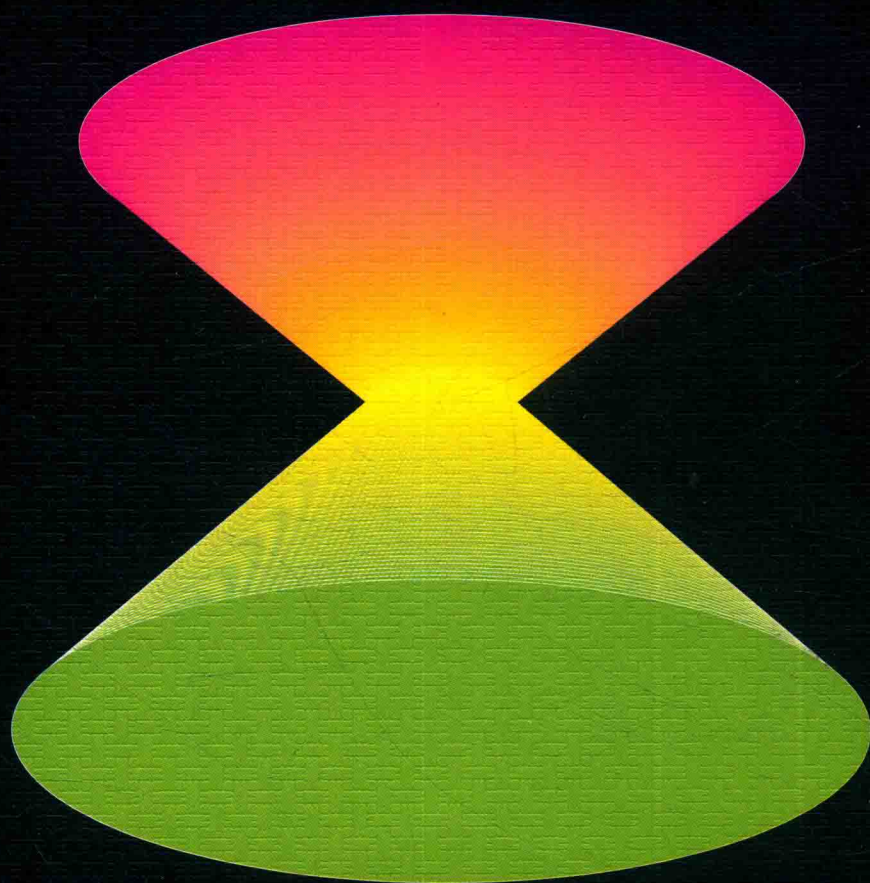


世界著名计算机教材精选

计算机体系结构

Gerard Blanchet 著
Bertrand Dupouy

陈宗斌 等译



COMPUTER ARCHITECTURE

清华大学出版社



世界著名计算机教材精选

计算机体系结构

Gerard Blanchet Bertrand Dupouy 著

陈宗斌 等译

清华大学出版社
北 京

Gerard Blanchet, Bertrand Dupouy

Computer Architecture

EISBN: 978-1-118-49653-4

Copyright © 2015 by Wiley Publishing, Inc.

All Rights Reserved. This translation published under license.

Simplified Chinese translation edition is published and distributed exclusively by Tsinghua University Press under the authorization by John Wiley & Sons, Inc., within the territory of the People's Republic of China only, excluding Hong Kong, Macao SAR and Taiwan. Unauthorized export of this edition is a violation of the Copyright Act. Violation of this Law is subject to Civil and Criminal Penalties.

