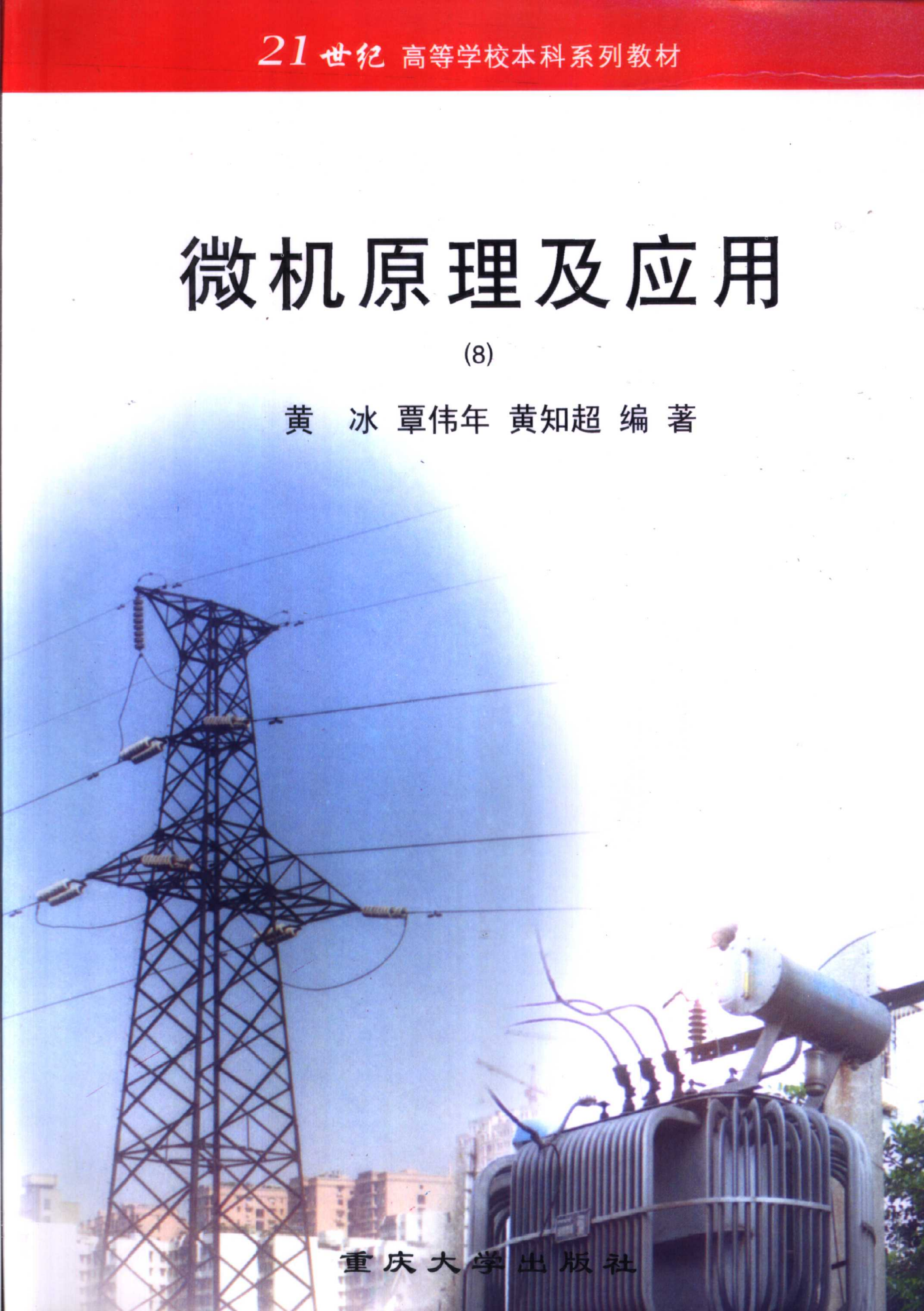
21世纪 高等学校本科系列教材

微机原理及应用

(8)

黄 冰 覃伟年 黄知超 编 著



重庆大学出版社

微机原理及应用

黄　冰　覃伟年　黄知超　编著

重庆大学出版社

内 容 提 要

本书是根据教育部《普通高等学校计算机基础教育教学基本要求(本科)》,以16/32位微型计算机为背景编写的。全书以Intel 8086为基础,追踪Intel主流系列高性能微机的技术发展方向,比较全面、系统、深入地介绍了80486微机系统、编程实例及实用接口技术。

