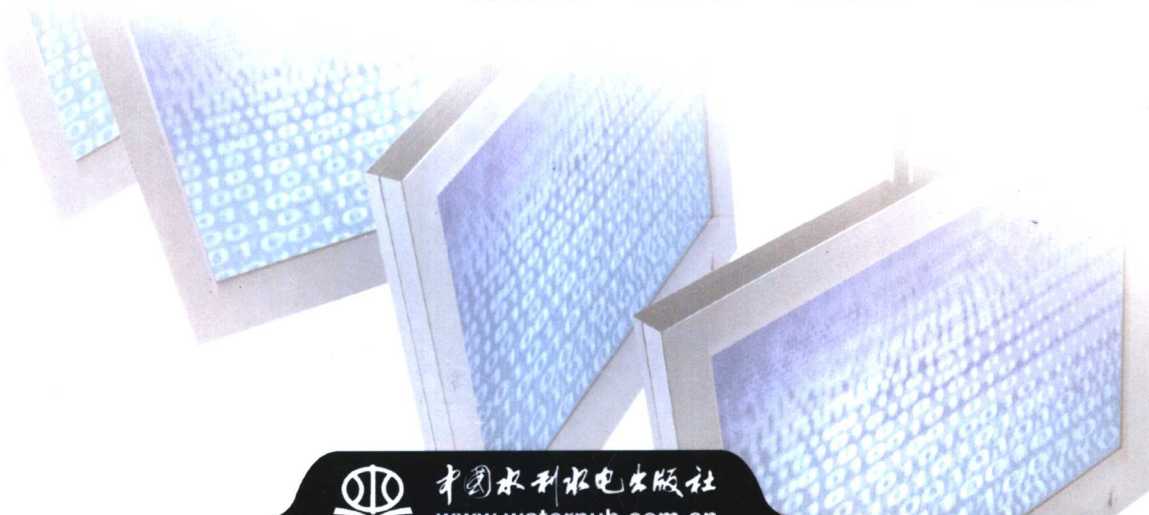




微机原理及接口技术

蒋新革 主 编

潘伟民 钟志明 王崇国 副主编



21 世纪高等院校计算机科学与技术规划教材

微机原理及接口技术

蒋新革 主 编

潘伟民 钟志明 王崇国 副主编

中国水利水电出版社

内 容 提 要

本书根据 21 世纪高等院校规划教材的编写要求,参考近年出版的多种教科书,以实用为目的进行编写。本书内容的组织以培养学生应用能力为主要目标,注重基本知识和应用技术,理论与实践相结合,以 Intel 8088/8086 微处理器和 IBM PC 系列机为主体,论述了 16 位微型计算机的基本原理、汇编语言和接口技术,并引出了 32 位微机,对于计算机在原理与接口方面的最新发展也做了适度介绍。全书共分 10 章,主要包括:微型计算机基础知识、微型计算机组成与工作原理、汇编语言程序设计、基本输入/输出接口、半导体存储器及其接口、中断控制技术、定时/计数器、A/D 与 D/A 转换接口技术、微型计算机外部设备及开发应用等有关知识。本书不但涉及内容广泛、涵盖知识点全面,而且条理清晰、通俗易懂、图文并茂,有利于学生系统性地学习。

本书可作为高等院校计算机类和机电类相关专业学生的教材,也可以作为高等教育自学教材,或作为从事微型计算机硬件和软件开发工程技术人员的学习和应用参考书。

本书所配电子教案可以从中国水利水电出版社网站上免费下载,网址为:
<http://www.waterpub.com.cn/softdown/>。

图书在版编目(CIP)数据

微机原理及接口技术 / 蒋新革主编. —北京:中国水利水电出版社, 2006

(21 世纪高等院校计算机科学与技术规划教材)

ISBN 7-5084-3844-2

I. 微... II. 蒋... III. ①微型计算机—理论—高等学校—教材②微型计算机—接口—高等学校—教材 IV. TP36

中国版本图书馆 CIP 数据核字 (2006) 第 073600 号

书 名	微机原理及接口技术
作 者	蒋新革 主 编 潘伟民 钟志明 王崇国 副主编
出版 发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 63202266 (总机)、68331835 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	787mm×1092mm 16 开本 16.5 印张 402 千字
版 次	2006 年 7 月第 1 版 2006 年 7 月第 1 次印刷
印 数	0001—5000 册
定 价	25.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