本书中文简体字翻译版由美国 John Wiley & Sons, Inc. 公司授权清华大学出版社在中华人民共和国境内(不包括中国香港、澳门特别行政区和中国台湾)独家出版发行。未经许可之出口,视为违反著作权法,将受法律之制裁。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。

北京市版权局著作权合同登记号图字 01-2015-5034 号

本书封面贴有 John Wiley & Sons 公司防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

计算机体系结构/(美)杰勒德·布兰切特(Gerard Blanchet),(美)伯特兰得·杜波乌(Bertrand Dupouy)著;陈宗斌等译. —北京:清华大学出版社,2017

书名原文:Computer Architecture

(世界著名计算机教材精选)

ISBN 978-7-302-46064-0

I. ①计… II. ①杰… ②伯… ③陈… III. ①计算机体系结构—高等学校—教材 IV. ①TP303

中国版本图书馆 CIP 数据核字(2017)第 005445 号

责任编辑:龙启铭 薛 阳

封面设计:傅瑞学

责任校对:焦丽丽

责任印制:李红英

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:清华大学印刷厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:17.5 字 数:413 千字

版 次:2017 年 7 月第 1 版 印 次:2017 年 7 月第 1 次印刷

印 数:1~2000

定 价:39.00 元

产品编号:058686-01

前 言

本书介绍了理解计算机的操作系统所需掌握的概念。本书基于以下几个方面编写。

(1) 计算机的组件以电子方式工作的细节超出了本书的范围。

(2) 本书重点放在概念上，关注的是机器的体系结构的构件、它们的功能以及它们的交互。

(3) 本书在任何必要的地方强调了软件与硬件资源之间必不可少的联系。

出于清晰起见，我们精心选择了一些示例，它们适用于所有领域的机器，而不会稀释书中的内容。这种选择可以帮助我们说明从第一代计算机起技术、概念和性能是如何演进的。

全书分为 5 个部分。前 4 部分的难度渐增，它们构成了本书的核心，这 4 个部分分别是“基本体系结构的元素”“编程模型和操作”“存储器层次结构”和“并行性和性能增强”。最后一部分包含附录，提供了本书中的练习的提示和解决方案以及编程模型。读者可以基于先前的知识和目标独立地阅读每个部分。

