

国外计算机科学教材系列

# 80×86、奔腾机汇编语言程序设计

Programming the 80286, 80386, 80486, and  
Pentium-Based Personal Computer

Barry B. Brey 著

金惠华 曹庆华 李雅倩 译



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY



PRENTICE HALL 出版公司

## 内 容 提 要

全书共 14 章。前 3 章讲授了数制、微处理器和它的结构等基础知识;第 4~8 章描述了微处理器的基本程序设计模型、实模式和保护模式操作,以及 Intel 系统微处理器每条指令的功能;第 9~11 章给出了使用汇编程序的应用程序;第 12~14 章分别介绍了 TSR 的结构与操作,8087 至 Pentium 系列算术协处理器的操作和程序设计,以及高级程序设计等内容。

本书取材广泛、实用,涵盖了用汇编语言控制微处理器和个人计算机资源的实用技术。深入浅出,每章有小结、例子和习题。书后附录是汇编语言程序设计完整的工具手册。

本书可作为大专院校计算机系相关专业学生的教材或参考书,也可供计算机工程技术人员参考。

© 1996 by Prentice-Hall, Inc.

本书中文简体版由电子工业出版社和美国 Prentice Hall 出版公司合作出版。未经许可,不得以任何手段和形式复制或抄袭本书内容。版权所有,侵权必究。

丛 书 名: 国外计算机科学教材系列

原 书 名: Programming the 80286, 80386, 80486, and Pentium-Based Personal Computer

书 名: **80x86、奔腾机汇编语言程序设计**

著 者: Barry B. Brey 著

译 校 者: 金惠华 曹庆华 李雅倩

责任编辑: 范官清

印 刷 者: 天竺颖华印刷厂印刷

出版发行: 电子工业出版社出版、发行 URL: <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036 发行部电话 68214070

经 销: 各地新华书店经销

开 本: 787×1092 1/16 印张:47.25 字数:1150 千字

版 次: 1998 年 8 月第 1 版 1998 年 8 月第 1 次印刷

印 数: 7000 册

书 号: ISBN 7-5053-4602-4  
TP·2183

定 价: 68.00 元

著作权合同登记号 图字:01-97-1861

凡购买电子工业出版社的图书,如有缺页、倒页、脱页者,本社发行部负责调换

**版权所有·翻印必究**

## 出版说明

计算机科学的迅速发展是 20 世纪科学发展史上最伟大的事件之一。从 1946 年第一台笨重而体积庞大的计算机的发明至今,仅仅半个多世纪,计算机已经变得小巧无比却又能力非凡。它的应用已经渗透到了社会的各个方面,成为当今所谓的信息社会的最显著的特征。

处于世纪之交科技进步的大潮中,我国正在加强计算机科学的高等教育,着眼于为下一世纪培养高素质的计算机人才,以适应信息社会加速度发展的需要。当前,全国各类高等院校已经或计划在各专业基础课程规划中增加计算机科学的课程内容,而作为与计算机科学密切相关的计算机、通信、信息等专业,更是在酝酿着教学的全面革新,以期规划出一整套面向 21 世纪的、具有中国高校计算机教育特色的课程计划和教材体系。值此,我们不妨借鉴并引进国外具有先进性、实用性和权威性的大学计算机教材,洋为中用,以更好地服务于国内的高校教育。

美国 Prentice Hall 出版公司是享誉世界的高校教材出版商,自 1913 年公司成立以来,即致力于教育图书的出版。它所出版的计算机教材在美国为众多大学所采用,其中有不少是专业领域中的经典名著。许多蜚声世界的教授学者成为该公司的资深作者,如:道格拉斯·科默(Douglas Comer),安德鲁·坦尼伯姆(Andrew Tanenbaum),威廉·斯大林(William Stallings)……几十年来,他们的著作教育了一批批不同肤色的莘莘学子,使这些教材同时也成为全人类的共同财富。

为了保证本系列教材翻译出版的质量,电子工业出版社和 Prentice Hall 出版公司共同约请北京地区的清华大学、北京大学、北京航空航天大学,上海地区的上海交通大学、复旦大学,南京地区的南京大学、解放军通信工程学院等全国著名的高等院校的教学第一线的几十位教师参加翻译工作。这中间有正在讲授同类教材的年轻教师和博士,有积累了几十年教学经验的教授和博士生导师,还有我国著名的计算机科学家。他们的辛勤劳动保证了本系列丛书得以高质量地出版面世。

如此大规模地引进计算机科学系列教材,在我们还是第一次。除缺乏经验之外,还由于我们对计算机科学的发展,对中国高校计算机教育特点认识的不足,致使在选题确定、翻译、出版等工作中,肯定存在许多遗憾和不足之处,恳请广大师生和其他读者提出批评、建议。