全书共分10章,包括绪论、Intel8086微处理器、宏汇编语言程序设计、Intel 80486微处理器、半导体存储器、I/O接口技术、中断系统、常用接口芯片、总线、典型微型计算机系统等。

本书内容先进、概念清楚、叙述简洁、实用性强,可作为高等院校各专业"微机原理及应用"课程的教学用书,也适用于广大科技人员作为培训教材或自学参考书。

图书在版编目(CIP)数据

微机原理及应用/黄冰,覃伟年,黄知超编著. —重庆:重庆大学出版社,2003.1
(电气工程及其自动化专业本科系列教材)

ISBN 7-5624-2376-8

Ⅰ.微… Ⅱ.①黄…②覃…③黄… Ⅲ.微型计算机—高等学校—教材 Ⅳ.TP36

中国版本图书馆CIP数据核字(2002)第094711号

微机原理及应用

黄 冰 覃伟年 黄知超 编著
责任编辑:曾显跃 版式设计:曾显跃
责任校对:任卓惠 责任印制:张永洋

*

重庆大学出版社出版发行
出版人:张鸽盛
社址:重庆市沙坪坝正街174号重庆大学(A区)内
邮编:400044
电话:(023) 65102378 65105781
传真:(023) 65103686 65105565
网址:http://www.cqup.com.cn
邮箱:fxk@cqup.com.cn (市场营销部)
全国新华书店经销
重庆大学建大印刷厂印刷

*

开本:787×1092 1/16 印张:24 字数:599千
2003年1月第1版 2003年1月第1次印刷
印数:1—6 000
ISBN 7-5624-2376-8/TP·317 定价:28.00元

前言

21世纪的到来和计算机技术的迅速发展、普遍应用，给高等学校的计算机基础教育提出了新的课题。特别是对于工科的学生来说，除了掌握计算机基础知识，能应用一些常用软件和进行高级语言程序设计外，还应对计算机的工作原理及其在实时控制等领域的应用有一定的了解。“微机原理及应用”就是为工科学生学习和掌握微型计算机的硬件和软件（汇编语言）而设置的一门技术基础课程。其任务是使学生从应用的角度出发，在理论和实践上掌握微型计算机的体系结构、汇编语言程序设计和I/O接口技术，具有利用微机从事应用开发的基本能力。

本书是根据教育部《普通高等学校计算机基础教育教学基本要求（本科）》的精神编写的。在编写过程中，我们力求讲透基本原理，做到基础性和先进性的统一。本书作为本科学生学习计算机硬件和汇编语言的入门课程，重点介绍一般计算机系统的基本结构和组成原理，使学生能举一反三，为以后学习、应用新型微型计算机打下基础；与此同时，兼顾内容的系统性和先进性，注意追踪微机及应用技术的新水平与发展趋势。其次，强调实用性，书中CPU和芯片内部结构尽量简化，一般只给出编程模型，强化外部接口和应用。另外，在教学机型的选择上，根据国内高校的教学与实验条件，追踪Intel 80x86、Pentium系列主流机型，重点介绍了8086/80486微处理器。Intel微处理器系列在不断发展，但其基本功能都具有相似性。本着由浅入深的原则，先介绍8086微处理器，在此基础上，再去学习复杂的高档微处理器。对于“接口技术”内容的选取与组织也是如此。随着微电子技术的发展，与各种高档微处理器配套的超大规模多功能外围接口芯片不断出现，而直接用这些多功能外围芯片介绍接口技术是很困难的。本书仍从8/16位微机中广泛应用的可编程通用接口芯片入手，使学生掌握软、硬件相结合的接口技术。而多功能外围芯片基本上是上述芯片功能的集成，在此基础上介绍则相对容易理解。

本教材课堂讲授的参考学时数为 60 ~ 80，如果教学时数少，可根据需要选学有关章节。必须指出，《微机原理及应用》是一门实践性很强的课程，应当有充足的时间来培养和提高学生的实际动手能力，建议安排不少于 20 学时的实验。