本书中 5 个部分的简介如下。

1. 基本体系结构的元素

(1) 第 1 章介绍了处理器的主要构件的发展史。

(2) 第 2 章详细列出了基本的模块及其特性，并且描述了它们是怎样连接的。

(3) 第 3 章重点介绍了信息的表示：整数、浮点数、定点数和字符。

2. 编程模型和操作。

(4) 第 4 章解释了指令集与体系结构之间的关系。

(5) 第 5 章提供了一个指令执行的详细示例，以阐明用于管控处理器操作的内部机制。还介绍了一些额外的元素，比如协处理器和矢量扩展。

(6) 第 6 章描述了与外设之间交换信息所涉及的一些规则：轮询、直接存储器存取和中断。

3. 存储器层次结构

(7) 第 7 章说明了一些关于存储器组织结构的元素：层次结构、分段和分页。

(8) 第 8 章介绍了缓存组织结构和访问方法。

(9) 第 9 章描述了虚拟存储器管理概念、规则和访问权限。

4. 并行性和性能增强

(10) 第 10 章通过展示流水线体系结构介绍了并行性：概念以及软件和硬件冲突解决方案。

(11) 第 11 章给出了 DLX 体系结构作为示例。

(12) 第 12 章讨论了多处理器环境中的缓存管理，还讨论了相干性 (Coherence) 和协议 (MSI、MEI 等)。

(13) 第 13 章展示了超标题体系结构冲突的操作、记分板和 Tomasulo 算法以及 VLIW 体系结构。

5. 附录

在本书附录中可以找到关于在不同章节中使用的编程模型以及给出的关于练习的提示和解决方案的补充材料。

本书非常适合用作高等院校计算机专业本科生的教材，对于从事计算机体系结构或计算机系统设计的程序员和工程技术人员，本书也颇具参考价值。

参加本书翻译的人员有：陈宗斌、傅强、宋如杰、蔡江林、陈征、戴锋、蔡永久、龚宇华、何正雄、黄定光、孙朝辉、孙丽、徐茜、许瑛琪、叶守运。

目 录

第 1 部分 基本体系结构的元素

第 1 章 简介	3
1.1 历史背景	3
1.1.1 自动化和机械计算器	3
1.1.2 从外部程序到存储程序	5
1.1.3 各代计算机简介	7
1.2 内部操作简介	10
1.2.1 与机器通信	10
1.2.2 执行指令	11
1.3 未来展望	12
第 2 章 基本模块	14
2.1 存储器	14
2.1.1 定义	14
2.1.2 几个技术元件	15
2.2 处理器	16
2.2.1 功能单元	16
2.2.2 处理器的寄存器	17
2.2.3 处理单元的元素	20
2.2.4 控制器的元件	21
2.2.5 地址计算单元	22
2.3 模块之间的通信	22
2.3.1 PCI 总线	23
第 3 章 信息的表示	25
3.1 复习	25
3.1.1 以 2 为基数	25
3.1.2 二进制、八进制和十六进制表示	26
3.2 数字表示约定	27
3.2.1 整数	27
3.2.2 实数	28
3.2.3 浮点表示的一个示例——IEEE-754 标准	31
3.2.4 动态范围与精度	33
3.2.5 实现	33
3.2.6 IEEE-754 标准的扩展	34

3.3	字符表示	34
3.3.1	8 位表示	34
3.3.2	现代表示	36
3.4	练习	37

第 2 部分 编程模型和操作

第 4 章	指令	41
4.1	编程模型	41
4.1.1	I8086 的寄存器	41
4.1.2	地址构造和寻址模式	42
4.2	指令集	44
4.2.1	移动指令	44
4.2.2	算术和逻辑指令	45
4.2.3	移位指令	45
4.2.4	转移	46
4.2.5	其他指令	48
4.3	编程示例	49
4.4	从汇编语言到基本指令	51
4.4.1	汇编器	51
4.4.2	汇编阶段	52
4.4.3	连接器	53
4.4.4	何时利用汇编语言编程	54
第 5 章	处理器	55
5.1	控制总线	55
5.1.1	复位线路	56
5.1.2	保持线路	56
5.1.3	等待控制线路	56
5.1.4	中断线路	57
5.1.5	概念图	57
5.2	指令的执行：一个示例	57
5.2.1	指令的执行	59
5.2.2	时序图	62
5.3	定序器的构成	63
5.3.1	传统的合成方法	63
5.3.2	微编程	65
5.4	扩展	65
5.4.1	协处理器	65
5.4.2	矢量扩展	67
5.4.3	DSP 和 GPU	71

5.5 练习	72
第 6 章 输入和输出	74
6.1 示例	75
6.1.1 示例: 控制热电偶	75
6.1.2 示例: 串行终端连接	79
6.2 交换器的设计和寻址	81
6.2.1 交换器的设计	81
6.2.2 交换器寻址	82
6.3 交换模式	83
6.3.1 轮询交换模式	83
6.3.2 直接存储器存取	84
6.3.3 中断	88
6.4 处理中断	88
6.4.1 工作原理	88
6.4.2 示例	90
6.4.3 软件中断	91
6.4.4 屏蔽与取消屏蔽中断	91
6.4.5 中断优先级或级别	92
6.4.6 类似的机制	92
6.5 练习	93

第 3 部分 存储器层次结构

第 7 章 存储器	97
7.1 存储器资源	97
7.2 特征	97
7.3 存储器层次结构	98
7.3.1 局部性原理	99
7.3.2 层次结构的组织和管理	99
7.3.3 定义和性能	100
7.4 存储器大小和保护机制	100
7.5 分段	101
7.5.1 使用段寄存器: 一个示例	101
7.5.2 使用段描述符	102
7.6 分页	103
7.7 存储器交叉存取和突发模式	105
7.7.1 C 存取	105
7.7.2 S 存取	105
7.7.3 突发模式	106
7.7.4 预取缓冲区	107