前 言

本课程是高等院校计算机类和机电类相关专业学生的一门主干专业基础课程。其任务是使学生掌握必要的计算机硬件和接口控制方面的知识,掌握微型计算机工作原理和外部设备的控制方法,熟悉汇编语言指令系统,了解计算机系统常见外围设备的功能和使用方法,为大学生进一步学习专业知识和提高综合素养,适应社会变化及进一步学习打下基础。

本课程所需的前导课程为计算机文化基础、高级语言程序设计、数字电路等基础课程。在本课程的学习过程中应该结合实验同步进行,特别是计算机接口方面的硬件实验,这对培养学生分析问题和解决问题的能力非常重要。

为适应不同专业不同学生的实际情况,本教材十分重视内容取舍。对于重点内容,必须讲透,对于可以讲可以不讲的内容坚决舍弃。各地在教学过程中,可以对部分内容适当变更。

全书共分 10 章,主要包括:微型计算机基础知识、微型计算机组成与工作原理、汇编语言程序设计、基本输入/输出接口、半导体存储器及其接口、中断控制技术、定时/计数器、A/D 与 D/A 转换接口技术、微型计算机外部设备及开发应用等。本书不但涉及内容广泛、涵盖知识点全面,而且条理清晰、通俗易懂、图文并茂,有利于学生系统性地学习。

本书由蒋新革担任主编,负责全书的策划和统稿;潘伟民、钟志明、王崇国担任副主编,编委有程如铁、乎西旦、李玉强、马晶、沙塔尔、王平等,协助完成本书的编写与统稿工作。

本书在编写过程中得到各参编学校的大力支持,并得到中国水利水电出版社的指导和热情帮助,在此表示由衷的感谢。

由于编者的水平有限,书中一定存在不少差错和疏漏,诚请广大读者和各位同仁批评指正。

编 者

2006 年 3 月

目 录

前言

第1章 微型计算机基础知识	1
本章要点	1
1.1 数制	1
1.1.1 进位计数制及其基数和权	1
1.1.2 计算机中常用的进制数	1
1.1.3 进制间的转换	3
1.2 数据在计算机中的表示	5
1.2.1 机器数与真值	5
1.2.2 数的定点与浮点表示	6
1.2.3 计算机中常用的编码	7
1.3 数据在计算机中的运算	9
1.3.1 二进制数的逻辑运算	9
1.3.2 整数的四则运算	10
1.4 逻辑功能部件	11
1.4.1 二极管及三极管	11
1.4.2 门电路及逻辑运算部件	11
1.4.3 算术逻辑部件 (ALU)	12
习题一	13
第2章 微型计算机的组成及工作原理	15
本章要点	15
2.1 微型计算机概述	15
2.1.1 微型计算机的发展与应用	15
2.1.2 微型计算机的组成	16
2.1.3 微型计算机的工作原理	18
2.1.4 微型计算机的特点与分类	20
2.2 微处理器	20
2.2.1 Intel 8086/8088 微处理器	21
2.2.2 Intel 8086/8088 存储器的结构	27
2.2.3 Intel 8086/8088 总线结构和总线周期	28
2.2.4 微处理器的发展	34
2.3 微型计算机总线接口	37
2.3.1 概述	37