本书共 10 章，由黄冰主编，并编写第 2、3、4 章及附录，第 1、8、9、10 章由黄知超编写，第 5、6 章由覃伟年编写，第 7 章由潘盛辉编写。

本书在编写过程中得到了各方面的支持和帮助，参考了国内外有关的教材和资料，在此一并表示衷心的感谢。

由于编者水平有限，疏漏之处在所难免，恳请读者批评指正。

编者

2002 年 8 月

目录

第 1 章　绪论 …… 1
1.1　微型计算机发展概况 …… 1
1.2　计算机中数和字符的表示 …… 2
1.3　微型计算机系统概论 …… 8
习题 …… 19
第 2 章　Intel 8086 微处理器 …… 20
2.1　8086 微处理器的内部结构 …… 20
2.2　8086 引脚功能 …… 26
2.3　8086 系统总线时序 …… 30
2.4　8086 寻址方式 …… 33
2.5　8086 指令系统 …… 37
习题 …… 58
第 3 章　宏汇编语言程序设计 …… 62
3.1　汇编语言的语句格式 …… 62
3.2　汇编语言的数据项 …… 63
3.3　汇编语言的表达式 …… 66
3.4　伪指令语句 …… 71
3.5　汇编语言程序设计概述 …… 75
3.6　顺序程序设计 …… 76
3.7　分支程序设计 …… 78
3.8　循环程序设计 …… 86
3.9　DOS 系统功能调用 …… 95
3.10　子程序设计 …… 98
3.11　宏指令 …… 113
3.12　汇编语言程序的建立、汇编、连接与调试 …… 118
习题 …… 125
第 4 章　Intel 80486 微处理器 …… 129
4.1　80486 内部结构 …… 130
4.2　80486 的工作方式 …… 136
4.3　80486 引脚功能 …… 142

4.4 80486 的寻址方式 …… 146
4.5 80486 常用指令介绍 …… 149
4.6 80486 编程举例 …… 159
习题 …… 170
第 5 章 半导体存储器 …… 175
5.1 存储器概述 …… 175
5.2 随机存储器 RAM …… 178
5.3 只读存储器 ROM …… 186
5.4 存储器与 CPU 的连接 …… 191
5.5 PC 机的存储器 …… 200
5.6 高速缓冲存储器系统 …… 202
习题 …… 206
第 6 章 I/O 接口技术 …… 208
6.1 概述 …… 208
6.2 程序控制的 I/O …… 214
6.3 DMA 方式 …… 217
习题 …… 233
第 7 章 中断系统 …… 235
7.1 概述 …… 235
7.2 16 位微机中断系统 …… 239
7.3 32 位微处理器的中断 …… 244
7.4 中断控制器 8259A …… 247
习题 …… 261
第 8 章 常用接口芯片 …… 264
8.1 并行接口芯片 8255A …… 264
8.2 定时器/计数器接口芯片 8253 …… 275
8.3 串行接口芯片 8251A …… 287
8.4 模拟接口 …… 306
8.5 多功能外围接口芯片 82380 …… 320
习题 …… 329
第 9 章 总线 …… 334
9.1 概述 …… 334
9.2 ISA 总线 …… 341
9.3 EISA 总线 …… 342
9.4 PCI 总线 …… 343
习题 …… 345
第 10 章 典型微型计算机系统 …… 346
10.1 IBM PC/XT 微型计算机系统 …… 346

10.2 80486 微型计算机系统 …… 349
10.3 Pentium 系列微型计算机系统 …… 350
习题 …… 352
附录 …… 353
附录 1 ASCII 码(美国标准信息交换码)表 …… 353
附录 2 80x86/Pentium 指令系统 …… 354
附录 3 指令对状态标志的影响(未列出的指令不影响标志) …… 360
附录 4 常用 DOS 系统功能调用 …… 361
附录 5 DEBUG 主要命令 …… 368
参考文献 …… 373

第1章 绪 论

1.1 微型计算机发展概况

电子计算机是由各种电子器件组成的能够自动、高速、精确地进行逻辑控制和信息处理的现代化设备。电子计算机按其性能来分,有巨型、中型、小型和微型计算机。自从1946年第一台电子计算机出现至今的50多年来,电子计算机已经历了电子管计算机、晶体管计算机、集成电路计算机、大规模/超大规模集成电路计算机四代的发展和变迁,并开始了以模拟人的大脑神经网络功能为基础的第五代计算机的研究,每一代都向着小体积、轻重量、高性能的方向发展。

微型计算机(简称微机)作为第四代计算机的典型代表于20世纪70年代随着大规模、超大规模集成电路的诞生而发展起来。由于微型计算机性能价格比在各种机型中占有领先地位,且小巧灵活,所以深受用户欢迎并发展迅速。构成微机的核心部件是微处理器MPU(Microprocessor),也叫中央处理器或中央处理单元CPU(Central Processing Unit)。微型计算机的发展是与微处理器的发展同步的,30多年来,微处理器的集成度和性能几乎每2~3年提高一倍,已经推出了四代微处理器产品,并进入第五代。微型计算机各代的划分通常是以其微处理器的字长、位数和功能为主要依据。