电子工业出版社

URL: <http://www.phei.com.cn>

Prentice Hall 出版公司

URL: <http://www.prenhall.com>

## 译者的话

对于学习和使用计算机的人们来说,通晓汇编语言程序设计和个人计算机系统是十分重要的。作为教材,《80x86、奔腾机汇编语言程序设计》是专门写给那些在校学生和工程技术人员。它可以帮助读者了解个人计算机的体系结构,丰富程序设计方面的知识,并学会应用汇编语言对计算机进行二次开发。

本书的特点是:

- 1 取材实用、广泛,涵盖了用汇编语言控制微处理器和个人计算机资源的实用技术。
- 2 讲述深入浅出,适合于不同层次的读者。
- 3 每章有小结,附有大量例子和习题,非常适合于自学。
- 4 书后的附录是汇编语言程序设计完整的工具手册。

我们之所以把本书全部译出,目的是为了帮助广大读者能够解决学习和开发工作中的实际问题。感兴趣的读者会发现这是一本非常实用的参考教材。

本书初稿由金惠华(1~3章和附录)、李雅倩(4~8章)、曹庆华(9~14章)翻译,由金惠华对全书统稿,李雅倩对全书做了统校。原书中的一些笔误已作改正。

由于译者水平有限,错误之处在所难免,敬请读者批评指正。

1998年2月于北京

# 序 言

本书是为那些需要通晓 Intel 个人计算机结构和程序设计的各专业大学生们编写的。这些专业包括计算机科学和计算机工程,或者说是计算机工程技术。对于微处理器系列和个人计算机系统的程序设计及其他方面感兴趣的任何人,都会发现这是一本非常实用的参考指南。

今天,任何应用计算机在某一领域搞研究的人都必须熟悉汇编语言程序设计和个人计算机的原理。当使用高级语言,如 C/C++ 来开发软件时尤其如此。虽然 C/C++ 可以实现绝大部分机器语言能够实现的功能,汇编语言软件还是经常被用以改进软件和硬件控制系统的效率。因为机器的控制要由汇编语言来完成,所以理解汇编语言的内部细节非常重要。对于动画游戏和虚拟现实应用软件来讲,汇编语言是唯一的选择,因为它能够提供高效率的代码。

具有数制方面的基础知识对学习本书是很有帮助的。一旦没有这方面的知识也没有关系,本书第一章到第三章介绍了数制的基础知识。除非使用微软宏汇编 Microsoft MASM 5.10 版或更新的 6.x 版编程,学习本书无需更多的学科基础。

Intel 微处理器在电子、通信、控制系统及台式计算机等许多领域里,获得了广泛的应用和认可。个人计算机往往是系统开发的起点,目前计算机已经高度集成化,出现了像 80386EX 那样的单片 PC 机。

## 组织和范围

本书每章都以“目的”段开头,简洁明确地列出本章内容。随后是这一章的主体部分,这一部分提供了大量的应用程序,阐明本章的主题。在每一章末尾的“小结”部分,对本章的主要内容作了总结。最后是一些问题和练习,以便读者进一步熟悉本章出现的概念,训练自己的编程技能。

本书包括数百个使用微软宏汇编的程序设计实例,这为学习如何编写 Intel 系列微处理器和个人计算机的应用程序,提供了一个模仿的机会。程序设计环境的操作包括链接、库、宏、DOS 功能调用和 BIOS 功能调用等。同时也覆盖了个人计算机结构和编程等方面的内容。

## 方法

由于 Intel 系列各微处理器功能上各不相同,本书在前一部分集中讨论了实模式下的汇编语言程序设计,它与 Intel 系列所有型号微处理器以及基于这些处理器的所有个人计算机兼容。将 80286、80386、80486 和 Pentium 微处理器与 8086/8088 微处理器进行比较和对照可以发现,整个微处理器序列的相似之处很多,一旦把基本原理弄懂了,更先进的型号也是容易学习的。

本书还介绍了 CONFIG.SYS 文件和 AUTOEXEC.BAT 文件的知识,就像它们在 DOS

中应用并配置个人计算机那样。通过对 DOS 的实用程序和批处理程序的举例分析,将使你了解到配置程序的基础知识。本书还对 Windows 的配置及其配置文件 WIN.INI、SYSTEM.INI 作了分析。

另外,为了解微处理器程序设计和操作,本书也讨论了数字协处理器(8087/80287/80387/80486/7 和 Pentium)的程序设计和操作。数字协处理器在系统中提供可用浮点计算的能力,这在控制系统、视频图像和计算机辅助设计(CAD)等应用领域是很有意义的。数字协处理器允许程序去完成复杂的算术运算,如用一般微处理器编程方法,是难以胜任的。

本书介绍的先进程序设计技术,包括学习用 VCPI 和由 EMM386.EXE 提供的 EMS 访问扩充存储器,由 DPMI 和 HIMEM.SYS 提供的 XMS 功能访问扩展存储器的程序设计,以及由 DPMI 提供的保护模式的应用程序设计。本书还分析了设备驱动程序的结构,并对几个常见的设备驱动程序作了介绍。其他先进的程序设计技术还包括 TSR、热键和鼠标驱动程序等。