2.3.2	ISA 总线和 EISA 总线.....	37
2.3.3	VESA (VL-Bus) 总线.....	38
2.3.4	PCI 总线.....	38
2.3.5	USB 总线.....	39
2.3.6	RS-485 总线.....	39
2.3.7	IEEE-488 总线特性—并行总线接口标准.....	40
2.3.8	RS-232 总线.....	40
	习题二.....	40
第 3 章	汇编语言程序设计.....	42
	本章要点.....	42
3.1	汇编语言的基本语法.....	42
3.1.1	基本概念.....	42
3.1.2	汇编语言的基本语法.....	43
3.2	寻址方式.....	43
3.2.1	寻址方式概述.....	43
3.2.2	与数据有关的寻址方式.....	44
3.2.3	与转移地址有关的寻址方式.....	48
3.3	指令系统.....	50
3.3.1	概述.....	50
3.3.2	数据传送类指令.....	50
3.3.3	算术运算类指令.....	54
3.3.4	逻辑运算与移位类指令.....	60
3.3.5	字符串操作类指令.....	62
3.3.6	控制转移类指令.....	63
3.3.7	控制处理器的指令.....	66
3.3.8	80x86/Pentium 的寻址方式与指令系统.....	66
3.4	汇编语言程序设计.....	67
3.4.1	汇编语言程序设计步骤.....	68
3.4.2	汇编语言简单程序设计.....	71
3.4.3	汇编语言分支程序.....	72
3.4.4	汇编语言循环程序.....	75
3.4.5	汇编语言子程序.....	77
3.4.6	常用系统功能调用和 BIOS.....	80
	习题三.....	83
第 4 章	输入/输出接口.....	86
	本章要点.....	86
4.1	输入/输出接口电路.....	86

4.1.1	输入/输出接口电路的概述.....	86
4.1.2	输入/输出数据传送的方式.....	88
4.2	可编程并行通信接口芯片 8255A.....	91
4.2.1	接口芯片 8255A 的结构.....	91
4.2.2	8255A 的控制字.....	93
4.2.3	8255A 的工作方式.....	96
4.2.4	8255A 的应用.....	98
4.3	可编程串行通信接口芯片 8251A.....	103
4.3.1	串行通信概述.....	103
4.3.2	串行接口标准 RS-232C.....	106
4.3.3	可编程串行通信接口芯片 8251A.....	109
4.3.4	8251A 的控制字和工作方式.....	112
4.4	可编程 DMA 控制器 8237.....	115
4.4.1	8237 内部功能结构.....	116
4.4.2	8237 的工作周期与引脚.....	119
4.4.3	8237 的工作模式及控制字.....	122
4.4.4	DMA 控制器 8237 的编程应用.....	126
	习题四.....	129
第 5 章	存储器及其结构.....	132
	本章要点.....	132
5.1	存储器概述.....	132
5.2	典型半导体存储器芯片.....	140
5.2.1	结构.....	140
5.2.2	存储器芯片的扩展.....	142
5.3	PC 机内存条的选择与安装.....	144
5.4	高速缓存技术.....	146
	习题五.....	150
第 6 章	中断控制系统.....	151
	本章要点.....	151
6.1	中断系统.....	151
6.2	可编程中断控制器 8259A.....	156
6.2.1	8259A 的结构功能.....	156
6.2.2	8259A 的工作方式与控制字.....	159
6.3	可编程中断控制器 8259A 的应用.....	164
	习题六.....	167
第 7 章	可编程计数/定时控制器 8253.....	169
	本章要点.....	169

7.1 可编程计数/定时控制器 8253 的结构	169
7.2 可编程计数/定时控制器 8253 的工作方式及控制字	171
7.3 8253 的应用	176
习题七	180
第 8 章 数/模 (D/A) 及模/数 (A/D) 转换器	182
本章要点	182
8.1 D/A 转换器概述	183
8.2 DAC0832 D/A 转换器	185
8.2.1 D/A 转换器的 0832 功能结构	185
8.2.2 0832 的三种工作方式	187
8.2.3 0832 的应用	187
8.3 A/D 转换器概述	191
8.4 ADC0809 模数 (A/D) 转换器	195
8.4.1 ADC0809 内部功能结构	195
8.4.2 0809 的工作过程	197
8.4.3 ADC0809 的应用	198
习题八	202
第 9 章 微型计算机外部设备简介	204
本章要点	204
9.1 外部设备概述	204
9.2 输入设备	205
9.2.1 键盘及其接口电路	205
9.2.2 鼠标	209
9.2.3 扫描仪	211
9.3 输出设备	213
9.3.1 发光二极管显示器	213
9.3.2 LCD 液晶显示器	215
9.3.3 一般显示器	217
9.3.4 打印机	219
9.4 外部设备驱动程序安装	220
习题九	221
第 10 章 微型计算机开发应用	222
本章要点	222
10.1 微型计算机自动控制系统的开发	222
10.2 双机串行通信的应用设计	224
10.3 数据采集系统的设计	226
10.4 IBM-PC/XT 控制的数据采集系统	229

