



HZ BOOKS
华章教育

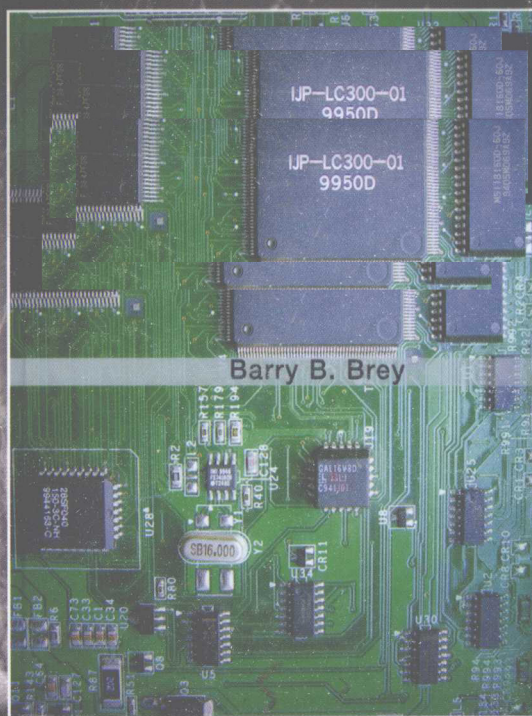
PEARSON

计 算 机 科 学 丛 书

原书第8版

Intel微处理器

(美) Barry B. Brey 著 金惠华 艾明晶 尚利宏 等译
德福瑞大学 北京航空航天大学



The Intel Microprocessors

8086/8088, 80186/80188, 80286, 80386, 80486, Pentium,
Pentium Pro Processor, Pentium II, Pentium III, Pentium 4,
and Core2 with 64-bit Extensions

Architecture, Programming, and Interfacing
Eighth Edition



机械工业出版社
China Machine Press

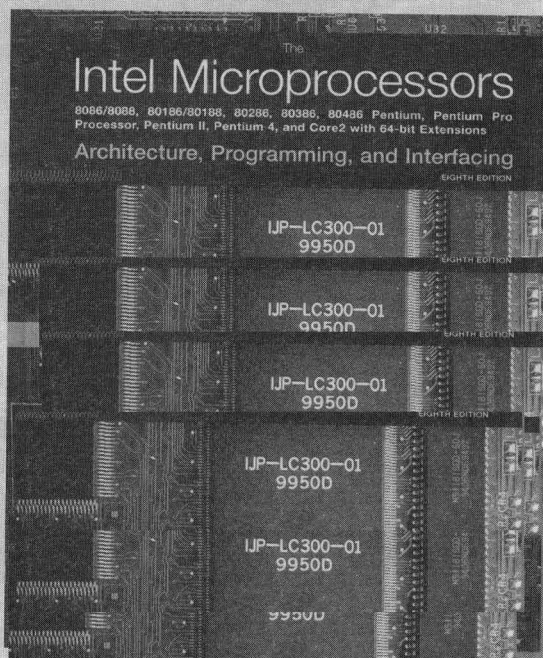
计 算 机 科 学 丛 书

原书第8版

Intel微处理器

(美) Barry B. Brey 著
德福瑞大学

金惠华 艾明品 尚利宏 等译
北京航空航天大学



The Intel Microprocessors

8086/8088, 80186/80188, 80286, 80386, 80486, Pentium,
Pentium Pro Processor, Pentium II, Pentium III,
Pentium 4, and Core2 with 64-bit Extensions
Architecture, Programming, and Interfacing
Eighth Edition



机械工业出版社
China Machine Press

本书重点讲解 Intel 系列微处理器 (8086/8088、80186/80188、80286、80386、80486、Pentium、Pentium Pro Processor、Pentium II、Pentium III、Pentium 4 和 Core2) 的体系结构、程序设计和接口通信技术, 并通过微型计算机原理把三者有机地整合在一起。本书以 Intel 系列微处理器为背景, 以 DOS、Windows 和 Visual C/C++ 为编程环境, 通过示例为读者深入揭示了微型计算机工作原理和最新的技术进步。许多示例都可以作为开发类似应用的样板或原型, 用以指引开发新的应用。

本书适合作为高等院校计算机、电子通信和自动控制等专业教材, 也可供工程技术人员参考。

Simplified Chinese edition copyright © 2010 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *The Intel Microprocessors: 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium, Pentium Pro Processor, Pentium II, Pentium III, Pentium 4, and Core2 with 64-bit Extensions: Architecture, Programming, and Interfacing, Eighth Edition* (ISBN 978-0-13-502645-8) by Barry B. Brey, Copyright © 2009, 2006, 2003, 2000, 1997, 1994, 1991, 1987.

All rights reserved.

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

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。

封底无防伪标均为盗版

版权所有, 侵权必究

本书法律顾问 北京市展达律师事务所

本书版权登记号: 图字: 01-2009-6760

图书在版编目 (CIP) 数据

Intel 微处理器 (原书第 8 版) / (美) 布雷 (Brey, B. B.) 著; 金惠华等译. —北京: 机械工业出版社, 2010. 6

(计算机科学丛书)

书名原文: *The Intel Microprocessors: 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium, Pentium Pro Processor, Pentium II, Pentium III, Pentium 4, and Core2 with 64-bit Extensions: Architecture, Programming, and Interfacing, Eighth Edition*

ISBN 978-7-111-30485-2

I. I… II. ①布… ②金… III. 微处理器, Intel 系列 IV. TP332

中国版本图书馆 CIP 数据核字 (2010) 第 074760 号

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: 秦 健

北京瑞德印刷有限公司印刷

2010 年 6 月第 2 版第 1 次印刷

184mm × 260mm · 44.25 印张

标准书号: ISBN 978-7-111-30485-2

定价: 89.00 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

购书热线: (010) 68326294; 88379649; 68995259

投稿热线: (010) 88379604

读者信箱: hzjsj@hzbook.com

出版者的话

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，不仅肇划了研究的范畴，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, Brain 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

译者序

这本讲述 Intel 系列微处理器程序设计和接口技术的参考书已经是第 8 版了。随着技术的发展和进步，作者多次改编增补删减，与时俱进，吐故纳新，在选材、知识点配置和编程环境选择方面充分体现了先进性和实用性。与前几版相比，第 8 版主要增加了讲解在 DOS 和 Windows 环境下如何使用可自由下载的 Visual C++ Express 编写 C/C++ 与汇编语言的接口，说明了在微处理器和微处理器接口方面出现的新成果。

