许多价格和性能不同的型号。该系列机的出现导致了 DEC 推出第一台小型计算机 PDP-8,它当时被认为是价格相当低廉的机器。而 Intel 公司则于 1971 年推出了第一台微处理器 Intel 4004。当苹果计算机系列于 1977 年首次推出时,世界上第一台个人计算机诞生了。1977 年,DEC 公司又推出了 VAX-11/780。此后,Intel 又推出了 80x86 微处理器系列。

1977 年,由 Altair, Processor Technology, North Star, Tandy, Commodore, Apple 以及许多其他公司推出的个人计算机(PC),在许多行业都提高了终端用户的生产率。由 Compaq, Apple, IBM, Dell 以及许多其他公司生产的个人计算机很快得到了普及,并改变了计算的面貌。

与小型机同时出现的是超级计算机。由 Control Data 公司于 1961 年推出的 CDC 6600 是第一台这类超级计算机,而 Cray 研究公司则在 1976 年推出了具有最好性价比的超级计算机 Cray-1。

20 世纪 80 年代和 90 年代,业界推出了许多具有多处理器的商用并行计算机。它们一般可分为两大类:(1)共享存储器系统;(2)分布式存储器系统。在单机中所包含的处理器数从在共享存储器计算机中的几个到在大规模并行系统中的几百至几千个。这一时代的并行机例子包括 Sequent Symmetry, Intel iPSC, nCUBE, Intel Paragon, Thinking Machines(CM-2, CM-5), MsPar(MP), Fujitsu(VPP500),以及其他机器。

计算发展的一个明显趋势是集中式的服务器将为计算机网络所替代。这些由网络连接的、廉价但功能强大的台式机能形成极强的计算能力。约在 1990 年,功能强大的个人计算机和工作站局域网(LAN)开始替代大型机和小型机。这些单独的台式机很快就用广域网(WAN)连接成更大的计算联合体。

因特网的普及使网络计算以及近期的网格计算变得更加引人入胜。网格是一种地域分布式计算平台,它们将能提供可信赖、一致而且廉价的对高端计算设施的访问。

表 1.1 是 Lawrence Tesler(1995)所提出的表的修改版本。该表列出了自 20 世纪 60 年代开始与每个计算年代相关的不同计算模式的主要特征。

表 1.1 计算的 4 个年代

特征	批处理	分时	台式机	网络
年代	20 世纪 60 年代	20 世纪 70 年代	20 世纪 80 年代	20 世纪 90 年代
场所	计算机房	终端室	桌面	移动
用户	专家	专业人员	个人	小组
数据	字母数字	文本, 数字	字体, 图形	多媒体
对象	计算	访问	显示	通信
界面	穿孔卡片	键盘和 CRT 显示器	观看和指点	询问和告知
操作	处理	编辑	布局	协调
连接性	无	外设电缆	局域网	因特网
拥有者	公司计算中心	分区的 IS 商店	部门终端用户	每个人

CRT, 阴极射线管; LAN, 局域网。

1.2 体系结构的发展和风格

计算机体系结构设计师总是尽力增强他们所设计的体系结构的性能。为达到这一目的他们已采用了许多方法, 其中的一个设计思想是在一条指令中做尽可能多的工作, 这样就可使用较少的指令数完成相同的作业。这一思想的直接结果是只需较少的存储器读/写操作, 而获得最终的运算加速。这一设计思想的另一个优点是, 增加指令的复杂性和寻址方式具有理论上的优点, 那就是可以减小高级语言指令和低级(机器)语言指令间的“语义差别”。用一条(机器)指令将几个二进制编码的十进制数(BCD)转换成二进制数是一个很好的例子, 它可以说明某些指令要完成的动作有多复杂。众多数量的寻址方式(在 VAX 机器中多达 20 多种)进一步增加了指令的复杂性。追随这一设计思想的机器称为复杂指令集计算机(CISC)。CISC 机器的例子包括 Intel Pentium, Motorola MC 68000 以及 IBM & Macintosh Power PC。

注意, 将更多的能力加到处理器会增加支持更高时钟速率的难度, 而这本来是能够做到的。难度增加是因为增加了在单个时钟周期内的计算复杂性。从 20 世纪 70 年代中期到 20 世纪 80 年代早期, 许多研究也指明了在典型的程序中, 所执行的指令 80% 以上是那些使用赋值语句、条件转移和过程调用的指令。此外, 简单赋值语句占据了这些操作的几乎 50%。这些发现导致了另一种不同设计思想的出现。在减少指令复杂性和寻址方式的同时, 该设计思想通过加速那些最常用的操作来促进体系结构的优化。采用这一设计思想的机器称为精简指令集计算机(RISC)。RISC 机器的例子包括 Sun SPARC 和 MIPS 机器。

以上体系结构设计的两种设计思想已导致了无法调和的争论, 即哪一种体系结构风格为“最好”。但是, 应该注意的是, 研究已经表明 RISC 体系结构的确可以使程序更快地执行。大多数当代的微处理器芯片均追随 RISC 模式。本书将介绍 CISC 和 RISC 机器各自的突出特点以及相应的示例。

1.3 技术进展

计算机技术正以空前的速度演进, 包括处理器和存储器的发展。的确, 技术改进刺激了计算机工业的发展。在单个芯片上集成的晶体管数(一个晶体管是一个可控的开关)已从几百个增加到以百万计, 这种令人难以置信的增加得益于晶体管制造工艺的进步。