10.5 十字路口交通灯控制系统	233
习题十	237
附录 A ASCII 码表	238
附录 B 中断向量地址一览表	240
附录 C DOS 系统功能调用 (INT 21H) 表	242
附录 D BIOS 功能调用	248
附录 E 调试程序 DEBUG 的常用命令	252
参考文献	255

第 1 章 微型计算机基础知识

本章要点

- 计算机的码制与数制
- 二进制编码的十进制数（BCD 码）
- 计算机的逻辑器件

1.1 数制

迄今为止，所有的电子计算机都是以二进制形式进行算术运算和逻辑运算的，微型计算机和其他数字电路设备也是如此。我们在操作计算机时，大多是通过键盘输入数据的，键盘上输入的是十进制数字和符号命令，微型计算机必须先把它们转换成二进制形式进行识别、运算和处理。对于运算的结果，微型计算机又将它们还原成十进制数字和符号在输出设备上反映出来。

在计算机中实际经历了从输入码（十进制）→扫描码→ASCII 码（二进制）→输出码（点阵码）的过程。但此过程是由微型计算机自动完成的。为了便于理解计算机的工作原理，这里先对微型计算机中常用的数制和数制间的转换进行讨论。

1.1.1 进位计数制及其基数和权

数制，即进位计数制，是人们利用符号来记数的科学方法。数制有多种形式，如十进制数、二进制数、八进制数等等。习惯上最常用的是十进制记数法。一个任意的十进制数可以表示为：

$$a_n a_{n-1} \dots a_0 . b_1 b_2 b_3 \dots b_{m-1} b_m$$

数制中的 3 个术语：

- 数码：用不同的数字符号来表示一种数制的数值，这种数字符号称为“数码”。
- 基：数制中所允许使用的数码个数称为基。
- 权：数制每一位所具有的值称为权。

1.1.2 计算机中常用的进制数

人们最常用的数是十进制数，计算机中采用的是二进制数，同时有的时候为了简化二进制数据的书写，也采用八进制和十六进制表示方法。下面将分别介绍这几种常用的进制，如表 1-1 所示。

表 1-1 计算机中常用的进制数

进制数	基数	记数原则	备注
十进制数	0,1,2,⋯,8,9	逢十进一，借一当十	人们最常用
二进制数	0,1	逢二进一，借一当二	计算机中采用
八进制数	0,1,2,3,4,5,6,7	逢八进一，借一当八	为简化二进制数据的书写
十六进制数	0,1,⋯,9,A,B,C,D,E,F	逢十六进一，借一当十六	为简化二进制数据的书写

1.1.2.1 二进制表示法

二进制数是基数为 2 的计数制。二进制数主要采用 0, 1 这两个阿拉伯数字作数码。各位的权值为 2^i 。

1. 二进制表示数值方法

$$N = \pm \sum_{i=-m}^n K_i \times 2^i \quad \text{其中: } K_i = 0 \text{ 或 } 1$$

2. 二进制运算规则

二进制数的运算规则为“逢二进一，借一当二”。

加法运算：

$$0+0=0 \quad 0+1=1 \quad 1+0=1 \quad 1+1=10 \text{ (逢二进一)}$$

减法运算：

$$0-0=0 \quad 10-1=1 \text{ (借位)} \quad 1-0=1 \quad 1-1=0$$

乘法运算：

$$0 \times 0 = 0 \quad 0 \times 1 = 0 \quad 1 \times 0 = 0 \quad 1 \times 1 = 1$$

除法运算

$$0/1=0 \quad 1/1=1$$

【例 1-1】二进制数 1011.1 表示如下：

$$(1011.1)_B = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1}$$

1.1.2.2 八进制表示法

八进制数是基数为 8 的计数制。八进制数主要采用 0, 1, 2, 3, 4, 5, 6, 7 这 8 个阿拉伯数字。各位的权值为 8^i 。

1. 八进制表示数值方法

$$N = \pm \sum_{i=-m}^n K_i \times 8^i \quad \text{其中: } K_i = 0, 1, 2, 3, 4, 5, 6, 7$$

2. 八进制运算规则

八进制数的运算规则为“逢八进一，借一当八”。

$$\text{【例 1-2】} (467.6)_O = 4 \times 8^2 + 6 \times 8^1 + 7 \times 8^0 + 6 \times 8^{-1}$$