第一代(1971—1973年)是4位和低档8位微机。代表产品是美国Intel公司制成的4004和8008微处理器以及MCS—4微型计算机。这一代产品采用了PMOS工艺,基本指令执行时间约为20μs,指令系统比较简单,运算功能较差,速度较慢。软件主要采用机器语言或简单的汇编语言,价格低廉。

第二代(1973—1978年)是中高档8位微机。1973—1975年为典型的第二代,以美国Intel公司的8080和Motorola公司的MC 6800为代表。1976—1978年为高档的8位微型计算机,以美国Zilog公司的Z80和Intel公司的8085为代表。这一代产品的特点是采用NMOS工艺,集成度比第一代提高1~4倍,运算速度提高10~15倍,基本指令执行时间为1~2μs,指令系统比较完善,已具备典型的计算机体系结构以及中断、DMA等控制功能,寻址能力也有所增强,软件除采用汇编语言外,还配有BASIC、FORTRAN等高级语言及相应的编译程序,并配有操作

系统。

第三代(1978—1984 年)是 16 位微机。Intel 公司的 8086/8088、Motorola 公司 MC 68000 和 Zilog 公司的 Z8000 作为典型产品相继问世,它们成为当时国内外市场上最流行的三种 16 位微处理器,采用 NMOS 工艺,集成度为 20 000 ~ 70 000 晶体管/片,基本指令执行时间约为 0.5μs。这类 16 位微处理器具有丰富的指令系统,采用多级中断系统、多重寻址方式和多种数据处理方式。构成的典型微机是 IBM PC 以及 IBM PC/XT。1984 年 Intel 公司研制成了 80286 等性能更为优越的 16 位微处理器,其特点是从单元集成过渡到系统集成以获得尽可能高的性能价格比。80286 是为满足多用户和多任务系统的需要而设计的,寻址空间已达 16MB,数据线是 16 位,速度比 8086 快 5 ~ 6 倍,它本身就包含存储器管理和保护机构,支持虚拟存储体系,IBM PC/AT 微机是其典型代表机型。

第四代(1984—1992 年)是 32 位微型计算机的大发展时期。最早的机型是采用 80386 微处理器的 IBM PC/80386 微机。20 世纪 80 年代后期推出的 Intel 80486、MC68030 和 MC68040 等属于高档 32 位微处理器。这些高档 32 位微处理器的显著特点是吸取了大中型计算机的体系结构的优点,采用了先进的超大规模集成电路 SLSI(Super Large Scale Integration)技术以及多级流水线和虚拟存储管理技术,从而使其集成度和运算速度更上一层楼。IBM PC/80486 就是采用这种高性能 32 位微处理器的典型微型计算机。

第五代(1993 年以后)是 64 位微机发展时期。Intel 公司先后推出了 Pentium/Pentium Pro/Pentium Ⅱ/Pentium Ⅲ/Pentium Ⅳ/Itanium 微处理器芯片,AMD 公司的 K6/K6 Ⅱ/K6 Ⅲ/K7 也是功能相当的微处理器芯片。这些芯片的最主要特点是采用了新的体系结构和现代先进的计算机技术(如整数嵌入技术、流水线技术、乱序执行技术以及超通道技术等),因而芯片的集成度和运算速度均大大优于 32 位微处理器。

展望未来,计算机的发展必将有很多新的突破。光学技术、超导技术、仿生技术的相互结合,必然产生一种全新的计算机,而人工智能的研究正在促进计算机面临一场新的革命。人工智能计算机、人工神经网络计算机……,这些"新一代"计算机的争相问世,将为 21 世纪人类进入智能信息社会作必要的准备。毫无疑义,计算机在未来社会中将会发挥越来越重要的作用。

1.2 计算机中数和字符的表示

1.2.1 无符号数表示

计算机中字长是一定的,因此,在表示有符号数与无符号数时数值范围是有区别的。若所表示的是无符号数,则机器字长的所有位都参与表示数值。

若计算机的字长为 n 位,则 n 位无符号数可表示的数 X 的范围是

$$0 \leqslant X \leqslant 2^n - 1$$

当 $n=8$ 时,可表示的无符号数的范围为 0 ~ 255;当 $n=16$ 时,可表示的无符号数的范围为 0 ~ 65 535。

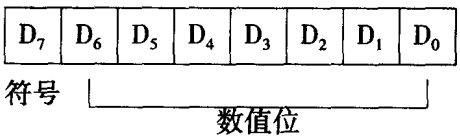
在计算机中最常用的无符号整数是表示地址的数。此外,如双精度数的低位字也是无符

号整数等。

1.2.2 有符号数的表示方法

(1)机器数与真值

计算机中的数是用二进制表示的,数的符号也是用二进制表示的。通常一个数的最高位为符号位,为0表示正数,为1表示负数。若字长为8位的计算机,则D_7为符号位,$D_6 \sim D_0$为数值位,如下所示:



例1.1 $X=+65$ 在机器中表示为:

$X=01000001B$

这种符号数码化的数称为机器数。机器数所代表的实际数值称为真值。

机器数可以用不同的码制来表示,常用的有原码和补码表示法。

(2)原码表示法

最高位为符号位,正数的符号位用0表示,负数的符号位用1表示,其余各位为数值位,这种表示法称为原码表示法。

例1.2 若 $X=+97$ 则 $[X]_{原}=01100001B$

若 $X=-97$ 则 $[X]_{原}=11100001B$

原码表示数0有两种表示形式:

$[+0]_{原}=00000000B$

$[-0]_{原}=10000000B$

n 位原码可表示的数 X 的范围是:

$$-2^{n-1}+1 \leqslant X \leqslant +2^{n-1}-1$$

当 $n=8$ 时,8位二进制原码所能表示的数值范围为 $-127 \sim +127$。

用原码表示机器数简单、直观,与真值的转换方便,缺点是进行减法运算时麻烦。为此,引入了补码表示法,它可以使减法运算简化为单一的加法运算。

(3)补码表示法

补码表示法中,正数的补码和原码相同,负数的补码可由其原码除符号位保持不变外,其余各位按位取反,再在最末位加1而形成。

例1.3 假设机器字长为8位,则

$[+97]_{补}=01100001B$

$[-97]_{补}=10011111B$

补码具有以下特点:

①$[+0]_{补}=[-0]_{补}=00000000$

②n 位二进制补码所能表示的数值范围为:

$$-2^{n-1} \leqslant X \leqslant +2^{n-1}-1$$

若 $n=8$,则8位二进制补码所能表示的数值范围为 $-128 \sim +127$。

③一个用补码表示的负数，如将$[X]_{补}$再求一次补，即将$[X]_{补}$除符号位外取反并在最末位加1就可得到$[X]_{原}$。用下式表示为：

$$[[X]_{补}]_{补}=[X]_{原}$$

表1.1　数的表示方法

二进制数码表示	无符号二进制数	原码	补码
00000000	0	+0	0
00000001	1	+1	+1
00000010	2	+2	+2
⋮	⋮	⋮	⋮
01111110	126	+126	+126
01111111	127	+127	+127
10000000	128	-0	-128
10000001	129	-1	-127
10000010	130	-2	-126
⋮	⋮	⋮	⋮
11111110	254	-126	-2
11111111	255	-127	-1

例1.4　若$[X]_{原}=11010101B$

$[X]_{补}=10101011B$

则　$[[X]_{补}]_{补}=11010101B=[X]_{原}$

(4)补码的加减运算

补码的加法运算规则是：

$$[X+Y]_{补}=[X]_{补}+[Y]_{补}$$

该式表明，当有符号的两个数采用补码形式表示时，进行加法运算可以把符号位和数值位一起进行运算(若符号位有进位，则丢掉)，结果为两数之和的补码形式。下面通过具体例子可以验证该公式的正确性。

例1.5　用补码进行下列运算：(+33)+(+15)；(+33)+(-15)

解

```
  00100001B  [+33]补          00100001B  [+33]补
+ 00001111B  [+15]补        + 11110001B  [-15]补
------------------          ------------------
  00110000B  [+48]补        | 1 | 00010010B  [+18]补
                                ↑
                                └自然丢失
```

补码的减法运算规则是：

$$[X-Y]_{补}=[X]_{补}+[-Y]_{补}$$

该式表明，求$[X-Y]_{补}$可以用$[X]_{补}$与$[-Y]_{补}$相加来实现，这里的$[-Y]_{补}$是对减数进行

求负操作。一般称已知 $[Y]_{补}$ 求得 $[-Y]_{补}$ 的过程叫变补或求负，方法是将 $[Y]_{补}$ 的每一位(含符号位)都按位取反，然后最末位加1。

例1.6 用补码进行 $X-Y$ 运算。

解 若 $X=+33$ $Y=+15$

则 $[X]_{补}=00100001B$ $[Y]_{补}=00001111B$ $[-Y]_{补}=11110001B$

$$\begin{array}{rll} & 00100001B & [X]_{补} \\ + & 11110001B & [-Y]_{补} \\ \hline \text{自然丢失}\rightarrow\ \boxed{1} & 00010010B & [+18]_{补} \end{array}$$