本书的特点是：

- 取材涵盖微机原理、汇编语言和接口通信技术的教学要求和知识点，各部分有机整合，适合国内教学要求。
- 以 Intel 系列微处理器为背景，以 DOS、Windows 和 Visual C/C++ 为编程环境，通过示例为读者深入揭示了微机工作原理和最新技术进步。许多示例都可以作为读者开发类似应用的样板或原型，指引读者开发新的应用。
- 每章开头提示本章学习目的，末尾概要总结知识要点，最后有大量习题检验学习成果。编排符合学习规律，适合读者自学。

这是一本非常实用的教材，有助于读者彻底掌握 Intel 系列微处理器程序设计和接口技术，灵活自如地使用微机的各种资源，解决学习和开发工作中的实际问题。

本书由金惠华译第 1～6 章，艾明晶译第 9～15 章，尚利宏译第 16～19 章，高洁译第 7 章，郝广奇译第 8 章，李雅倩译附录，崔代锐、尚利荣、邓媛、刘云峰、徐其志参与了部分章节初译、示例习题核对及文稿录入。全书由金惠华统稿审校。译稿对原书中的笔误和疏漏进行了更正。由于译审者水平有限，译文中难免有不妥之处，敬请读者批评指正。

译者

2010 年 4 月

前 言

这本非常实用的参考书写给那些需要彻底掌握 Intel 系列微处理器程序设计和接口技术的大学生们。如今，任何在计算机应用领域里学习或工作的人都必须懂得汇编语言程序设计、一种 C 语言和接口技术，因为 Intel 系列微处理器已经在电子、通信、控制系统，特别是台式计算机系统等许多方面都得到了广泛而且有时独一无二的應用。第 8 版主要增加了讲解在 DOS 和 Windows 环境下如何使用可以从 Microsoft 自由下载的 Visual C++ Express 编写 C/C++ 与汇编语言的接口。很多应用程序包含 Visual C++ 作为用内嵌汇编程序学习汇编语言的基础。更新部分详细说明了在微处理器和微处理器接口方面出现的新成果。

组织结构和取材范围

为了培养综合的学习方法，每章开头都简明叙述了本章的目标。各章都包含了大量程序设计应用和实例，以阐明主题。每章末尾的数条小结对于指导学习事半功倍，并总结了前面讲解过的内容。习题部分则是对所学知识的进一步强化，并提供了实践机会。

本书大量地使用微软宏汇编（Microsoft Macro Assembler）程序和 Visual C++ 环境中的内嵌汇编程序作实例，为学习编写 Intel 系列微处理器的程序提供了机会。有关程序设计环境的操作，包括链接器、库、宏、DOS 功能调用、BIOS 功能调用和 Visual C/C++ 程序开发等。对于各种版本 Visual C++ 在 16 位和 32 位两种编程环境下的内嵌汇编器（C/C++）都做了详细说明。本书是用 Visual C++ Express 2005 或 2008 作为开发环境写的，但也可以几乎不作更改地使用任何版本的 Visual Studio。

本书还详尽说明了系列中每种微处理器、存储系统和各种 I/O 系统（包括磁盘存储器、ADC 和 DAC、16550 UART、PIA、定时器、键盘/显示控制器、算术协处理器和视频显示系统），并讨论了 PC 机的各种总线（AGP、ISA、PCI、PCI Express、USB、串口和并口）。通过这些系统，可以学习到实用的微处理器接口技术。

学习方法

由于 Intel 系列微处理器各不相同，本书开头集中讨论实模式下的程序设计，它与 Intel 系列所有型号微处理器兼容。针对这些系列成员的指令，比较 8086/8088 微处理器和 80386、80486、Pentium、Pentium Pro、Pentium II、Pentium III 及 Pentium 4 的异同，会发现所有这些微处理器非常相似，因此一旦懂得了基本类型的 8086/8088，就可以较容易地学习更高级的版本及其指令。注意，8086/8088 及随后的升级产品 80186/80188 和 80386EX 嵌入式微处理器仍然用于嵌入式系统中。

本书还讲解了算术协处理器、MMX 扩展和 SIMD 扩展的程序设计和操作，它们在系统中提供浮点计算的能力，这在控制系统、视频图像和计算机辅助设计（CAD）等应用领域是很重要的。算术协处理器允许程序完成复杂的算术运算，而用普通微处理器编程方法是难以胜任的。MMX 和 SIMD 指令可以使整数或者浮点数并行高速操作。

本书描述了 8086～80486 和所有 Pentium 微处理器的引脚及功能。在接口技术部分，首先讨论用于 8086/8088 的一些通用外围接口部件。说明基本部件后，再重点研究更先进的 80186/80188、80386、80486、Pentium 到 Pentium 4 微处理器。对 80286 的叙述很少，因为它与 8086 和 80386 很相似。我们将重点放在尽可能详细地讲述 80386、80486 和各种 Pentium 版本的微处理器上。

通过首先考虑各种先进微处理器的操作和程序设计，进而学习所有系列成员的接口技术，能够提供 Intel 系列微处理器的工作和实用背景。读者完成本书的学习后将能够：

1) 开发软件, 控制微处理器应用接口。通常, 开发出的软件应能用于所有型号的微处理器, 包括基于 DOS 的应用和基于 Windows 的应用。主要强调在 Windows 环境下开发内嵌汇编和 C++ 混合语言程序。

2) 使用 MFC 控件处理程序和函数调用编写汇编语言和 C++ 程序, 控制键盘、视频显示系统及磁盘存储器。

3) 使用宏序列、过程、条件汇编、流程控制汇编指令开发软件, 并链接到一个 Visual C++ 程序中。

4) 使用查找表和算法开发代码变换软件。

5) 对算术协处理器编程, 求解复杂的方程式。

6) 开发 MMX 和 SIMD 扩展软件。

7) 解释 Intel 系列的各种处理器的区别, 明确每一型号的特性。

8) 描述并使用微处理器的实模式和保护模式操作。

9) 设计存储器、I/O 系统到微处理器的接口。

10) 对 Intel 系列中各微处理器及其软件和硬件接口进行详细且全面的比较。

11) 解释嵌入式应用中实时操作系统的功能。

12) 解释磁盘及视频系统的操作。

13) 建立小型系统与 PC 系统的 ISA、PCI、串口、并口和 USB 总线之间的接口。

内容概述

第 1 章以基于微处理器的计算机系统为重点, 介绍了 Intel 微处理器系列, 包括微处理器的历史、操作和基于微处理器系统中存储数据的方法, 还包括数制及其变换。第 2 章介绍了微处理器程序设计模型和系统结构, 解释了实模式和保护模式的工作原理。