1.1.2.3 十六进制表示法

十六进制数是基数为 16 的计数制。十六进制数主要采用 0, 1, ..., 9, A, B, C, D, E, F 这 16 个符号来表示数值，各位的权值为 16^i 。

1. 十六进制表示数值方法

$$N = \pm \sum_{i=-m}^n K_i \times 16^i \quad \text{其中: } K_i = 0, \dots, 9, A, \dots, F$$

2. 十六进制运算规则

十六进制数的运算规则为“十六进一，借一当十六”。

$$\text{【例 1-3】} (56D.3)_H = 5 \times 16^2 + 6 \times 16^1 + 13 \times 16^0 + 3 \times 16^{-1}$$

1.1.3 进制间的转换

由于计算机内采用二进制,人使用十进制数,为了数据的表示,平时又常用到十六进制数和八进制数,这就需要掌握各种进制间的相互转换关系。下面介绍各数制之间的转换方法。

1.3.1.1 二进制数和十进制数之间的转换

1. 二进制数转换为十进制数

方法:按二进制数的位权进行展开相加即可。

【例 1-4】11101.101

$$=1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\ =16+8+4+0+1+0.5+0.25+0.125=29.875$$

2. 十进制数转换为二进制数

方法:将整数部分和小数部分分别进行转换,然后再把转换结果进行相加。

- 整数转换采用除2取余法:用2不断地去除要转换的数,直到商为0。再将每一步所得的余数,按逆序排列,便可得转换结果。
- 小数转换采用乘2取整法:每次用2与小数部分相乘,取乘积的整数部分,再取其小数部分乘2直到小数部分为0。将所取整数顺序放在小数点后即为转换结果。

【例 1-5】将 $(137)_{10}$ 转换为二进制数。

2	137	余数(结果)	低位
2	68	1	↓ 高位
2	34	0	
2	17	0	
2	8	1	
2	4	0	
2	2	0	
2	1	0	
2	0	1	

转换结果: $(137)_{10} = (10001001)_2$

【例 1-6】将 $(0.625)_{10}$ 转换为二进制数。

$$0.625 \times 2 \longrightarrow 1.25 \times 2 \longrightarrow 0.5 \times 2 \longrightarrow 1.0$$

取整: 高位 低位

转换结果: $(0.625)_{10} = (0.101)_2$

1.1.3.2 二进制数和八进制数、十六进制数间的转换

1. 二进制数到八进制数、十六进制数的转换

方法:

- 二进制数到八进制数转换采用“三位化一位”的方法。从小数点开始向两边分别进行,每三位分一组,向左不足三位的,从左边补0;向右不足三位的,从右边补0。
- 二进制数到十六进制数的转换采用“四位化一位”的方法。从小数点开始向两边分别进行,每四位分一组,向左不足四位的,从左边补0;向右不足四位的,从右边补0。

【例 1-7】将 $(1000110.01)_2$ 转换为八进制数和十六进制数。

1) 二进制数到八进制数的转换:

$$\begin{array}{ccccccc} 1 & 000 & 110 & .01 & \longrightarrow & 001 & 000 & 110 & . & 010 \\ (1000110.01)_B = & (1 & 0 & 6 & . & 2)_O \end{array}$$

2) 二进制数到十六进制数的转换:

$$\begin{array}{ccccccc} 100 & 0110 & .01 & \longrightarrow & 0100 & 0110 & .0100 \\ (1000110.01)_B = & (4 & 6 & . & 4)_H \end{array}$$

2. 八进制、十六进制数到二进制数的转换

方法: 采用“一位化三位(四位)”的方法。按顺序写出每位八进制(十六进制)数对应的二进制数, 所得结果即为相应的二进制数。

【例 1-8】将(352.6)₀转换为二进制数。

$$\begin{array}{ccc} \frac{3}{011} & \frac{5}{101} & \frac{2}{010} & . & \frac{6}{110} \\ & & & & = (11\ 101\ 010.11)_B \end{array}$$