## 学习目标

1. 开发 8086/8088、80286、80386、80486 和 Pentium 等微处理器的应用界面控制软件。通常,所开发的软件能在所有的微处理器上工作,这里包括基于 DOS 的应用软件。
2. 使用 DOS 功能调用去控制键盘、视频显示系统和磁盘存储器的汇编语言程序。
3. 使用 DOS 内部和外部命令控制个人计算机系统,编写批处理程序。
4. 使用 BIOS 功能控制键盘、显示器和计算机系统其他部件。
5. 使用 INT33H 通过鼠标驱动器去调用鼠标。
6. 开发使用中断钩子和热键的应用程序,以便访问终止并驻留内存程序。
7. 设计数字协处理器的程序,以便解决复杂方程式。
8. 说明微处理器系列各成员间的差异和每个成员间的主要特点。
9. 说明和使用 80286、80386、80486 和 Pentium 微处理器的实模式和保护模式操作。
10. 使用 EMS、XMS、VCPI 及 DMPI 驱动程序实现保护模式操作、访问扩充存储器及扩展存储器。
11. 用 DOS 提供的 IOCTL 功能编写设备驱动程序。
12. 用 DOS 提供的 EXEC 功能去执行另外的程序。
13. 由 C/C++、PASCAL、BASIC 和 FORTRAN 那样的高级语言调用汇编语言程序。

## 内容概述

第一章到第三章介绍了 Intel 系列微处理器,重点是以微处理器为基础的个人计算机系统。开始的几章内容介绍了微处理器的历史、操作和在微机系统中用于存放数据的方法;介绍了个人计算机的结构、它的存储器映像、I/O 映像、磁盘系统以及 DOS 内部和外部命令。在这一部分还讨论了 DOS 配置文件——CONFIG.SYS 和 AUTOEXEC.BAT,以及 Windows 配置文件——WIN.INI 和 SYSTEM.INI;研究了控制个人计算机系统的批处理程序的结构和开发。在学习了计算机基础知识以后,第四章到第八章的内容描述了微处理器的基本程序设计模型,实模式和保护模式操作,以及 Intel 系列微处理器每条指令的功能等内容。为说明指令,这一部分还给出了一些简单的应用程序,举例说明了指令操作,并介绍了

## 译者的话

对于学习和使用计算机的人们来说,通晓汇编语言程序设计和个人计算机系统是十分重要的。作为教材,《80x86、奔腾机汇编语言程序设计》是专门写给那些在校学生和工程技术人员的。它可以帮助读者了解个人计算机的体系结构,丰富程序设计方面的知识,并学会应用汇编语言对计算机进行二次开发。

本书的特点是:

- 1 取材实用、广泛,涵盖了用汇编语言控制微处理器和个人计算机资源的实用技术。
- 2 讲述深入浅出,适合于不同层次的读者。
- 3 每章有小结,附有大量例子和习题,非常适合于自学。
- 4 书后的附录是汇编语言程序设计完整的工具手册。

我们之所以把本书全部译出,目的是为了帮助广大读者能够解决学习和开发工作中的实际问题。感兴趣的读者会发现这是一本非常实用的参考教材。

本书初稿由金惠华(1~3章和附录)、李雅倩(4~8章)、曹庆华(9~14章)翻译,由金惠华对全书统稿,李雅倩对全书做了统校。原书中的一些笔误已作改正。

由于译者水平有限,错误之处在所难免,敬请读者批评指正。

1998年2月于北京

# 目 录

<b>第一章 微处理器和个人计算机入门</b> .....	(1)
1-1 历史回顾.....	(1)
1-2 基于微处理器的个人计算机 .....	(10)
1-3 数制 .....	(23)
1-4 计算机数据格式 .....	(30)
1-5 小结 .....	(38)
1-6 问题和练习 .....	(40)
<b>第二章 磁盘、DOS 和批处理程序</b> .....	(45)
2-1 磁盘及其组织 .....	(45)
2-2 DOS 内部命令 .....	(55)
2-3 DOS 外部命令 .....	(61)
2-4 CONFIG.SYS 文件.....	(69)
2-5 批处理程序(.BAT).....	(72)
2-6 小结 .....	(79)
2-7 问题和练习 .....	(80)
<b>第三章 Windows 和配置 Windows</b> .....	(83)
3-1 Windows 简介 .....	(83)
3-2 WIN.INI 文件 .....	(86)
3-3 SYSTEM.INI 文件 .....	(87)
3-4 其他.INI 文件 .....	(89)
3-5 改善 Windows 的性能 .....	(90)
3-6 DOS 应用程序 PIF 文件 .....	(91)
3-7 小结 .....	(92)
3-8 问题和练习 .....	(93)
<b>第四章 微处理器及其结构</b> .....	(95)
4-1 80386、80486 及 Pentium 的内部结构 .....	(95)
4-2 实模式存储器寻址.....	(100)
4-3 保护模式存储器寻址.....	(104)
4-4 存储器分页.....	(110)
4-5 小结.....	(112)