当我们了解了基本的计算机后, 第 3 章到第 6 章讲解了 Intel 微处理器系列每条指令的功能, 还提供了简单的应用程序来说明这些指令的操作, 使读者建立程序设计的基本概念。

第 7 章介绍 Visual C/C++ Express 如何与内嵌汇编程序及单独的汇编语言程序设计模块一起使用, 并说明如何配置一个简单的带汇编应用程序的 Visual C/C++ Express 程序。

有了程序设计基础之后, 第 8 章提供了一些使用带内嵌汇编程序的 Visual C++ Express 编写的应用程序, 这些应用程序包括通过消息处理函数在 Windows 环境下使用键盘和鼠标进行程序设计。把磁盘文件解释成 File 类, 就像键盘和视频显示器一样通过 Windows 在 PC 机上操作。这一章提供了在 Windows 环境下几乎可在 PC 机系统上开发任何程序的工具。

第 9 章介绍了 8086/8088 系列, 作为学习后面章节中基本存储器和 I/O 接口的基础, 本章还解释了系统缓冲和系统定时。

第 10 章解释存储器接口, 包括使用集成译码器的接口和用 VHDL 的可编程逻辑器件的接口, 提供了 8 位、16 位、32 位和 64 位存储系统, 因而 8086 ~ 80486 和 Pentium ~ Pentium 4 微处理器与存储器之间可以有接口。

第 11 章详细讨论了 I/O 接口技术, 包括 PIA、定时器、16550 UART 和 ADC/DAC。本章还说明了直流电机和步进电机的接口。

在理解了这些基本 I/O 部件及它们与微处理器的接口后, 第 12 章和第 13 章提供了一些高级 I/O 技术, 包括中断和直接存储器存取 (DMA) 及其应用 (打印机接口、实时时钟、磁盘存储器和视频显示系统)。

第 14 章详细叙述了 8087 ~ Pentium 4 系列算术协处理器的操作和程序设计技术, 以及 MMX 和 SIMD 指令。今天, 几乎没有不利用协处理器就能高效运行的应用程序。记住, 自从 80486 以后, 所有 Intel 微处理器都有了协处理器; 自 Pentium 后都有一个 MMX 部件; 自 Pentium II 后都有一个 SIMD 部件。

第 15 章阐明了如何通过并口、串口、ISA 和 PCI 总线使小型系统与 PC 机接口。

第 16 章和第 17 章涵盖 80186/80188 ~ 80486 这些先进的微处理器，探讨了它们与 8086/8088 微处理器的区别，以及它们的增强功能和特性，讲述了用于 80386 和 80486 微处理器的高速缓冲存储器、交叉存储和猝发存储。第 16 章还包括实时操作系统（RTOS），第 17 章还讨论了内存管理和内存分页技术。

第 18 章详述了 Pentium 和 Pentium Pro 微处理器，这些微处理器也基于最初的 8086/8088 微处理器。

第 19 章介绍了 Pentium II、Pentium III、Pentium 4 和 Core2 微处理器，包括一些新特性、封装类型和加到原指令系统中的指令集。

附录使本书更加充实。附录 A 列出了全部 DOS INT 21H 功能调用，还详细说明了汇编程序和 Windows Visual C++ 接口的使用。附录 B 给出所有 8086 ~ Pentium 4 和 Core2 指令的完整列表，包括许多指令示例和十六进制机器编码，以及时钟定时信息。附录 C 简要列出了改变标志位的所有指令。附录 D 提供了本书编号为偶数的习题的答案。

致谢

非常感谢下列专家的反馈意见：Brigham Young 大学的 James K. Archibald 和 Broome 社区学院的 William H. Murray III。

联络方式

我们可以通过 Internet 保持联络。我的网站包含本人全部教科书的信息和许多到 PC 机、微处理器、硬件和软件的重要链接。此外，还可以从中获得每周一次详述 PC 机许多方面的讲座，许多话题给出了本书未涉及的特别有趣的“技术环节”。如果你需要任何帮助，请在 bbrey@ee.net 上与我联系，我会在 24 个小时以内回答所有我的电子邮件。

我的网站：<http://members.ee.net/brey>

目 录

出版者的话
译者序
前言

第1章 微处理器和计算机导论	1
1.1 历史背景	1
1.1.1 机械时代	1
1.1.2 电子时代	2
1.1.3 程序设计的进步	3
1.1.4 微处理器时代	4
1.1.5 现代微处理器	5
1.2 基于微处理器的PC系统	12
1.2.1 存储器和I/O系统	12
1.2.2 微处理器	17
1.3 数制	20
1.3.1 数字	20
1.3.2 按位计数法	20
1.3.3 其他数制转换到十进制	21
1.3.4 十进制转换成其他进制	22
1.3.5 二进制编码的十六进制	23
1.3.6 补码	24
1.4 计算机数据格式	24
1.4.1 ASCII和Unicode数据	25
1.4.2 BCD数据	26
1.4.3 字节数据	27
1.4.4 字数据	28
1.4.5 双字数据	30
1.4.6 实数	31
1.5 小结	32
1.6 习题	33
第2章 微处理器及其体系结构	36
2.1 微处理器的内部体系结构	36
2.1.1 程序设计模型	36
2.1.2 多功能寄存器	38
2.2 实模式存储器寻址	41
2.2.1 段和偏移	41
2.2.2 默认段和偏移寄存器	42
2.2.3 段和偏移寻址机制允许重定位	43