1.1.3.3 几种常见数制间的转换方法

几种常见数制间的转换方法如图 1-1 所示。各数制间的关系如表 1-2 所示。

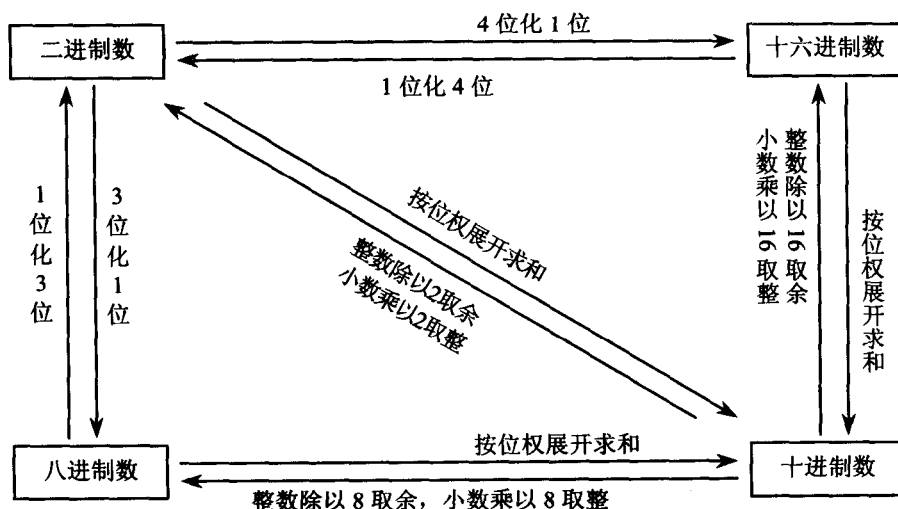


图 1-1 数制间的转换关系

表 1-2 各种数制对照表

二进制	十六进制	十进制	八进制	二进制	十六进制	十进制	八进制
0000	0	0	0	1000	8	8	10
0001	1	1	1	1001	9	9	11
0010	2	2	2	1010	A	10	12
0011	3	3	3	1011	B	11	13
0100	4	4	4	1100	C	12	14
0101	5	5	5	1101	D	13	15
0110	6	6	6	1110	E	14	16
0111	7	7	7	1111	F	15	17

1.2 数据在计算机中的表示

1.2.1 机器数与真值

数值数据可分为无符号数和有符号数两种。无符号数指所有的数位都用来表示数据的数值，没有符号位，为了方便，通常将它作为正数处理。有符号数分正、负两种，为使计算机能够表示和识别，把符号也数字化，一般规定用“0”表示正号“+”，用“1”表示负号“-”，这正好是二进制中的两个不同数码，符号位放在数的最高位（即最左边的一位）。一个数在机器中的表示形式（数值数据在计算机中的编码）称为机器数，而它的数值（即用“+”、“-”表示符号再加上绝对值）叫做机器数的真值。例如真值 $X_1 = -1011\text{B}$ ，其对应的机器数为 11011B ；真值 $X_2 = -0.1101\text{B}$ ，其机器数为 1.1101B （对于小数，小数点左边的1为符号位，不是数值的数字）。

带符号的二进制数在机器中的表示方法通常有：原码、补码、反码三种。

1.2.1.1 原码表示法

原码是最简单的一种表示法，用最高位表示符号，其中：0——正、1——负，其他位表示数值的绝对值，它又称为带符号的绝对值或称为符号绝对值表示法。

定义：

$$[X]_{\text{原}} = \begin{cases} X & 0 \leq X \leq 1 \quad \text{定点小数} \\ 1 + |X| & -1 < X \leq 0 \end{cases}$$

$$[X]_{\text{原}} = \begin{cases} X & 0 \leq X \leq 2^n - 1 \quad \text{定点整数} \\ 2^{n+1} + |X| & -(2^n - 1) \leq X \leq 0 \end{cases}$$