4-6 问题和练习·····	(115)
<b>第五章 寻址方式·····</b>	<b>(117)</b>
5-1 数据寻址方式·····	(117)
5-2 程序存储器寻址方式·····	(140)
5-3 堆栈存储器寻址·····	(142)
5-4 小结·····	(145)
5-5 问题与练习·····	(148)
<b>第六章 数据传送指令·····</b>	<b>(151)</b>
6-1 MOV 回顾·····	(151)
6-2 PUSH/POP·····	(160)
6-3 装入有效(Load-effective)地址·····	(165)
6-4 串行数据传送·····	(169)
6-5 杂项数据传送指令·····	(175)
6-6 段替代前缀·····	(180)
6-7 汇编程序·····	(180)
6-8 小结·····	(190)
6-9 问题与练习·····	(192)
<b>第七章 算术和逻辑运算·····</b>	<b>(195)</b>
7-1 加法,减法和比较·····	(195)
7-2 乘法和除法·····	(205)
7-3 BCD 码和 ASCII 码算术运算·····	(210)
7-4 基本的逻辑指令·····	(213)
7-5 移位和循环指令·····	(220)
7-6 串比较指令·····	(223)
7-7 小结·····	(225)
7-8 问题与练习·····	(227)
<b>第八章 程序控制指令·····</b>	<b>(231)</b>
8-1 转移类·····	(231)
8-2 过程·····	(242)
8-3 中断概述·····	(249)
8-4 机器控制及其他混杂的指令·····	(253)
8-5 小结·····	(256)
8-6 问题与练习·····	(259)

<b>第九章 键盘、显示器的 DOS 和 BIOS 功能调用</b> .....	(261)
9-1 模块化编程 .....	(261)
9-2 使用键盘和显示器 .....	(271)
9-3 数据变换 .....	(283)
9-4 小结 .....	(296)
9-5 问题与练习 .....	(297)
<b>第十章 条件汇编和视频显示</b> .....	(299)
10-1 条件汇编 .....	(299)
10-2 低分辨率图形显示器(VGA) .....	(316)
10-3 高分辨图形显示器(VGA) .....	(323)
10-4 小结 .....	(339)
10-5 问题与练习 .....	(240)
<b>第十一章 磁盘存储器功能和鼠标器</b> .....	(342)
11-1 顺序存取磁盘文件 .....	(342)
11-2 随机存取文件 .....	(351)
11-3 磁盘目录 .....	(354)
11-4 使用鼠标 .....	(358)
11-5 小结 .....	(366)
11-6 问题与练习 .....	(367)
<b>第十二章 中断钩子和内存驻留程序</b> .....	(369)
12-1 中断钩子 .....	(369)
12-2 热键 .....	(380)
12-3 可卸载的 TSR .....	(396)
12-4 小结 .....	(406)
12-5 问题与习题 .....	(407)
<b>第十三章 数值运算协处理器</b> .....	(409)
13-1 数值运算协处理器的数据格式 .....	(409)
13-2 80X87 协处理器结构 .....	(413)
13-3 指令系统 .....	(418)
13-4 使用数值运算协处理器进行编程 .....	(443)
13-5 小结 .....	(453)
13-6 问题与练习 .....	(454)

<b>第十四章 汇编语言高级编程技巧</b> .....	(457)
14-1 使用 DOS EXEC 系统调用执行程序 .....	(457)
14-2 内存管理驱动程序 .....	(463)
14-3 IOCTL 和设备驱动程序 .....	(482)
14-4 保护模式操作 .....	(513)
14-5 汇编语言与高级语言的接口 .....	(527)
14-6 小结 .....	(533)
14-7 问题与练习 .....	(534)
<b>附录 A 汇编语言、磁盘操作系统(DOS)、基本输入输出系统(BIOS)、 鼠标及内存管理</b> .....	(536)
<b>附录 B 指令系统小结</b> .....	(614)
<b>附录 C 标志位的变化</b> .....	(701)
<b>附录 D 双号问题与练习的标准答案</b> .....	(703)
<b>词汇表</b> .....	(727)

# 第一章 微处理器和个人计算机入门

## 引言

本章介绍了微处理器的概况,内容包括对计算机历史的回顾,还讨论了在以微处理器为基础的计算机系统中,即 PC 机中,微处理器的功能。同时也介绍了计算机领域中的一些术语和行话,这样当讨论微处理器和计算机时就可以使用和理解计算机行话了。