2.3 保护模式存储器寻址简介	43
2.3.1 选择子和描述符	44
2.3.2 程序不可见寄存器	47
2.4 内存分页	48
2.4.1 分页寄存器	48
2.4.2 页目录和页表	50
2.5 平展模式内存	51
2.6 小结	52
2.7 习题	52
第3章 寻址方式	54
3.1 数据寻址方式	54
3.1.1 寄存器寻址	57
3.1.2 立即寻址	58
3.1.3 直接数据寻址	60
3.1.4 寄存器间接寻址	63
3.1.5 基址加变址寻址	65
3.1.6 寄存器相对寻址	67
3.1.7 相对基址加变址寻址	68
3.1.8 比例变址寻址	70
3.1.9 RIP相对寻址	71
3.1.10 数据结构	71
3.2 程序存储器寻址	72
3.2.1 直接程序存储器寻址	72
3.2.2 相对程序存储器寻址	72
3.2.3 间接程序存储器寻址	73
3.3 堆栈存储器寻址	74
3.4 小结	76
3.5 习题	78
第4章 数据传送指令	80
4.1 MOV回顾	80
4.1.1 机器语言	80
4.1.2 Pentium 4和Core2的64位模式	86
4.2 PUSH/POP指令	87
4.2.1 PUSH指令	87
4.2.2 POP指令	89
4.2.3 初始化堆栈	90
4.3 装入有效地址	91
4.3.1 LEA指令	91
4.3.2 LDS、LES、LFS、LGS和LSS	

指令	92	5.5.4 位扫描指令	136
4.4 数据串传送	94	5.6 串比较指令	137
4.4.1 方向标志	94	5.6.1 SCAS 指令	137
4.4.2 DI 和 SI	94	5.6.2 CMPS 指令	137
4.4.3 LODS 指令	94	5.7 小结	138
4.4.4 STOS 指令	95	5.8 习题	139
4.4.5 MOVS 指令	96	第 6 章 程序控制指令	141
4.4.6 INS 指令	98	6.1 转移指令	141
4.4.7 OUTS 指令	99	6.1.1 无条件转移指令	141
4.5 其他数据传送指令	99	6.1.2 条件转移和条件设置	145
4.5.1 XCHG 指令	99	6.1.3 LOOP 指令	148
4.5.2 LAHF 和 SAHF 指令	100	6.2 控制汇编语言程序的流程	149
4.5.3 XLAT 指令	100	6.2.1 WHILE 循环	151
4.5.4 IN 和 OUT 指令	101	6.2.2 REPEAT-UNTIL 循环	152
4.5.5 MOVSX 和 MOVZX 指令	102	6.3 过程	153
4.5.6 BSWAP 指令	103	6.3.1 CALL 指令	154
4.5.7 CMOV 指令	103	6.3.2 RET 指令	156
4.6 段超越前缀	103	6.4 中断概述	157
4.7 汇编程序详述	104	6.4.1 中断向量	157
4.7.1 伪指令	104	6.4.2 中断指令	158
4.7.2 存储器组织	108	6.4.3 中断控制	159
4.7.3 程序举例	110	6.4.4 PC 机的中断	159
4.8 小结	112	6.4.5 64 位模式中断	160
4.9 习题	113	6.5 机器控制及其他指令	160
第 5 章 算术和逻辑运算指令	115	6.5.1 控制进位标志位	160
5.1 加法、减法和比较指令	115	6.5.2 WAIT 指令	160
5.1.1 加法指令	115	6.5.3 HLT 指令	161
5.1.2 减法指令	119	6.5.4 NOP 指令	161
5.1.3 比较指令	121	6.5.5 LOCK 前缀	161
5.2 乘法和除法指令	122	6.5.6 ESC 指令	161
5.2.1 乘法指令	122	6.5.7 BOUND 指令	161
5.2.2 除法指令	124	6.5.8 ENTER 和 LEAVE 指令	161
5.3 BCD 码和 ASCII 码算术运算指令	127	6.6 小结	162
5.3.1 BCD 算术运算指令	127	6.7 习题	163
5.3.2 ASCII 算术运算指令	128	第 7 章 在 C/C++ 中使用汇编语言	165
5.4 基本逻辑运算指令	130	7.1 在 16 位 DOS 应用程序中使用汇编	
5.4.1 AND 指令	130	语言与 C/C++ 语言	165
5.4.2 OR 指令	131	7.1.1 基本规则和简单程序	166
5.4.3 XOR 指令	132	7.1.2 _asm 块中不能使用的 MASM	
5.4.4 测试和位测试指令	133	功能	167
5.4.5 NOT 指令和 NEG 指令	134	7.1.3 使用字符串	167
5.5 移位指令和循环移位指令	134	7.1.4 使用数据结构	169
5.5.1 移位指令	134	7.1.5 混合语言编程的例子	171
5.5.2 双精度移位指令	135	7.2 在 32 位应用程序中使用汇编语言	
5.5.3 循环移位指令	136	与 Visual C/C++ 语言	172