若 $X=-33$ $Y=-15$

则 $[X]_{补}=11011111B$ $[Y]_{补}=11110001B$ $[-Y]_{补}=00001111B$

$$\begin{array}{rll} & 11011111B & [X]_{补} \\ + & 00001111B & [-Y]_{补} \\ \hline & 11101110B & [-18]_{补} \end{array}$$

引入补码后，将减法运算转化为易于实现的加法运算，且符号位也当作数据相加，从而可简化运算器的结构，提高运算速度。因此，在微型计算机中，有符号数通常都用补码表示，得到的是补码表示的结果。

当字长由8位扩展到16位时，对于用补码表示的数，正数的符号扩展应该在前面补0，而负数的符号扩展应该在前面补1。

例如，已经知道机器字长为8位，则 $[+46]_{补}=00101110B$，$[-46]_{补}=11010010B$，如果要把它们从8位扩展到16位，那么

$[+46]_{补}=0000\ 0000\ 0010\ 1110B=002EH$

$[-46]_{补}=1111\ 1111\ 1101\ 0010B=FFD2H$

(5)有符号数运算时的溢出问题

当两个有符号数进行加减运算时，如果运算结果超出可表示的有符号数的范围时，就会发生溢出，使计算结果出错。显然，只有两个同符号数相加或两个异号数相减时，才会产生溢出。

例1.7 设机器字长为8位，以下运算都会发生溢出。

$(+88)+(+65)=+153>127$

$(+88)-(-65)=+153>127$

$(-83)-(+80)=-163<-128$

1.2.3 定点数和浮点数

在计算机中，数值数据有两种表示法：定点表示法和浮点表示法。它们分别称为定点数和浮点数。下面作简单介绍：

(1)定点数

定点数是指小数点在数中的位置是固定不变的，常用的定点数有纯小数和纯整数两种。

①纯小数：小数点固定在符号位之后，如1.1100111，此时机器中所有数均为小数。

②纯整数；小数点固定在最低位之后，如11100111.，此时机器中所有数均为整数。

机器字长为8位的有符号纯整数与纯小数表示范围如图1.1所示。从图中可以看出，小数点在计算机中是不表示出来的，采用哪一种表示方法，由人们事先约定。

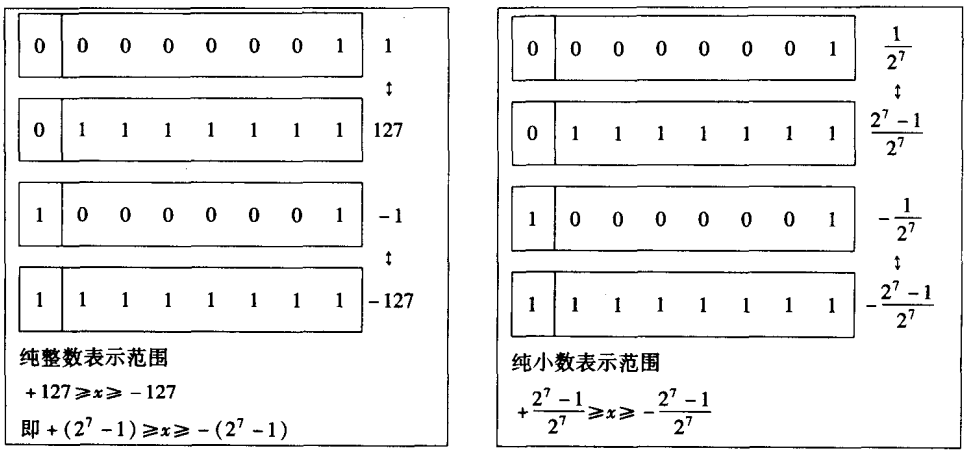


图 1.1　8 位有符号纯整数与纯小数

(2)浮点数

浮点数由阶码和尾数两部分组成。对任意一个有符号的二进制数 N 的普遍形式可表示为:

$$N = 2^E \times M$$

式中,E 称为阶码,它是一个有符号的可变整数。设

$$E = e_j e_{k-1} \cdots e_0$$

e_j为阶符:若 $e_j = 0$,则 E 是正数;若 $e_j = 1$,则 E 为负数。$e_{k-1} \cdots e_0$是阶值。

M 称为 N 的尾数,它是一个有符号的纯小数。设

$$M = m_j m_1 \cdots m_n$$

m_j为尾符:若 $m_j = 0$,则 M 为正数;若 $m_j = 1$,则 M 为负数。尾数 M 的符号就是浮点数 N 的符号。$m_1 \cdots m_n$是尾值。