这一章使用方框图来详述计算机系统的操作。本章也介绍了个人计算机的存储器和输入/输出系统。

最后,为了要使每种数据都能用于软件开发,说明了各种数据在存储器中的存放方法。以数字表示的数据,不论浮点还是二进制编码的十进制(BCD)都像整数一样地存放,而字符数据都以 ASCII 码(American Standard Code for Information Interchange)存储。在本章还讨论了要在十进制数与任何其他进制数字(例如二进制、八进制及十六进制数字)之间转换的步骤。

## 目的

学完这一章你将能够:

1. 正确使用计算机术语,包括位、字节数据、实存储器系统、扩充存储器系统(EMS)、扩展存储器系统(XMS)、DOS、BIOS 和 I/O 等术语。
2. 简洁地描述计算机的历史,并且列出由计算机实现的应用。
3. 知道软件开发先驱者的名字,并且说明他们创建的程序。
4. 画出计算机系统的框图,并且说明部分的用途。
5. 叙述微处理器的功能,并且评述它的基本操作。
6. 了解个人计算机中存储器系统容量的定义。
7. 进行十进制、二进制和十六进制之间数据的转换。
8. 区分整数、浮点数和二进制代码表示的十进制数(BCD)和 ASCII 码之间的差别,并且描述合理的数字及字母信息。

## 1.1 历史回顾

第一节我们概述微处理发展历史上的重要事件,特别是与当前使用的功能异常强大的 80386、80486 和 Pentium 处理器相关的那段历史。尽管研究历史不是理解微处理器所必须的,但是它提供展现了计算机快速发展的历史画面。

## 机械式计算器时代

计算装置的思想并不是新东西,在现代电气的和电子的器件出现以前,它就已经存在了。用机器计算的概念在公元前 500 年已有记载,那时巴比伦人发明了算盘,成为第一个机械式的计算器。算盘用串珠实现计算。古代的巴比伦神父用它记录他们巨大的粮食仓库。算盘被广泛使用,并一直使用到今天,到 1642 年为止并没什么改进。1642 年数学家 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,当其中一个转动一圈时,第二个齿轮推进一个位置。这和自动里程表中一样,是所有机械计算器的基础。附带说一句,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所作开拓性工作而命名的。

最早把轮式机械用于信息自动计算的实践始于 1800 年初。这是在发明灯炮之前,更是在了解电子以前。在这个计算机时代的开端,人们就梦想着使用程序计算数据的机器,而不只是像计算器那样计算。

机械式计算机的早期的开拓者是 Charles Babbage。1823 年,受英国天文协会委托,Babbage 在 Lovelace 伯爵夫人 Augusta Ada Byron 的帮助下,开始制造可编程的计算机。这个机器要为英国海军制作导航表。他接受了挑战,并开始建造他称之为“分析机”(analytical engine)的机器,这个机器就是机械式计算机,它存储 1000 个 20 位 10 进制数字和一个可变的程序,它能修改机器去执行各种计算任务。这个机器通过穿孔卡片输入,与 1950 年至 1960 年计算机使用的那种方式类似。它采纳法国人 Joseph Jacquard 提出的用穿孔卡片的主意,后者在 1801 年就用穿孔卡片作为今天被称作 Jacquard's 织布机的编织机器的输入。Jacquard's 织布机用穿孔卡片选择生产布匹复杂的编织图案,叫作穿孔卡片编程织布机。

多年工作以后,Babbage 的梦想开始淡薄,他认识到他那个时代的机器不可能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,当时不可能以足够的精密度制造出来,以便使分析机可靠地工作。

## 电子计算器时代

1800 年迎来了电动机的诞生(出自 Michael Faraday 的构想),随之把多个电动机加进 Pascal 的机械计算器机构中。直到 1970 年这些电动的机械计算器还是一件通用的办公设备,那时出现了由 Bomar 首先提出较小的手持电子计算器。Monroe 也是电子计算器的领先者,但是它是台式的、4 功能的机器,大小相当于出纳机。1889 年 Herman Hollerith 研制了存储数据的穿孔卡片。像 Babbage 那样,他显然借鉴了 Jacquard 穿孔卡片的思想。他开发了一个新式电机驱动的机器,这个计算机器,存储和校对存储在穿孔卡片上的信息。用机器计算的想法引起了英国政府的兴趣,因此政府委托 Hollerith 用穿孔卡片系统存储 1890 年的人口普查资料并制成表格。

1896 年 Hollerith 组成了叫作 Tabulating Machine Company(制表机械公司)的公司。这个公司开发了行式机器,用来穿孔制表卡片。在吞并若干个公司后,Tabulating Machine Company 变成 International Business Machines Corporation(国际商业机械公司),现在叫作 IBM。为了纪念 Herman Hollerith 我们经常要把计算机系统里的穿孔卡片叫作 Hollerith