7.2.1 使用控制台 I/O 访问键盘和显示器的例子	173	9.2 时钟产生器 8284A	232
7.2.2 直接访问 I/O 端口	174	9.2.1 8284A 时钟产生器	232
7.2.3 开发 Windows 的 Visual C++ 应用程序	174	9.2.2 8284A 的操作	233
7.3 汇编和 C++ 混合目标码	180	9.3 总线缓冲及锁存	233
7.3.1 用 Visual C++ 链接汇编语言	181	9.3.1 多路分离总线	234
7.3.2 在 C/C++ 程序中添加新的汇编语言指令	184	9.3.2 缓冲系统	236
7.4 小结	185	9.4 总线时序	237
7.5 习题	185	9.4.1 基本的总线操作	237
第 8 章 微处理器程序设计	187	9.4.2 一般的时序	238
8.1 模块化程序设计	187	9.4.3 读时序	238
8.1.1 汇编程序和链接程序	187	9.4.4 写时序	241
8.1.2 PUBLIC 和 EXTRN	189	9.5 就绪和等待状态	241
8.1.3 库	190	9.5.1 READY 输入	241
8.1.4 宏	193	9.5.2 RDY 和 8284A	242
8.2 使用键盘和视频显示器	195	9.6 最小模式与最大模式	244
8.2.1 读取键盘	195	9.6.1 最小模式操作	244
8.2.2 使用视频显示器	198	9.6.2 最大模式操作	244
8.2.3 在程序中使用定时器	201	9.6.3 8288 总线控制器	244
8.2.4 鼠标	202	9.7 小结	246
8.3 数据转换	204	9.8 习题	246
8.3.1 二进制转换为 ASCII 码	204	第 10 章 存储器接口	248
8.3.2 ASCII 码转换为二进制	205	10.1 存储器器件	248
8.3.3 显示和读入十六进制数	206	10.1.1 存储器引脚	248
8.3.4 使用查找表实现数据转换	208	10.1.2 ROM 存储器	249
8.3.5 使用查找表的示例程序	209	10.1.3 静态 RAM (SRAM) 器件	251
8.4 磁盘文件	210	10.1.4 动态 RAM (DRAM) 存储器	254
8.4.1 磁盘的组织	211	10.2 地址译码	258
8.4.2 文件名	212	10.2.1 为什么要进行存储器译码	258
8.4.3 顺序存取文件	212	10.2.2 简单的与非门译码器	258
8.4.4 随机存取文件	220	10.2.3 3-8 线译码器 (74LS138)	259
8.5 程序举例	222	10.2.4 双 2-4 线译码器 (74LS139)	261
8.5.1 时间/日期显示程序	222	10.2.5 PLD 可编程译码器	262
8.5.2 数字排序程序	223	10.3 8088 和 80188 (8 位) 存储器接口	265
8.5.3 数据加密	225	10.3.1 基本的 8088/80188 存储器接口	265
8.6 小结	226	10.3.2 与快闪存储器接口	268
8.7 习题	227	10.3.3 错误校正	270
第 9 章 8086/8088 硬件特性	228	10.4 8086、80186、80286 和 80386SX (16 位) 存储器接口	271
9.1 引脚和引脚功能	228	10.5 80386DX 和 80486 (32 位) 存储器接口	278
9.1.1 引脚	228	10.5.1 存储体	278
9.1.2 电源要求	229	10.5.2 32 位存储器接口	279
9.1.3 直流特性	229	10.6 Pentium ~ Core2 (64 位) 存储器接口	281
9.1.4 引脚定义	229		

10.7 DRAM	284	的实例	344
10.7.1 DRAM 回顾	284	11.7 小结	345
10.7.2 EDO 存储器	286	11.8 习题	346
10.7.3 SDRAM	286	第 12 章 中断	348
10.7.4 DDR	286	12.1 基本中断处理	348
10.7.5 DRAM 控制器	287	12.1.1 中断的目的	348
10.8 小结	287	12.1.2 中断	349
10.9 习题	288	12.1.3 中断指令: BOUND、INTO、 INT、INT 3 和 IRET	351
第 11 章 基本 I/O 接口	289	12.1.4 实模式中断操作	351
11.1 I/O 接口概述	289	12.1.5 保护模式中断操作	352
11.1.1 I/O 指令	289	12.1.6 中断标志位	352
11.1.2 独立编址 I/O 与存储器 映像 I/O	290	12.1.7 将一个中断向量存入向量表	353
11.1.3 PC 机 I/O 映像	291	12.2 硬件中断	354
11.1.4 基本输入输出接口	291	12.2.1 INTR 和 $\overline{INTA}$	355
11.1.5 握手	293	12.2.2 82C55 键盘中断	358
11.1.6 关于接口电路的注释	294	12.3 扩展中断结构	360
11.2 I/O 端口地址译码	296	12.3.1 使用 74ALS244 扩展	360
11.2.1 译码 8 位 I/O 地址	296	12.3.2 菊花链中断	361
11.2.2 译码 16 位 I/O 地址	297	12.4 8259A 可编程中断控制器	362
11.2.3 8 位与 16 位 I/O 端口	298	12.4.1 8259A 概述	362
11.2.4 32 位 I/O 端口	300	12.4.2 连接单个 8259A	363
11.3 可编程外围设备接口	303	12.4.3 级联多个 8259A	364
11.3.1 82C55 基本描述	303	12.4.4 8259A 编程	364
11.3.2 82C55 编程	304	12.4.5 8259A 编程实例	368
11.3.3 方式 0 操作	305	12.5 中断实例	375
11.3.4 与 82C55 接口的 LCD 显示器	309	12.5.1 实时时钟	375
11.3.5 方式 1 选通输入	319	12.5.2 中断处理键盘	377
11.3.6 方式 1 选通输出	321	12.6 小结	379
11.3.7 方式 2 双向操作	322	12.7 习题	379
11.3.8 82C55 方式小结	324	第 13 章 直接存储器存取及 DMA 控制 I/O	381
11.3.9 串行 EEPROM 接口	325	13.1 基本 DMA 操作	381
11.4 8254 可编程间隔定时器	326	13.2 8237 DMA 控制器	382
11.4.1 8254 功能描述	326	13.2.1 软件命令	386
11.4.2 8254 编程	327	13.2.2 地址寄存器和计数寄存器 编程	386
11.4.3 直流电机速度与方向控制	331	13.2.3 8237 与 80X86 微处理器相连	387
11.5 16550 可编程通信接口	334	13.2.4 用 8237 进行存储器到存储器 传输	387
11.5.1 异步串行数据	335	13.2.5 DMA 处理的打印机接口	392
11.5.2 16550 功能描述	335	13.3 共享总线操作	394
11.5.3 16550 编程	336	13.3.1 定义的总线类型	395
11.6 模/数转换器 (ADC) 与数/模 转换器 (DAC)	340	13.3.2 总线仲裁器	395
11.6.1 DAC0830 数/模转换器	341	13.4 磁盘存储系统	400
11.6.2 ADC080X 模/数转换器	342		
11.6.3 使用 ADC0804 和 DAC0830			