浮点数 N 在计算机内的表示形式如下所示。

e_j	$e_{k-1}e_{k-2}\cdots e_0$	m_j	$m_1 m_2 \cdots m_n$
阶符	阶值	尾符	尾值

(3)规格化数与溢出

为了便于浮点数的运算,数采用规格化表示。对尾数规格化作如下定义:若 $m_j \neq m_1$,则称尾数 M 为规格化数;若 $m_j = m_1$,则称尾数 M 为非规格化数。

如果尾数不是规格化数,那么要用移位手段把它变为规格化数。尾数每左移一位,阶码就减 1,尾数每右移一位,阶码就加 1,直至 $m_j \neq m_1$为止。以左移操作实现尾数的规格化称为左规,以右移实现规格化称为右规。

存储在计算机中的数一定是规格化数。两数的运算结果也应为规格化数,如果不是,那么必须通过移位方式把它变为规格化数。若是规格化数,则 $1/2 \leqslant |M| < 1$。

例如,$N = 2^{011} \times 0.0010100$,显然,尾数 0.001010 为非规格化数,将 0.0010100 左移两位后,变成 0.1010000,此数已是规格化数,不再左移,从阶码 011 中减去 010,得 001。所以,规格化后的 N 应为 $2^{001} \times 0.1010000$。

当浮点数超出机器所能表示数的范围时,称为溢出。对规格化的浮点数,当阶码小于机器

所能表示的最小数时,称为下溢,此时机器将把此数作0处理;若阶码大于机器所能表示的最大范围时,称为上溢。溢出发生时,机器就产生溢出中断,进入中断处理。

浮点表示法比定点表示法所表示的数的范围大,精度高,但运算规则比较复杂,成本较高。早期的微型计算机采用定点表示,机器中数均为整数,没有处理浮点数的指令。为了弥补这方面的不足,专门设计了相应的数值协处理器(8087、80287、80387等)来实现对浮点数的运算。80486、80586的数值协处理器已集成在CPU芯片内部。在本教材中,若无特别说明,数据均采用纯整数定点表示。

1.2.4 计算机中的二进制编码

(1)BCD码(Binary Coded Decimal)

BCD码是以4位二进制的不同组合表示十进制数十个数码的方法,又称二—十进制编码。

常用的BCD码为8421 BCD码,即每位十进制数码用4位二进制数来表示,4位二进制数从高到低的权值分别为2^3、2^2、2^1、2^0即8421。8421 BCD码如表1.2所示。

表1.2 8421BCD码

十进制数	8421BCD码	十进制数	8421BCD码
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

这种BCD码与十进制数的关系直观,其相互转换也很简单。

例1.8 十进制数和BCD码相互转换。

将十进制数86.5转换为BCD码:$86.5=(1000\ 0110.0101)_{BCD}$

将BCD码1001 0111.0100转换为十进制数:$(1001\ 0111.0100)_{BCD}=97.4$

在IBM PC机中,根据在存储器中的不同存放格式,BCD码又分为压缩型BCD码和非压缩型BCD码。压缩型BCD码是在一个字节中存放两个十进制数码,而非压缩型BCD码每个字节只存放一个十进制数。

例如,将十进制数8762用压缩型BCD码表示,则为:

1000 0111 0110 0010

在存储器中的存放格式为:

01100010
10000111

而用非压缩型BCD码表示,则为:

0000 1000 0000 0111 0000 0110 0000 0010

在存储器中的存放格式为:

00000010
00000110
00000111
00001000

(2) ASCII 码(American Standard Code for Information Interchange)

由于计算机处理信息要涉及各种字符,这些字符都必须用二进制代码来表示,在微型计算机中普遍采用美国信息交换标准码,即 ASCII 码。附录 1 给出了 ASCII 码表,以便查阅。

ASCII 码采用 7 位二进制编码,总共有 128 个字符,包括 52 个英文大、小写字母,10 个阿拉伯数字 0 ~ 9,32 个通用控制字符和 34 个专用字符。例如,数字 0 ~ 9 的 ASCII 码分别为 30H ~ 39H,英文大写字母 A ~ Z 的 ASCII 码为 41H ~ 5AH。ASCII 码表中有一些符号是作为计算机控制字符使用的,这些控制符号有专门用途,表中给出了这些控制字符的含义。例如,回车字符 CR 的 ASCII 码为 0DH,换行符 LF 的 ASCII 码为 0AH 等。