卡,以表示敬意。穿孔卡片用的 12 位代码叫作 Hollerith 码。

用电机驱动的机械式机器,继续支配着信息处理世界,直到 1942 年第一台电子计算机出现。德国发明家 Konrad Zuse 研制出了电子计算机——Z3,第二次世界大战期间德国人用它设计飞机和导弹。德国政府给予 Had 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 最终得到了某些迟到的称颂,对他在 1930 年数字式电子领域内的开拓性工作表示敬意。

最近通过解密的英国军事文件发现第一台真正的电子计算机是在 1943 年安装运转的,它被用来破译神秘的德国军事代码。这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”,多半是因为它的尺寸很大。“巨人”的设计目标是它可以去破译由机械加密机产生的德国军事密码,但它不能解决其他问题。巨人不是可编程的,它是固定程序的计算机系统,今天通常称之为专用计算机。

第一个通用可编程计算机系统是 1946 年在宾夕法尼亚大学研制出来的。这台现代计算机叫作 ENIAC(Electronics Numerical Integrator and Calculator)。ENIAC 体积庞大,包含 17000 多个真空管和超过 500 英里的导线。这个庞然大物的重量超过 30 吨,而每秒只能执行约 100,000 次运算。ENIAC 推动世界进入了电子计算机时代。ENIAC 采用重新连接它的线路来编程,该过程需要许多天才能完成。工人们改变插塞板上的电连接,样子很像早期的电话接线总机。ENTAC 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:1948 年贝尔实验室研制了晶体管,其后 1958 年由得克萨斯仪器公司的 Jack Kilby 发明了集成电路。集成电路导致 1960 年开发出数字集成电路(RTL 即电阻-晶体逻辑),并且在 1971 年由 Intel 公司推出了第一个微处理器。当时 Intel 和他的一个工程师(Marcian T. “Ted”Hoff)研制出 4004 微处理器,这个器件开始了今天还在继续加速进行着的微处理器革命。

## 程序设计的进步

在可编程的机器被研制出来以后,就出现了程序和程序设计语言。第一个真正意义上的可编程序的电子计算机系统是通过重新连接线路来编程的。由于实际应用起来非常麻烦,因此在计算机系统发展的早期,计算机语言似乎是用来控制计算机的。这一类型的语言叫作机器语言,是用二进制代码,以被叫作程序的一组 1 或 0 的组合存储在存储器系统中的。这比用重新连接机器来编程更有效,但是它仍然是非常耗费时间的。数学家 John Von Neumann 首先开发了接受指令并且把它们存储在存储器中的系统。计算机常常称作冯·纽曼机器,以纪念 John Von Neumann。

在像 UNIVAC I 和 II 那样的计算机系统 1950 年初投入使用后,用汇编语言大大简化了把二进制代码输入计算机作为其指令的那种烦琐工作。这允许程序设计者用助记符代替一个隐秘的二进制数,例如表示加法运算的 ADD 代替了二进制数 01000111。用汇编语言辅助程序设计,计算机仍是不容易编程的,直到 1957 年 Grace Hopper 开发被叫作 FLOW-MATIC 的第一个高级程序设计语言。同年后期,IBM 为他的计算机系统开发了 FORTRAN(FORmula TRANslator)。FORTRAN 语言允许程序设计员使用解数学问题的公式去开发程序。FORTRAN 还被一些科学家采用。比 FORTRAN 大约晚一年提出了类似的语言 ALGOL(ALGORithmic Language)。

在商业应用领域,第一个真正成功并广为流传的程序设计语言是 COBAL(Computer Business-oriented Algorithmic Language)。尽管最近几年 COBAL 的使用减少了,但是在许多大的商业系统中它仍然是起主要作用。另外相当通用的商业语言是 RPG(Report Program Generator),它允许用规范的输入、输出和计算格式进行程序设计。在这些早期的程序设计语言之后,出现了更多的语言,比较流行的有: BASIC、C/C++、PASCAL 和 ADA。BASIC 和 PASCAL 都被设计成为教学用语言,但已离开教室,应用在许多计算机系统中。BASIC 语言,可能是最容易学习的,据估计,用户为个人计算机所写的程序有 80% 使用的是 BASIC 语言。最近,被称作 VISUAL BASIC 的 BASIC 新版本已经出现在 WINDOWS 环境的程序设计中, VISUAL BASIC 语言最后可能挤垮 C/C++ 和 PASCAL。

在科学应用领域,C/C++ 和 PASCAL 用于控制程序。两种语言,特别是 C/C++ 允许设计员几乎超越程序设计环境和计算机系统,完成控制。在许多场合 C/C++ 正在替代汇编语言,即使这些汇编语言在程序设计中仍然起着重要的作用。为个人计算机写的视频游戏程序几乎只用汇编语言。为了更有效地实现机器控制功能,汇编语言也散布在 C/C++ 和 PASCAL 中。