【例 1-9】求 $X_1 = 0.1011$ ， $X_2 = -0.1011$ 的原码表示。（8 位）

$$[X_1]_{\text{原}} = X_1 = 01011000$$

$$[X_2]_{\text{原}} = 1 + |X_2| = 11011000$$

↑ 小数点位置

【例 1-10】求 $X_1 = 1011$ ， $X_2 = -1011$ 的原码。（8 位）

$$[X_1]_{\text{原}} = 00001011$$

$$[X_2]_{\text{原}} = 10001011$$

↑ 小数点位置

【例 1-11】0 的表示形式。（8 位）

$$[+0]_{\text{原}} = 00000000$$

$$[-0]_{\text{原}} = 10000000$$

特点：

- 原码与真值的对应关系简单。
- 0 的编码不惟一，处理运算不方便。

1.2.1.2 补码表示法

采用原码表示法简单易懂，但它的缺点是加减运算复杂。设计补码表示法的目的有两个：一是使符号位能和有效数值部分一起参加数值运算从而简化运算规则，节省运算时间；二是使减法运算转化为加法运算，从而进一步简化计算机中运算器的线路设计。

1. 模的概念

把一个计量单位称之为模或模数，用 M 表示。例如，时钟是以 12 进制进行计数循环的，即以 12 为模。在时钟上，时钟加上（正拨）12 的整数位或减去（反拨）12 的整数位，时钟的位置不变。从 0 点出发逆时针拨 10 格即减去 10h，也可看成从 0 点出发顺时针拨 2 格（加上 2h），即 2 点（ $0-10=-10=-10+12=2$ ）。因此，在模 12 的前提下，-10 可映射为 +2。

计算机的运算部件与寄存器都有一定字长的限制，因此它的运算也是一种模运算。如一定位数的计数器，在计满后会发生溢出，又从头开始计数。产生溢出的量就是计数器的模。如 8 位的二进制数，它的模数为 $2^8=256$ 。

由此可看出，对于一个模数为 12 的循环计数来说，加 2 和减 10 的效果是一样的；因此，在以 12 为模的系统中，凡是减 10 的运算都可以用加 2 来代替，这就使减法问题转化成加法问题了。10 和 2 对模 12 而言互为补数。在计算机中称为“补码”。

2. 补码的表示

在补码表示法中，正数的补码和原码相同。负数的补码则是符号位为“1”，数值部分按位取反后再在末位（最低位）加 1。

【例 1-12】求 0.1011 和 -0.1011 的补码。（8 位）

$$[0.1011]_{\text{补}}=[0.1011]_{\text{原}}=01011000$$

$$[-0.1011]_{\text{补}}=[-1011000]_{\text{反}}+1=10100111+1=10101000$$

$$[0]_{\text{补}}=[+0]_{\text{补}}=[-0]_{\text{补}}=00000000$$

特点：

- 采用补码后，可以方便地将减法运算转化为加法运算，运算过程得到简化。
- 数值 0 的补码只有一个，即 $[0]_{\text{补}}=00000000\text{B}=00\text{H}$ 。

1.2.1.3 反码表示法

反码：最高一位表示符号，数值位是对负数取反。注意数 0：

$$[+0]_{\text{反}}=00000000$$

$$[-0]_{\text{反}}=11111111$$

$$[+1100111]_{\text{反}}=01100111$$

$$[-1100111]_{\text{反}}=10011000$$

特点：

- 通过反码表示法可以获得将负数转换成补码的简便方法。
- 正 0 和负 0 的反码表示是不同的。

1.2.2 数的定点与浮点表示

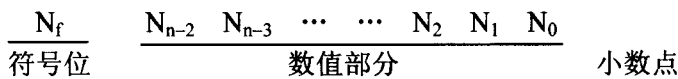
在计算机中，用二进制数表示数值数据小数点的方法有两种，即定点法和浮点法。定点法，约定小数点的位置是固定不变的，这样的数称为定点数；浮点法，指小数点的位置不是固定的，而是浮动的，这样的数称为浮点数。

1.2.2.1 定点数表示法

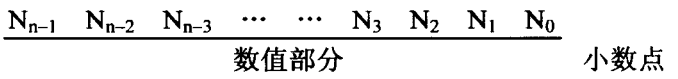
由于小数点约定在固定位置，小数点就不再使用记号“.”，计算机也无须为它设置专门的硬件设备。原则上讲，小数点位置固定哪一位都可以，但通常采用纯小数或纯整数。所以计算机可以处理的定点数是定点整数和定点小数。