7.8 保护, I386 的示例.....	108
第 8 章 缓存	109
8.1 高速缓冲存储器.....	109
8.1.1 操作原理和体系结构.....	109
8.1.2 高速缓冲存储器的操作.....	109
8.1.3 缓存设计.....	111
8.2 替换算法.....	114
8.2.1 LRU 方法.....	114
8.2.2 多级缓存的情况.....	118
8.2.3 性能和模拟.....	119
第 9 章 虚拟存储器	122
9.1 一般概念.....	122
9.1.1 操作.....	122
9.1.2 存取信息.....	123
9.1.3 地址翻译.....	123
9.2 存取方法的规则.....	124
9.2.1 页错误.....	124
9.2.2 多级分页.....	125
9.2.3 服务信息、保护和访问权限.....	125
9.2.4 页大小.....	126
9.3 程序执行的示例.....	127
9.3.1 引入翻译缓存.....	128
9.3.2 执行.....	129
9.3.3 备注.....	131
9.4 二级分页的示例.....	132
9.4.1 管理.....	132
9.4.2 处理服务位.....	132
9.4.3 信息访问中的步骤.....	134
9.5 页式分段.....	136
9.5.1 36 位扩展.....	138
9.6 练习.....	138
9.7 文档摘录.....	139
9.7.1 MMU 简介.....	139
9.7.2 TLB 的描述.....	140
9.7.3 TLB 特性.....	142

第 4 部分 并行性和性能增强

第 10 章 流水线体系结构	147
10.1 动机和思想.....	147

10.1.1	RISC 机器	147
10.1.2	操作原理	148
10.1.3	流水线体系结构的代价	150
10.2	流水线管理问题	150
10.2.1	结构风险	150
10.2.2	依赖冲突	153
10.2.3	转移	154
10.3	处理转移	155
10.3.1	延迟的转移和软件处理	155
10.3.2	转移预测	157
10.3.3	转移目标缓冲区	160
10.3.4	全局预测	161
10.3.5	示例	163
10.4	中断和异常	166
10.4.1	中断	166
10.4.2	陷阱和错误	167
第 11 章	体系结构的示例	168
11.1	简介	168
11.1.1	流水线描述	168
11.1.2	指令集	170
11.1.3	指令格式	171
11.2	执行指令	172
11.2.1	读取和解码指令	172
11.2.2	存储器读取	172
11.2.3	存储器写操作	174
11.2.4	寄存器到寄存器的操作	174
11.2.5	条件转移	176
11.2.6	具有立即寻址的指令	179
11.3	DLX 中的冲突解决	179
11.3.1	转发技术	179
11.3.2	处理转移	182
11.4	练习	185
第 12 章	多处理器环境中的缓存	190
12.1	缓存一致性	191
12.1.1	示例	191
12.1.2	要考虑的元素	192
12.1.3	一致性的定义	192
12.1.4	方法	193
12.2	嗅探协议的示例	194

12.2.1	MSI 协议	194
12.2.2	MEI 协议	197
12.2.3	MESI 协议	197
12.2.4	MOESI 协议	199
12.3	改进	200
12.4	基于目录的一致性协议	200
12.5	一致性	203
12.5.1	相容性与一致性	203
12.5.2	表示法	203
12.5.3	原子相容性	204
12.5.4	顺序相容性	204
12.5.5	因果相容性	205
12.5.6	弱相容性	206
12.6	练习	207
第 13 章	超标量体系结构	210
13.1	超标量体系结构的原理	210
13.1.1	风险	211
13.2	寻求解决方案	212
13.2.1	原理	212
13.2.2	示例	214
13.3	处理指令流	215
13.3.1	记分牌的原理	215
13.3.2	记分牌实现	216
13.3.3	详细的示例	217
13.3.4	关于优先级约束的注释	221
13.3.5	Tomasulo 算法的原理	222
13.3.6	详细的示例	224
13.3.7	循环执行和 WAW 风险	229
13.4	VLIW 体系结构	231
13.4.1	超标量体系结构的局限性	231
13.4.2	VLIW 体系结构概述	232
13.4.3	预测	233
13.5	练习	235