通常,7 位 ASCII 码在最高位加一个 0 组成 8 位代码。因此,字符在计算机内部存储时,正好占一个字节。在存储和传送信息时,最高位常用作奇偶校验位,用来检验代码在存储和传送过程中是否发生错误。奇校验时,每个代码的二进制形式中应有奇数个 1。例如,传送字母 A,其 ASCII 码的二进制形式为 1000001,因有两个 1,故奇校验位为 1,8 位代码将是 11000001。偶校验每个代码中应有偶数个 1。若用偶校验传送字符 A,则 8 位代码为 01000001。

7 位二进制码称为标准的 ASCII 码。近年来,在标准 ASCII 码基础上,为表示更多符号,将 7 位 ASCII 码扩充到 8 位,可表示 256 个字符,称为扩充的 ASCII 码。扩充的 ASCII 码可以表示某些特定的符号,如希腊字符、数学符号等。扩充的 ASCII 码只能在不用最高位做校验位或其他用途时使用。

1.3 微型计算机系统概论

1.3.1 微处理器、微型机、微机系统之间的关系

(1)微处理器

微处理器是一个由算术逻辑运算单元、控制器单元、寄存器组以及内部系统总线等组成的大规模集成电路芯片,它具有 CPU 的全部功能。因此,微处理器通常又简称为 CPU。

(2)微型计算机

微型计算机是以微处理器芯片为核心,配上内存芯片、I/O 接口电路以及相应的辅助电路构成的装置,它又简称为微型机。

(3)微型计算机系统

微型计算机系统是以微型计算机为主体,配上输入设备、输出设备、外存储器设备、电源、机箱以及基本系统软件组成的系统,它又简称为微机系统。

1.3.2 微机硬件系统组成

(1)微机硬件系统基本结构

微机硬件系统的基本结构由中央处理器CPU、存储器、接口电路、外部设备以及系统总线等组成,如图1.2所示。

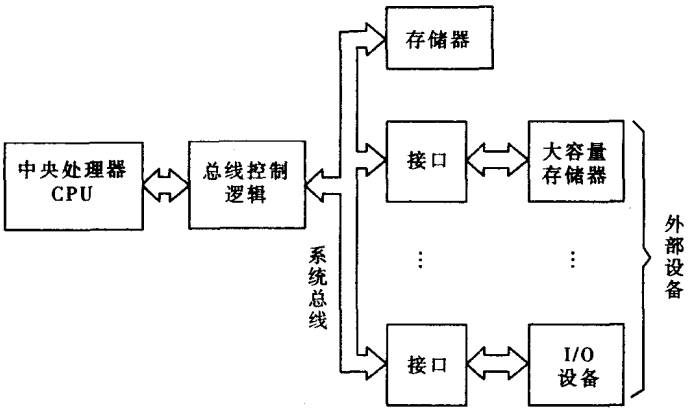


图1.2 微机硬件系统基本结构

中央处理器(CPU)包括运算器、控制器和寄存器组三个主要单元。运算器的功能是完成数据的算术运算和逻辑运算操作。控制器把指令从存储器中取出,译码后发出相应的控制信号,使各部件相互协调工作,从而完成整个微机系统的控制。寄存器组则是用来存放CPU频繁使用的数据和地址信息,这样可加快CPU访问的速度。

存储器是微机存放、记忆程序和数据的装置。它由许多存储单元构成,每一个存储单元可以存放和记忆若干位二进制代码。通常把位于主机内部的用于暂时存放程序和数据的存储器称为内存,也称为主存。它由只读存储器(ROM)和随机存储器(RAM)两部分组成。位于主机外部的用于存放大量信息的存储器则称为外存。

外部设备一般包括I/O(Input/Output)设备和外存储器,也称为计算机的外设。I/O设备是指负责计算机与外界通信用的输入和输出设备,如显示器、键盘、打印机、鼠标器、扫描仪、数字化仪、条码读入器等多种类型的外部设备。外存储器则是指机器外部可存储大量信息的存储器,如磁盘、磁带、光盘等,存取信息的速度要比内存慢得多。因此,除必要的系统程序外,一般程序(包括数据)是存放在外存中的,只有当运行时,才把它从外存传送到内存的某个区域,再由CPU控制执行。

接口电路是设置在外设与CPU之间的专门电路,又称I/O接口,用于协调CPU与外设之间的信息传输。

系统总线把CPU、存储器和接口电路连接起来,用来传送各部分之间的信息。系统总线包括数据总线、地址总线和控制总线,简称三总线。数据总线传送数据(包括指令代码、原始数据、中间数据和结果数据),地址总线上的信息(即地址)指出数据的来源和目的地,控制总线传送CPU对存储器或I/O设备的控制命令和I/O设备对CPU的请求信号。系统总线的工作由总线控制逻辑负责指挥。