13.4.1 软盘存储器	400	15.1.2 8 位 ISA 总线输出接口	461
13.4.2 笔式驱动器	403	15.1.3 8 位 ISA 总线输入接口	466
13.4.3 硬盘存储器	403	15.1.4 16 位 ISA 总线	468
13.4.4 光盘存储器	405	15.2 外围部件互连 (PCI) 总线	468
13.5 视频显示器	406	15.2.1 PCI 总线的引脚图	469
13.5.1 视频信号	407	15.2.2 PCI 总线的地址/数据线	469
13.5.2 TTL RGB 显示器	407	15.2.3 配置空间	470
13.5.3 模拟 RGB 显示器	408	15.2.4 PCI 总线的 BIOS	472
13.6 小结	411	15.2.5 PCI 接口	474
13.7 习题	412	15.2.6 PCI Express 总线	474
第 14 章 算术协处理器、MMX 和 SIMD 技术	413	15.3 并行打印机接口 (LPT)	475
14.1 算术协处理器的数据格式	413	15.3.1 端口介绍	475
14.1.1 带符号的整数	413	15.3.2 使用并行端口而不需要 ECP 支持	477
14.1.2 二进制编码的十进制 (BCD) ..	414	15.4 串行 COM 端口	477
14.1.3 浮点数	414	15.5 通用串行总线 (USB)	480
14.2 80X87 的结构	416	15.5.1 连接器	480
14.3 指令系统	421	15.5.2 USB 数据	480
14.3.1 数据传送指令	421	15.5.3 USB 命令	481
14.3.2 算术运算指令	422	15.5.4 USB 总线节点	482
14.3.3 比较指令	423	15.5.5 USBN9604/3 编程	482
14.3.4 超越运算指令	424	15.6 加速图形端口 (AGP)	485
14.3.5 常数操作指令	424	15.7 小结	485
14.3.6 协处理器控制指令	424	15.8 习题	485
14.3.7 协处理器指令	426	第 16 章 80186、80188 及 80286 微处理器	487
14.4 算术协处理器编程	438	16.1 80186/80188 的结构	487
14.4.1 计算圆的面积	438	16.1.1 80186/80188 的型号	487
14.4.2 求谐振频率	439	16.1.2 80186 基本结构框图	488
14.4.3 使用一元二次方程求根	440	16.1.3 80186/80188 基本特征	488
14.4.4 使用内存数组存储结果	441	16.1.4 引脚	490
14.4.5 将单精度浮点数转换为字符串 ..	442	16.1.5 直流工作特性	492
14.5 MMX 技术简介	443	16.1.6 80186/80188 时序	492
14.5.1 数据类型	443	16.2 80186/80188 增强功能编程	495
14.5.2 指令系统	444	16.2.1 外设控制块 (PCB)	495
14.6 SSE 技术概述	452	16.2.2 80186/80188 的中断	495
14.6.1 浮点数	453	16.2.3 中断控制器	496
14.6.2 指令集	454	16.2.4 定时器	500
14.6.3 控制/状态寄存器	454	16.2.5 DMA 控制器	505
14.6.4 编程实例	455	16.2.6 片选单元	507
14.6.5 优化	458	16.3 80C188EB 接口举例	510
14.7 小结	458	16.4 实时操作系统 (RTOS)	516
14.8 习题	459	16.4.1 实时操作系统 (RTOS) 概述 ..	516
第 15 章 总线接口	461	16.4.2 实例系统	517
15.1 ISA 总线	461	16.4.3 线程系统	519
15.1.1 ISA 总线的发展	461		

16.5	80286 简介	523	18.3	Pentium 的存储管理	576
16.5.1	硬件特性	523	18.3.1	分页单元	576
16.5.2	新增指令	524	18.3.2	存储管理模式	576
16.5.3	虚拟存储机	525	18.4	Pentium 的新指令	577
16.6	小结	526	18.5	Pentium Pro 微处理器简介	581
16.7	习题	526	18.5.1	Pentium Pro 的内部结构	582
第 17 章	80386 和 80486 微处理器	528	18.5.2	引脚连接	583
17.1	80386 微处理器简介	528	18.5.3	存储系统	586
17.1.1	存储系统	530	18.5.4	输入/输出系统	587
17.1.2	输入/输出系统	536	18.5.5	系统时序	587
17.1.3	存储器和 I/O 控制信号	537	18.6	Pentium Pro 的特性	587
17.1.4	时序	537	18.7	小结	588
17.1.5	等待状态	538	18.8	习题	589
17.2	特定的 80386 寄存器	538	第 19 章	Pentium II、Pentium III、 Pentium 4 和 Core2 微处理器	590
17.2.1	控制寄存器	538	19.1	Pentium II 微处理器简介	590
17.2.2	调试和测试寄存器	540	19.1.1	存储系统	595
17.3	80386 存储管理	541	19.1.2	输入/输出系统	596
17.3.1	描述符和选择子	541	19.1.3	系统时序	596
17.3.2	描述符表	544	19.2	Pentium II 软件变化	597
17.3.3	任务状态段 (TSS)	545	19.2.1	CPUID 指令	597
17.4	向保护模式转换	547	19.2.2	SYSENTER 和 SYSEXIT 指令	597
17.5	虚拟 8086 模式	556	19.2.3	FXSAVE 和 FXRSTOR 指令	598
17.6	内存分页机制	557	19.3	Pentium III	598
17.6.1	页目录	557	19.3.1	芯片组	598
17.6.2	页表	557	19.3.2	总线	598
17.7	80486 微处理器简介	559	19.3.3	引脚	599
17.7.1	80486DX 和 80486SX 微处理器 的引脚	560	19.4	Pentium 4 和 Core2	600
17.7.2	80486 的基本结构	563	19.4.1	存储器接口	600
17.7.3	80486 的存储系统	564	19.4.2	寄存器组	601
17.8	小结	566	19.4.3	超线程技术	602
17.9	习题	566	19.4.4	多核技术	602
第 18 章	Pentium 和 Pentium Pro 微处理器	568	19.4.5	CPUID	602
18.1	Pentium 微处理器简介	568	19.4.6	特定模型寄存器	605
18.1.1	存储系统	571	19.4.7	性能监视寄存器	605
18.1.2	输入/输出系统	572	19.4.8	64 位扩展技术	606
18.1.3	系统时序	572	19.5	小结	607
18.1.4	分支预测逻辑	574	19.6	习题	607
18.1.5	高速缓存结构	574	附录 A	汇编程序、Visual C++ 和 DOS	608
18.1.6	超标量体系结构	574	附录 B	指令系统一览	614
18.2	Pentium 的特定寄存器	574	附录 C	标志位的变化	674
18.2.1	控制寄存器	574	附录 D	偶数号习题的答案	676
18.2.2	EFLAG 寄存器	575			
18.2.3	内置自检 (BIST)	575			

第 1 章 微处理器和计算机导论

引言

本章介绍 Intel 系列微处理器的概况，讨论计算机的发展历史和基于微处理器的计算机系统中微处理器的功能。并且介绍计算机领域中使用的术语，这样，当我们讨论微处理器和计算机时就可以理解计算机行话了。