第 5 部分 附录

附录 A	提示和解决方案	241
A.1	信息的表示	241
A.2	处理器	244
A.3	输入和输出	245

A.4	虚拟存储器	246
A.5	流水线体系结构	247
A.6	多处理器环境中的缓存	253
A.7	超标量体系结构	256
附录 B	编程模型	258
B.1	I8086 中的指令编码	258
B.2	DLX 体系结构的指令集	259
B.2.1	浮点数上的操作	259
B.2.2	移动操作	259
B.2.3	算术和逻辑运算	260
B.2.4	转移	260
参考文献		262

第 1 部 分

基本体系结构的元素

第 1 章 简 介

本章在提供了一些历史背景之后,将着重介绍计算机(Computer)的主要组件[MOR 81, ROS 69, LAV 75]。这将引导我们描述计算器的类别,我们将称之为经典体系结构机器(Classic Architecture Machine)或经典体系结构单处理器(Classic Architecture Uniprocessor)。我们将研究由其中每个模块执行的功能,然后在后面的章节中更详细地描述它们。

1.1 历史背景

1.1.1 自动化和机械计算器

第一批为人所知的机械计算器[SCI 96]是由 Wilhelm Schickard (1592—1635) (大约在 1623 年)、Blaise Pascal (大约在 1642 年) 和 Gottfried Wilhelm Leibniz (1646—1716) (大约在 1673 年) 设计的 (如图 1.1 所示), 它们通过一种齿轮机械装置以 10 为基数来工作。

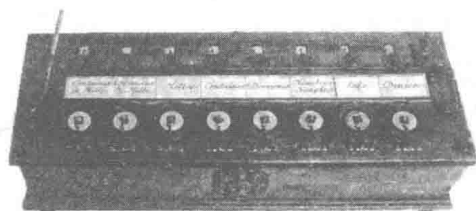


图 1.1 Blaise Pascal 的 Pascaline

由用户决定把一系列操作放在一起。自动执行一系列过程的需要最终导致了计算机的设计。

简单任务的序列已经在音乐盒、管风琴、自动钢琴的设计中得以实现,其中带有别针的圆筒、凸轮系统和穿孔纸带用于确定旋律。Joseph-Marie Jacquard (1752—1834) 设计的织布机是自动化的另一个示例 (如图 1.2 所示)。一系列穿孔卡片指示要执行的基本操作序列: 每个孔允许一根针穿过, 支撑卡片四面体以与梭子相同的速度旋转, 其中后者携带有被编织的线。Jacquard 的发明是在 1804—1805 年提出的, 并于 1806 年被法国正式认可为一种公共利益。1812 年, 法国有 11 000 台这样的织布机[ENC 08]。在里昂周围的工场里, 仍然可以发现一些织布机在工作。

这个系统可以让我们初识后来出现的基于可编程自动机或计算器的设备, 它们专用于控制工业流程。

Charles Babbage (1792—1871) 是从事将自动机与机械计算器结合起来的机器设计的第一人。他已经设计了一款计算器, 即差分机 (Difference Engine), 可以在伦敦的科学博物馆看到它。1841 年在都灵举行的研讨会上, 他展示了一个用于更通用机器的项目。他与



图 1.2 Jacquard 的织布机的示例，由法国里昂的 La Maison des Canuts 惠赠

Ada Lovelace (Byron 勋爵的女儿) 合作，允许他描述更详细、更雄心勃勃的机器，它预示了我们的现代计算机。这台机器（称为分析机 (Analytical Engine) [MEN 42]）可以自主地执行算术运算序列（如图 1.3 所示）。与 Jacquard 的织布机一样，它是通过穿孔磁带控制的。用户在这种“程序磁带”上描述需要被机器执行的操作序列。每次执行新任务时，都把磁带送入机器中。这是由于尽管 Babbage 的机器能够记住中间结果，但它无法记住程序，它们总是需要某种外部支持。这被称为外部程序 (External Program) 机器。这种机器引入了存储器 (Memory)（被 Babbage 称为仓库 (Store)）和处理器 (Processor)（磨粉机 (Mill)）的概念。另一种创新与以前所做的相反，其中针是基于穿孔磁带上是否有孔存在而进行接合的，它们不会直接地接合输出设备 (Output Device)。在手摇风琴中，磁带中的每个孔都有一个音符与之关联；这被正式描述为输出逻辑上等同于 (Logically Equal) 输入。不过，在分析机中，我们已经可以说对程序和数据进行编码。

