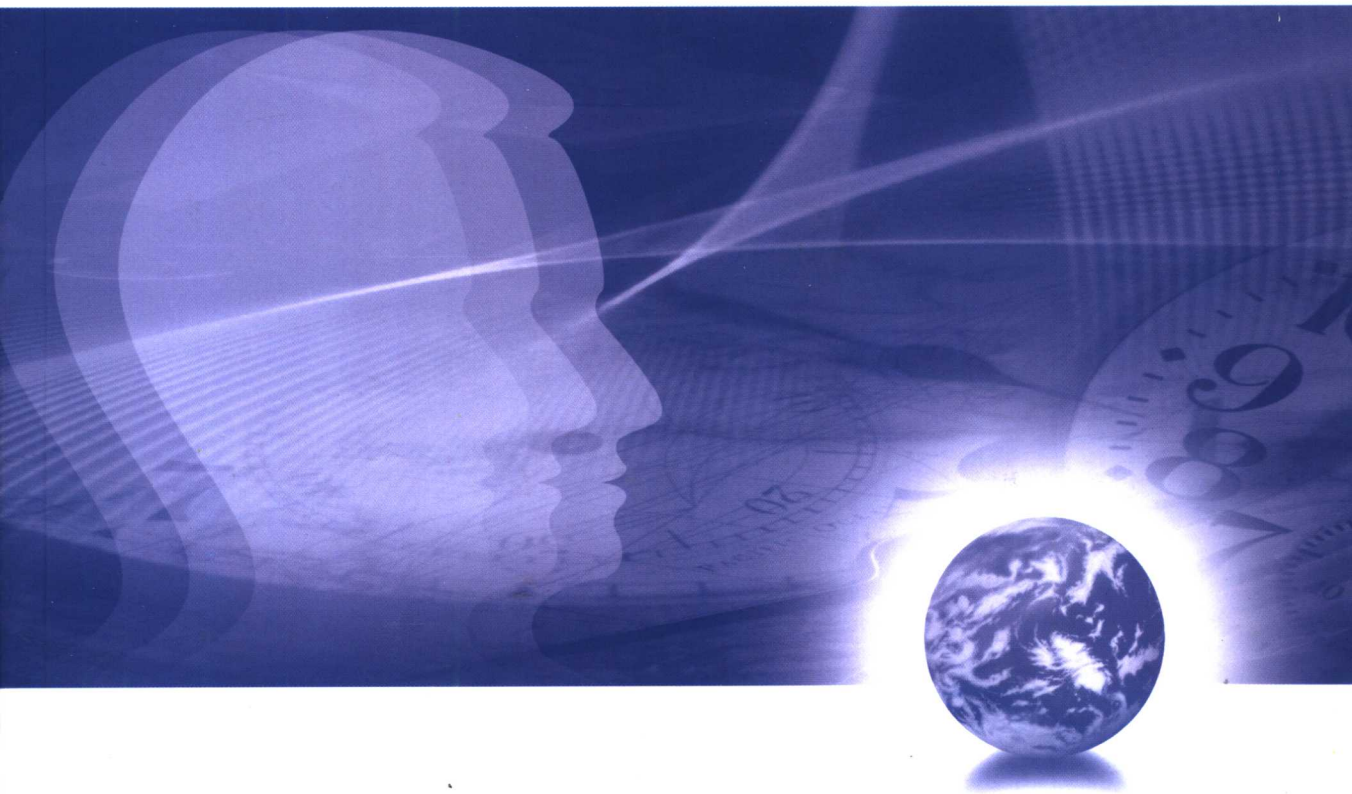
高等院校计算机科学与技术
“十五”规划教材

单片机原理及接口技术



胡　健　主编
刘玉宾　朱焕立　等编著

机械工业出版社
CHINA MACHINE PRESS

高等院校计算机科学与技术“十五”规划教材

单片机原理及接口技术

胡 健 主编

刘玉宾 朱焕立 等编著

机械工业出版社

本书详尽介绍了 MCS-51 系列单片机的硬件结构、指令系统、程序设计、中断系统、系统扩展、接口技术及应用等内容。在内容安排上坚持改革与创新相结合，强调实践性和应用性，重在方法的介绍，旨在锻炼学生的综合、分析能力和基本技能的培养。