集成的规模已从小规模(SSI)、中规模(MSI)、大规模(LSI),发展到了超大规模(VLSI)乃至最近的晶片规模(WSI)。表 1.2 给出了各种工艺中每个芯片上的典型器件数。

表 1.2 芯片上的器件数

集成度	工艺	典型器件数	典型功能
SSI	双极	10 ~ 20	门和触发器
MSI	双极和 MOS	50 ~ 100	加法器和计数器
LSI	双极和 MOS	100 ~ 10 000	ROM 和 RAM
VLSI	CMOS(大多数)	10 000 ~ 5 000 000	处理器
WSI	CMOS	> 5 000 000	数字信号处理和专用

应该提及的是,最小器件特征尺寸的持续减小导致了芯片上器件数量的持续增长,而这又促成了更多方面的进展。其中在 RAM 存储器中器件数的增长又有助于设计者平衡存储器的大小和速度。特征尺寸大小的改进为引入改进的设计风格提供了极好的机会。

1.4 性能衡量

在这一节中,我们将论述评估计算机性能的重要问题,并着重讨论用来评估计算机的许多性能衡量方法。让我们在一开始就明确:计算机的性能可以反映在不同方面。例如,用户衡量计算机的性能是基于执行一个给定作业(程序)所需的时间,而实验室工程师衡量其系统的性能则是基于在给定的时间内所完成的总工作量。用户将程序的执行时间作为性能的衡量标准,而实验室工程师则认为吞吐量是更为重要的性能衡量标准。评估计算机性能的度量标准有助于比较不同的设计。

性能分析应有助于回答这样的问题,即使用给定的计算机,程序能执行得多快?为了回答这一问题,我们需要确定由一台计算机执行一个给定作业所需的时间。我们定义时钟周期时间为一个周期时钟信号(参见图 1.1)的两个连续上升(下降)沿间的时间。时钟周期能用来为单位计算计数,因为计算结果的存储是与上升(下降)时钟沿同步的。计算机执行一个作业所需的时间通常以时钟周期数来表示。

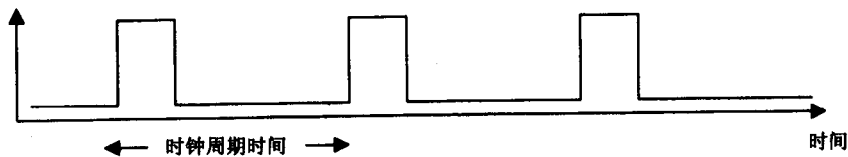


图 1.1 时钟信号

我们用 CC (cycle count)表示执行一个作业所需的 CPU 时钟周期数,用 CT (cycle time)表示周期时间,并用 $f = 1/CT$ 表示时钟频率。则由 CPU 执行一个作业所需的时间可表达为

$$\text{CPU 时间} = CC \times CT = CC/f$$

计算在一个给定的程序中所执行的指令数,比计算执行该程序所需的 CPU 时钟周期数可能更为容易。为此,每条指令所需的平均时钟周期数(CPI , clock cycles per instruction)已作为另一种性能衡量的指标。下面的方程说明了如何计算 CPI 。

$$CPI = \frac{\text{程序的 CPU 时钟周期}}{\text{指令数}}$$

$$\text{CPU 时间} = \text{指令数} \times CPI \times \text{时钟周期时间}$$

$$= \frac{\text{指令数} \times CPI}{\text{时钟速率}}$$

已知一台给定机器的指令集是由 ALU(简单的赋值、算术及逻辑指令)、load(装载)、store(存储)和 branch(转移)等指令类型所组成。则在已知每一种指令 CPI 的情况下,总 CPI 就可计算如下:

$$CPI = \frac{\sum_{i=1}^n CPI_i \times I_i}{\text{指令数}}$$

其中, I_i 是程序中执行指令类型 i 的次数,而 CPI_i 则表示执行这类指令所需的平均时钟周期数。

例 1.1 根据执行一组基准测试程序所记录的以下性能测量值(如表 1.3 所示),计算机 A 的总 CPI 。假定 CPU 的时钟速率为 200 MHz。

表 1.3 例 1.1 表

指令类型	出现的百分比	每条指令的周期数
ALU	38	1
Load 和 store	15	3
Branch	42	4
其他	5	5

假定执行 100 条指令,则总 CPI 可计算如下:

$$CPI_A = \frac{\sum_{i=1}^n CPI_i \times I_i}{\text{指令数}} = \frac{38 \times 1 + 15 \times 3 + 42 \times 4 + 5 \times 5}{100} = 2.76$$

注意, CPI 反映的是处理器的组成和指令集的体系结构,而指令数反映的是指令集的体系结构和所使用的编译技术。这两个参数是相辅相成的,所以在评估一台给定的计算机或比较两台机器的性能时,必须同时考虑 CPI 和指令数两者。

近年来受到较多关注的另一个性能衡量指标是 $MIPS$ (每秒百万指令,即单位时间的指令执行速率),它定义为

$$MIPS = \frac{\text{指令数}}{\text{执行时间} \times 10^6} = \frac{\text{时钟速率}}{CPI \times 10^6}$$

例 1.2 假设在另一台机器 B 上执行前面所提及的相同基准测试程序组,所记录的测量值如表 1.4 所示。

表 1.4 例 1.2 表

指令类型	出现的百分比	每条指令的周期数
ALU	35	1
Load 和 store	30	2
Branch	15	3
其他	20	5

现假定时钟速率为 200 MHz,则前一例中机器 A 和本例中机器 B 的 $MIPS$ 各为多少?