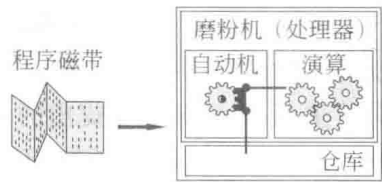


图 1.3 Babbage 的分析机

这种机器被分成三个截然不同的组件，它们具有不同的功能：自动机-计算器 (Automaton-Calculator) 部分、数据 (Data) 和程序 (Program)。

虽然穿孔磁带的每一行都包含本质上“合乎逻辑”的数据——孔是否存在，但是对于自动机和计算器单元则不能这样说，其中前者纯粹是机械式的，后者在 2 以 10 为基数的表

示法上工作。

在 1890 年美国的人口普查期间,必须自动处理数据的思想深入人心,这次人口普查包含 6200 万人。它是一次招标的主题,合同则被 Herman Hollerith(1860—1929)获得。Hollerith 建议使用一种已经被某些铁路公司使用的穿孔卡片系统(如图 1.4 所示)。像传说的那样,卡片的大小是 7.375 英寸×3.25 英寸,对应于当时 1 美元的纸钞的大小。由 Herman Hollerith 开办的 Tabulating Machine Company 公司最终在 1924 年变成了 International Business Machines (IBM) 公司。

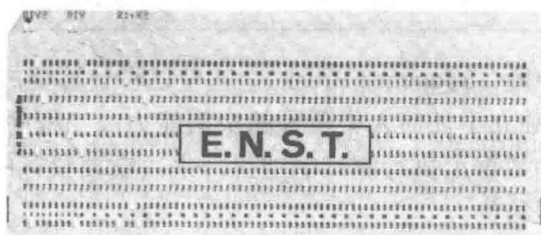


图 1.4 穿孔卡片：依据“Hollerith”代码对每个字符进行编码

1937 年,哈佛大学的 Howard Aiken 给 IBM 的建议是:通过用于穿孔卡片机器的机械和电子设备构建巨型计算器。该机器于 1943 年完成,重达 10 000 磅,配备有能够记住 72 个数字的累加器,并且可以在 6s 内计算两个 23 位的数字的乘积。通过编码到穿孔纸带(如图 1.5 所示)上的指令控制它。

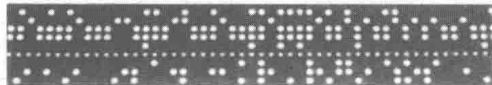


图 1.5 穿孔纸带

尽管获得了 Babbage 的知识,这台机器仍然缺乏处理条件指令的能力。不过,与 Babbage 的分析机相比,它具有另外两个特性:用于控制操作序列和寄存器的时钟,以及一种用于记录数据的临时存储器。

另一个先驱是在第二次世界大战期间英国设计的 Robinson,用于解码德国军队在 Enigma 机器上创建的加密消息。

1.1.2 从外部程序到存储程序

20 世纪 40 年代,对自动化计算器的研究成为一个繁荣兴旺的领域,它在很大程度上是受英国的 A. Turing 以及美国的 H. Aiken、P. Eckert 和 J. Mauchly[MAU 79]所激励的,并且部分基于 J. V. Atanasoff (1995) 的工作(1937 年和 1942 年之间的 Automatic Electronic Digital Computer (AEDQ))。

最初构建的机器是电动机械式的,后来则依靠真空管技术。它们设计用于特定的过程,每次需要更改操作序列时,都不得不重新接线。它们仍然是在外部编程的机器。J. Von Neumann[VON 45, GOL 63]为现代计算器使用的体系结构(即 Von Neumann 体系结构)建立了基础。