方框图及其功能说明详述计算机系统的操作，框图中的块表示 PC 机的存储器和输入/输出的相互联系。本章详细说明数据如何在存储器中存储，以便开发软件时使用各类数据。数值型数据以整数、浮点数和二进制编码的十进制（BCD）形式存储；而字母型数据以 ASCII 码（American Standard Code for Information Interchange，美国标准信息交换码）和 Unicode 码的形式存储。

目的

读者学习完本章后将能够：

- 1) 使用适当的计算机术语交谈，例如位、字节、数据、实存储系统、保护模式存储系统、Windows、DOS、I/O 等。
- 2) 简洁地叙述计算机的历史，并且列出计算机系统能执行的应用程序。
- 3) 说明 80X86 和 Pentium ~ Pentium 4 系列各个成员的概况。
- 4) 画出计算机系统的方框图，并且说明每块的功能。
- 5) 叙述微处理器的功能，并且详述它的基本操作。
- 6) 定义 PC 中存储系统的内容。
- 7) 进行二进制、十进制和十六进制数据之间的转换。
- 8) 区分和表示数字及字母信息，如整数、浮点数、BCD 和 ASCII 数据。

1.1 历史背景

本节概述导致微处理器发展的历史事件，并且具体讲解功能强大且十分流行的 80X86[⊖]、Pentium、Pentium Pro、Pentium III、Pentium 4[⊖] 和 Core2 微处理器。尽管研究历史不是理解微处理器所必需的，但是它从历史的角度展示了计算机的快速发展。

1.1.1 机械时代

计算系统的思想并不是新的，远在现代电气科学和电子器件出现以前它就已经存在了。用机器计算的概念在公元前 500 年就有记载，那时的巴比伦人发明了算盘（abacus），这是第一个机械式计算器。算盘用串珠实现计算，古代的巴比伦神父用它管理他们的巨大粮仓。算盘直到今天还在使用，始终没有改进，到 1642 年，当时的数学家 Blaise Pascal 发明了由齿轮和转轮构成的计算器。每个齿轮有十个齿牙，当其中一个齿轮转动一圈时，第二个齿轮推进一个齿牙。这和汽车里程表的原理一样，是所有机械计算器的基础。顺便说一下，PASCAL 程序设计语言就是为了纪念 Blaise Pascal 在数学和机械计算器方面的开拓性工作而命名的。

第一个实际用于自动计算信息的轮式机械计算器可追溯到 19 世纪初，这是在人类发明灯泡和深入了解电之前。在这个计算机的萌芽时代，人们梦想有会用程序计算数据的机器——而不仅仅用计算器

⊖ 80X86 是 8086、8088、80186、80188、80286、80386、80486 以及 Pentium 系列的简写形式。

⊖ Pentium、Pentium Pro、Pentium II、Pentium III、Pentium 4 和 Core2 是 Intel 公司的注册商标。

计算几个数据。

1937年人们通过一些设计图和日记发现：机械式计算机的一位早期的开拓者是 Charles Babbage。在 Lovelace 伯爵夫人 Augusta Ada Byron 的帮助下，受大不列颠皇家天文协会委托，Babbage 于 1823 年研制可编程的计算机，这个机器要为皇家海军绘制导航表。他接受了挑战，并开始建造他称为分析机 (Analytical Engine) 的机器。这个机器就是由蒸汽驱动的机械式计算机，它存储 1000 个 20 位长的十进制数字和一个可变的程序，程序能修改机器功能以便执行各种计算任务。这个机器通过穿孔卡片输入，酷似 20 世纪五六十年代计算机使用的穿孔卡片。他可能借鉴了法国人 Joseph Jacquard 提出的用穿孔卡片的思想，后者在 1801 年就在他发明的现今称为“Jacquard 织布机”的编织机器中使用了穿孔卡片作为输入。Jacquard 织布机用穿孔卡片为其生产的布匹选择复杂的编织图案，人们称其为穿孔卡片编程织布机。

努力多年以后，Babbage 对他的梦想逐渐失去信心，因为他认识到那个时代的机械师不可能制造出完成工作所需要的机械零件。分析机需要 50 000 多个机械零件，无法以足够的精密度制造出来，因此无法使分析机可靠地工作。

1.1.2 电子时代

19 世纪出现了电动机 (由 Michael Faraday 构想)，并且出现了许多电动机驱动的计算机，这些都建立在 Blaise Pascal 的机械计算器基础上。这些电动的机械计算器一直作为通用办公设备使用，直到 20 世纪 70 年代初出现了由 Bomar 公司首先推出，叫 Bomar Brain 的小型手持电子计算器。Monroe 也是电子计算器的先驱者，但他的机器是台式的，相当于一台 4 功能收款机的大小。

1889 年 Herman Hollerith 研制了存储数据的穿孔卡片，如同 Babbage 一样，他也显然借鉴了 Jacquard 穿孔卡片的思想。他还开发了由一种新式电机驱动的机械式计算器，这个计算器可计算、分类和比较存储在穿孔卡片上的信息。用机器进行计算的想法引起了美国政府的兴趣，因此委托 Hollerith 用穿孔卡片系统存储 1890 年人口普查的资料并制成表格。

1896 年，Hollerith 组建了 Tabulating Machine Company 公司 (制表机械公司)，这个公司开发了用穿孔卡片制表的行式机器。经过数次兼并后，Tabulating Machine Company 成为 International Business Machines Corporation (国际商用机器公司)，现在称为 IBM。为了纪念 Herman Hollerith，我们通常将计算机系统中使用的穿孔卡片称为 Hollerith 卡，穿孔卡片使用的 12 位代码称为 Hollerith 码。

用电机驱动的机械式机器，一直主导着信息处理世界，直到 1941 年出现第一台电子计算机。作为工程师为柏林 Henschel 飞机公司工作的德国发明家 Konrad Zuse 发明了第一台现代计算机。在 1936 年 Zuse 构造了他的一个机械版的系统，并且随后在 1939 年构造了他的第一个电-机计算机系统，叫做 Z2。他的 Z3 计算机，如图 1-1 所示，第二次世界大战期间德国人用它设计飞机和导弹。Z3 是工作时钟为 5.33 Hz 的一台继电器逻辑机器 (比最新的几个 GHz 级的微处理器慢得太多了)。如果当时德国政府给予 Zuse 足够的资金，他很可能研制出功能更强的计算机系统。今天 Zuse 最终得到了一些迟到的称颂，对他在数字电子领域和 Z3 计算机系统的开创性工作表示敬意。