ADA 语言被美国国防部迅猛地使用,取名叫 ADA 语言是由于 Lovelace 伯爵夫人 Augusta Ada Byron 的缘故,她早在十九世纪初就开始与 Charles Babbage 一起工作。

## 微处理器时代

世界上的第一个微处理器(Intel 4004)是 4 位微处理器,按今天的标准讲,它是个弱小的单片可编程控制器。它只寻址 4096 个 4 位宽存储单元(1 位是取值为 1 或者 0 的二进制数,4 位宽的存储单元通常叫作半字节)。4004 指令系统只有 45 条指令,用 P 沟道 MOS-FET 技术制造,允许以 50 KIPs(每秒千条指令)的速度执行指令。与每秒执行 100000 条指令的 1946 年 30 吨的 ENIAC 计算机相比,它的速度是慢的,但 4004 的重量轻,还不到 1 盎司。

起先,这个器件应用量很大。4 位微处理器开始用于早期的视频游戏和基于微处理器的小型控制系统中。这种早期的视频游戏,像推圆盘游戏是 Bulley 推出的。这种早期的微处理器的主要问题是它的速度、字宽和存储容量。当 Intel 推出对早期 4004 的改进型号 4040 时,4 位微处理器的改革就结束了。尽管 4040 对字宽度和存储器尺寸方面的改进是不够的,但是 4040 是在高速下操作。其他公司,特别是得克萨斯仪器仪表公司也生产了 4 位微处理器(TMS-1000),4 位微处理器在低档应用领域依然存在,像微波炉小控制器,并且仍然可以从某些微处理器厂商那儿得到它们。大部分计算器也仍然使用 4 位微处理器,来处理 4 位 BCD(二进制编码的十进制)代码。

1971 年,认识到微处理器是个可营利的产品,Intel 公司推出了 4004 微处理器的扩展 8 位型 8008。8008 扩大了寻址存储空间(16K 字节),并且增加了指令(总数为 48 条),以便争取它在许多高档应用中的机会。字节通常是 8 位宽的二进制数,K 是 1024 字节。通常存储器尺寸按 K 字节计算。

由于工程师们研究了 8008 微处理器的许多应用需求,他们发现在一些地方它的存储空间稍小,速度慢,而且指令系统限制它的用途。Intel 认识到这些局限后,即在 1973 年推出了 8080 微处理器,这也是第一个现代 8 位微处理器。大约 6 个月以后,Intel 推出了 8088 微

处理器。Motorola 推出了它的 MC6800 微处理器,从此,研制和开发微处理器的闸门被打开了。从某种意义上讲,8080 和 MC6800 迎来了微处理器时代。不久其他公司也相继推出了他们自己的微处理器。表 1-1 列出了这些早期的微处理器以及他们的厂家。这些早期的微处理器产品中,只有 Intel 和 Motorola 继续成功地生产更新的和改进型号的微处理器。Zilog 仍在制造微处理器,但是,它作出后来证明是相当成功的决策:与其集中于通用微处理器,不如集中精力于微控制器和嵌入式控制器方面。Rockwell 几乎放弃开发现代电路的微处理器的机会。

表 1-1 早期的 8 位微处理器

制造厂	部分型号
Fairchild	F-8
Intel	8080
MOS Technology	6502
Motorola	MC6800
National Semiconductor	IMP-8
Rockwell International	PPS-8
Zilog	Z-8

8080 的特点是什么? 8080 不仅能访问更多的存储器,并执行增加了的指令集,而且指令执行起来要比 8008 快十倍。换句话说 8008 系统用  $20\mu\text{s}$ (每秒 50,000 条指令)。而 8080 系统只需要用  $2.0\mu\text{s}$ (每秒 500,000 条指令)。另一方面 8080 与 TTL(晶体管-晶体管逻辑)直接兼容,而 8008 则不是,这样就使得设计接口更加容易而且开销也少。8080 寻址的存储器空间比 8008 多 4 倍(64K 字节比 16K 字节)。这些改进成为导致微处理器空前繁荣的 8080 时代到来的原因。

第一台个人计算机 MITS Altair 8800 是 1974 年推出的。(选择 8800 这个名字,大概是为了避免侵犯 Intel 的版权。)为 Altair 8800 计算机编写的 Basic 语言解释程序是由 Bill Gates 开发的,他后来创立了微软公司。Altair 8800 的汇编程序是由数字研究公司写的,它现在为个人计算机研制了 DR-DOS。