1. 定点整数

(1) 带符号整数：某个 N 位二进制数，其最高位为符号位，其他 $N-1$ 位为数值部分：

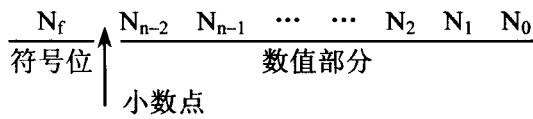


(2) 无符号整数：所有的数位都用来表示数值。



2. 定点小数

用最高位表示符号，其他 N-1 位表示数值部分，将小数点定在数值部分的最高位左边。

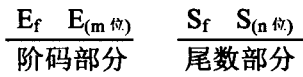


1.2.2.2 浮点数表示

浮点数的表数范围与基数有关，也与阶码和尾数的码制（即符号数的表示方法）及位数有关。通常一个二进制数 N 总可以写成下面的形式：

$$N = \pm S * R^{\pm E}$$

在计算机内，存储的格式：



其中：E_f：阶码，表示阶码的符号，E：阶码，指出小数点的位置，S_f：数码，数值的符号位，S：尾数，决定数值的精度。

1.2.3 计算机中常用的编码

1.2.3.1 BCD 码

在将人们习惯使用的十进制数转换成计算机内部的二进制数时，除了可以通过上面介绍的方法把十进制数换算成二进制数之外，还可以用二进制数表示十进制数，保留各位之间“逢 10 进 1”的关系，BCD 码是一种用 4 位二进制数字来表示一位十进制数字的编码，也称为二进制编码表示的十进制数（Binary Code Decimal），简称 BCD 码。表 1-3 给出了十进制数 0~15 的 BCD 码。

表 1-3 十进制与 8421BCD 码对照表

十进制	BCD 码	十进制	BCD 码
0	0000	8	1000
1	0001	9	1001
2	0010	10	00010000
3	0011	11	00010001
4	0100	12	00010010
5	0101	13	00010011
6	0110	14	00010100
7	0111	15	00010101

BCD 码有两种格式：

(1) 压缩 BCD 码格式 (Packed BCD Format)。用 4 个二进制位表示一个十进制位，就是用 0000B~1001B 来表示十进制数 0~9。例如：十进制数 4256 的压缩 BCD 码表示为：0100 0010 0101 0110 B

(2) 非压缩 BCD 码格式 (Unpacked BCD Format)。用 8 个二进制位表示一个十进制位，其中，高 4 位无意义，我们一般用 xxxx 表示，低 4 位和压缩 BCD 码相同（注：上表即为非压缩 BCD 码格式）。

【例 1-13】十进制数 4256 的非压缩 BCD 码表示为：

xxxx0100 xxxx0010 xxxx0101 xxxx0110 B

1.2.3.2 ASCII 码

常用的编码方式美国标准信息交换代码 (American Standard Card for Information Interchange) 简称 ASCII 码。

这种编码由 7 位二进制数码 ($b_6b_5b_4b_3b_2b_1b_0$) 编码组成，其中 b_6 为最高位， b_0 为最低位。共有 128 个编码，包括大写、小写英文字母，十进制数字，标点符号和控制符号。其中前 32 个代码是控制字符，编程人员利用控制字符可以控制机器进行某种操作。例如往打印机接口电路送代码 OAH (代表控制字符 LF，指换行操作)，则打印机的纸就向前走一行。

表 1-4 列出了 ASCII 7 位编码字符集的一部分，从中可看出 ASCII 码字符的排列有一定的规律。如数字 0~9 的编码是 0110000~0111001，它们的高 3 位均是 011，低 4 位正好与其对应数值的二进制数相符，也就是该数字所对应的 8421BCD 码，因而由 ASCII 码的数字字符转换成十进制数字的 8421BCD 码非常容易，只要把数字字符码减去 30H，并把高 4 位的 0 去掉，就是所求的 8421BCD 码。

表 1-4 ASCII 编码

$b_6b_5b_4 \backslash b_3b_2b_1b_0$	011	100	101
0000	0	@	P
0001	1	A	Q
0010	2	B	R
0011	3	C	S
0100	4	D	T
0101	5	E	U
0110	6	F	V
0111	7	G	W
1000	8	H	X
1001	9	I	Y
1010	:	J	Z
1011	:	K	[
1100	<	L	\
1101	=	M	]
1110	>	N	↑
1111	?	O	-