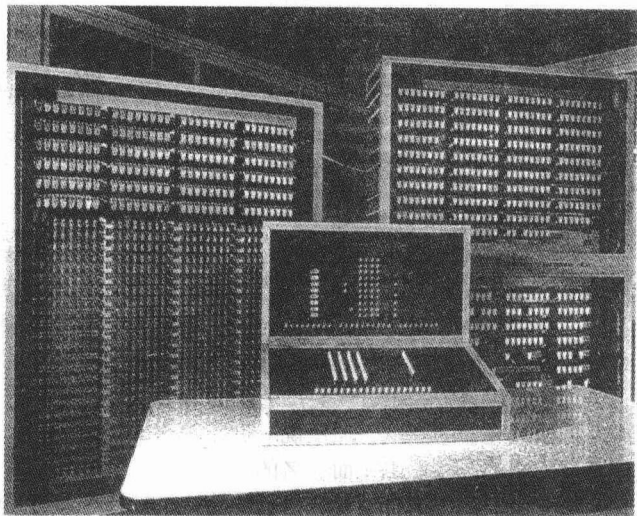


图 1-1 Konrad Zuse 研制的 Z3 计算机，时钟频率为 5.33 Hz
(照片由 Konrad 的儿子 Horst Zuse 提供)

最近发现（通过解密的英国军事文件）第一台真正的电子计算机于1943年安装运转，用于破译德国军事密码。这第一台使用了真空管的电子计算机系统是由 Alan Turing 发明的。Turing 称他的机器为巨人（Colossus），或许是因为机器的尺寸庞大。巨人的问题是，虽然它可以破译由英格玛机（Enigma machine）生成的德国军事密码，但是却不能解决其他问题。巨人不是可编程的，它是固定程序的计算机系统，今天通常称为专用计算机。

第一台通用可编程电子计算机系统于1946年由宾夕法尼亚大学研制成功。这是第一台现代计算机，称为 ENIAC（Electronic Numerical Integrator and Calculator，电子数字积分器和计算器）。ENIAC 是个庞大的机器，它使用了 17 000 多个真空管和超过 500 英里长的导线。这个庞然大物的重量超过 30 吨，而每秒只能执行约 10 万次运算。ENIAC 推动世界进入了电子计算机时代。ENIAC 采用重新连接线路的方法实现编程，这个过程需要许多工人花几天时间才能完成。工人们改变插接板上的电路连接，此操作方式很像早期的电话接线总机。ENIAC 的另一个问题是真空管器件的寿命低，需要经常维护。

随后的突破性进展是1947年12月23日由贝尔实验室的 John Bardeen, William Shockley 和 Walter Brattain 研制出了晶体管。其后，在1958年得克萨斯仪器公司的 Jack Kilby 发明了集成电路。集成电路导致20世纪60年代数字集成电路（RTL，即电阻-晶体管逻辑）的发展，以及1971年 Intel 公司第一种微处理器的诞生。当时 Intel 工程师 Federico Faggin, Ted Hoff 和 Stan Mazor 研制出了4004微处理器（美国专利号：No. 3,821,715），该微处理器启动了今天还在继续加速进行着的微处理器革命。

1.1.3 程序设计的进步

既然开发出了可编程序的机器，因此程序和程序设计语言也开始相继出现。如上所述，第一个可以编写程序的电子计算机系统是通过重新连接线路实现编程的。由于这在实际应用中很麻烦，因此在计算机系统发展的早期，产生了用于控制计算机的计算机语言。第一种这样的语言叫作机器语言，是由多个1和0组成的二进制代码，以指令组的形式存储在计算机系统中，被称为程序。这种方法比通过重新连接机器线路进行编程的方法有效，但是开发程序仍然非常耗费时间，因为完全要用数码来编程。数学家冯·诺依曼（John Von Neumann）首先开发了接受指令并且可将指令存储到存储器中的系统。为了纪念他，计算机常常称为冯·诺依曼机器（Von Neumann machine）。（回想一下，Babbage 比冯·诺依曼更早地提出了这一概念。）

20世纪50年代初，随着 UNIVAC 之类的计算机系统投入使用，汇编语言简化了以二进制代码为计算机输入指令的繁琐工作。汇编语言允许程序员用助记符代替二进制码，例如用 ADD 表示加法，代替二进制码 0100 0111。虽然汇编语言可以帮助进行程序设计，但是编程仍然很不容易，直到1957年 Grace Hopper 开发了称为 FLOWMATIC 的第一个高级程序设计语言。同年，IBM 为它的计算机系统开发了 FORTRAN（FORmula TRANslator，公式翻译器）。FORTRAN 语言允许程序员开发使用公式解决数学问题的程序，至今，一些科学家仍然使用 FORTRAN。比 FORTRAN 约晚一年出现了另一种类似的语言 ALGOL（ALGOrithmic Language，算法语言）。

第一个真正成功并广泛用于商业的程序设计语言是 COBOL（COmputer Business Oriented Language，面向商业计算机的语言）。尽管近几年 COBAL 的使用减少了，但是在许多大的商业系统中它仍然发挥着主要作用。另外一种一度很流行的商业语言是 RPG（Report Program Generator，报告程序生成器），它允许通过规范输入、输出和运算的格式进行程序设计。

在这些早期的程序设计语言之后，更多的语言相继出现了，比较普及的是 BASIC、Java、C#、C/C++、PASCAL 和 ADA。BASIC 和 PASCAL 被设计成为教学用语言，但其使用范围早已扩展到了许多计算机系统中。BASIC 语言可能是所有语言中最容易学习的，据估计80%的PC用户程序是用 BASIC 语言编写的。在十年前，BASIC 新版本 Visual BASIC 的出现使 Windows 环境中的程序设计更容易了。Visual BASIC 语言可能最终取代 C/C++ 和 PASCAL 作为一种科学语言，但这值得怀疑。比起 C#更贴近硬件，它更加表面；现实点可能替代 C/C++ 和包括 Java 的大多数其他语言，可能最终替换 BASIC。这当然仅是推测，只有将来才会看到哪种语言最终能成为霸主。

在科学界，C/C++ 偶尔还有 PASCAL 和 FORTRAN 通常用于控制程序。最近一个对嵌入式系统的