**8085 微处理器** 1977 年 Intel 公司推出了 8080 的更新型号 8085。这是 Intel 公司开发的最后一款 8 位通用的微处理器。尽管只比 8080 稍微先进了一些,然而 8085 以更高的速度执行软件。总而言之,如果 8080 要花费  $2.0\mu\text{s}$ (每秒 500,000 条指令),则 8085 只须花费  $1.3\mu\text{s}$ (每秒执行 769230 条指令)。8085 的主要优点是它有内部的时钟发生器,内部的控制系统和较高的时钟频率。这种较高的元件集成度降低了成本,并且推动了 8085 微处理器的实际应用。Intel 已经销售了 1 亿片 8085 微处理器,在 8 位微处理器的生产和销售方面获得了极大的成功。因为 8085 也被许多其他公司(第二货源)制造,所以这种微处理器现在已经超过了 2 亿块。含有 8085 的应用仍然被使用和设计着,并且有可能继续使用下去。其他公司,Zilog 销售 5 亿多 8 位微处理器,他们的 Z-80 使用与 8085 兼容的机器语言代码,这意味着执行 8085-Z-80 兼容代码的微处理器已经超过了 7 亿。

## 现代微处理器

1978 年 Intel 推出了 8086 微处理器,并在同年早些时候推出 8088,这两个器件都是 16



位微处理器,执行指令只需要 400ns(2.5MIPs,或每秒执行 2.5 百万条指令)。这表示主要改进使执行速度大大超过 8085。另外,8086 和 8088 可寻址 1M 字节存储器,比 8085 多 16 倍(1M 字节存储器容纳 1024K 字节存储单元,即 1,048,576 字节)。较快的执行速度和大存储空间允许 8086 和 8088 在许多应用方面能替换类似的微型计算机。在 8086/8088 上发现的另一个特点是指令高速缓冲存储或指令队列,在指令执行前就预取几条指令进来。指令队列加速了许多顺序存取指令的操作,并且为现代微处理器开发更大的指令高速缓冲存储器奠定了基础。

8086 和 8088 扩大了存储容量并增加了指令功能,使微处理器可用于许多复杂的应用中。改进的指令系统包括早期微处理器所没有的乘法和除法指令。另一方面,指令的数量从 4004 的 45 条增加到 8085 的 246 条,到 8086 和 8088 微处理器已超过 20,000 条。注意这些微处理器因为指令的数量和复杂程度,被叫作 CISC(Complex instructions set computers,复杂指令系统计算机)。增加的指令使开发高效和复杂应用的任务更容易,即使最初它们的数量太多,学习起来太费时间。16 位微处理器比 8 位微处理器还增加了更多的内部寄存器存储空间,这使得用它编写的软件的功能更加强大。

因为需要更大的存储容量,16 位微处理器成为发展的主流。1981 年,当 IBM 决定在它的个人计算机中使用 8088 微处理器时,就确立了 Intel 系列的名气。像电子表格、文字处理、拼写检查和计算机辞典等应用就特别强调存储器的容量,并要求比 8 位微处理器内 64K 字节更大的存储空间。16 位的 8086 和 8088 微处理器为这些应用提供 1M 字节存储器。后来,甚至提供 1M 字节存储空间的存储器也限制了大的电子表格以及其他方面的应用。这就导致 Intel 于 1983 年推出 80286 微处理器以更新 8086。

**80286 微处理器** 80286 微处理器(也是 16 位微处理器),除寻址 16M 字节存储空间取代 1M 字节存储空间外,其他方面和 8086、8088 完全相同。除了几条为管理额外 15M 字节存储器而增加的指令之外,80286 的指令系统也和 8086、8088 几乎相同。80286 增加了时钟速度,使用最初投放的时钟频率为 8.0MHz 的微处理器,它执行指令时间减少到 250ns(4.0MIPs)。有些变化还存在于指令内部的执行,对多数指令来讲,与 8086/8088 指令相比其执行的速度增加了 8 倍。

**32 位微处理器** 微处理器应用从一开始就要求更快的处理速度,更大的存储容量和更宽的数据通道。于是 Intel 在 1986 年推出 80386,80386 微处理器结构比 16 位 8086~80286 作了较大的翻改。80386 是 Intel 的第一个包含 32 位数据总线和 32 位地址线的微处理器(注意,Intel 早期生产过一个成绩不佳的 32 位微处理器叫作 iapx-432)。由于 32 位总线,80386 寻址高达 4G 字节存储器(1G 存储器包含 1024M,即 1,073,741,824 个单元)。4G 字节存储器可以惊人地存储 1,000,000 页双倍篇幅的打印数据。80386 还有几个变型版,例如 80386SX,通过 16 位数据和 24 位地址总线,可以寻址 16M 字节的存储空间,而 80386SL/80386SLC 通过 16 位数据和 25 位地址总线寻址 32M 字节存储空间。80386SLC 型含有内部高速缓冲存储器,允许它以更高的速度处理数据。近期 Intel 推出了 80386EX 微处理器,80386EX 称作嵌入式(embedded) PC,因为在单片集成电路上包含了 AT 级 PC 机的全部元件,80386EX 还包含 24 根数据输入/输出线,26 位地址总线,16 位数据总线,DRAM 刷新控制器和可编程的片选逻辑。

使用 GUI 即图形用户接口的软件系统需要高处理器速度和大存储容量。现代图像显