本书所有例题均上机调试通过。

本书语言通俗、结构紧凑，具有一定的系统性和实用性，可作为高等院校电类、机类、机电类及计算机类各专业的单片机教材，也可作为单片机爱好者的自学教材。

图书在版编目（CIP）数据

单片机原理及接口技术 / 胡健主编. －北京：机械工业出版社，2004.10
（高等院校计算机科学与技术“十五”规划教材）
ISBN 7-111-15552-1

I. 单… II. 胡… III. ①单片微型计算机—理论—高等学校—教材②单片微型计算机—接口—高等学校—教材 IV. TP368.1

中国版本图书馆 CIP 数据核字（2004）第 112811 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
责任编辑：丁 诚
责任印制：石 冉
三河市宏达印刷有限公司印刷 · 新华书店北京发行所发行
2005 年 1 月第 1 版 · 第 1 次印刷
787mm×1092mm 1/16 · 15.5 印张 · 379 千字
0 001-5000 册
定价：22.00 元

出 版 说 明

信息技术高度普及的今天，具备一定层次的信息技术素养成为社会素质教育的一个重要目标，由此对高等院校的计算机专业教育提出了更高更新的要求。教育水平提高的关键是教学质量，那么对教学质量有直接影响的教材建设就成为了计算机专业教育的重中之重。

适逢高等院校计算机专业教育改革的关键时期，为配合相关的教材建设，机械工业出版社会同全国在该领域内享誉盛名、具备雄厚师资和技术力量的高等院校，包括清华大学、上海交通大学、南京大学、成都电子科技大学、东南大学、西安电子科技大学、解放军理工大学、北京科技大学等重点名校，组织了多位长期从事教学工作的骨干教师，集思广益，对当前高等院校的教学现状开展了广泛而深入的研讨，继而紧密结合当前技术发展需要并针对教学改革所提出的问题，精心编写了这套面向普通高等院校计算机专业的系列教材，并陆续出版。

本套教材内容覆盖了普通高等院校计算机专业学生的必修课程，另外还恰如其分地添加了一些选修课程，总体上分为基础、软件、硬件、网络和多媒体五大类。在编写过程中，对教学改革力度比较大、内容新颖以及各院校急需的并且适应社会经济发展的新教材，优先选择出版。

本套教材注重系统性、普及性和实用性，力求达到专业基础课教材概念清晰、深度合理的标准，并且注意与专业课教学的衔接；专业课教材覆盖面广、深浅适中，在体现相关领域最新发展的同时注重理论联系实际。全套教材体现了教育改革的最新思想，可作为高等院校计算机科学与技术专业的教学用书，同时也是培训班和自学使用的最佳教材。

机械工业出版社

前　言

单片机技术是一门应用性很强的专业课，其理论与实践技能是从事电类、机类、机电类及计算机类专业技术工作的人员所不可缺少的。理论与实践的密切结合，是本课程的重要特点。多年来，我们在单片机技术课程的教学中，一直采用教、学、做相结合的教学模式，效果令人满意。由于计算机技术发展很快，社会迫切要求相关专业学生掌握新技术、新器件的应用。为了适应我国高等教育发展的需要，我们编写了本书，以满足自动化、电子工程、信息技术及机电技术类各专业教学要求。

本书的配套教材《单片机原理及接口技术实践教程》可帮助师生完成实践环节的训练。

本教材主要以当前仍处于主流地位的 MCS-51 单片机为主进行介绍，编写中尽量做到能反映当今单片机应用的最新技术，本书不仅注重学生在应用方面的训练，同时又重视基本知识的讲解。

本书语言通俗、结构紧凑，具有一定的系统性和实用性，可作为电类、机类、机电类及计算机类各专业的单片机教材，也可作为单片机爱好者的自学教材。

本书由胡健主编，刘玉宾、朱焕立、刘瑞新、孔昭平、杜广朝、万书栋、李小雄、何瑞、吴丽、范文军、郭红山、王留奎、赵转丽、李开先、商坤等编写，刘瑞新审。本书在编写过程中得到了许多同行的帮助和支持，提出了许多宝贵意见和建议，在此向他们表示感谢。

由于编者水平有限，加上时间仓促，书中难免有错误与不妥之处，请广大读者批评指正。

编　者

2004 年 6 月

目　录

第 1 章　计算机基础知识

本章主要讲述计算机的基础知识，包括各种数制及其相互转换、带符号二进制数的编码及其运算、二-十进制编码、ASCII 码；单片机的概念、发展状况、应用领域及嵌入式系统的概念。

1.1　数制与编码

计算机在处理数字和字符时，需要解决三个问题：一是数制及其转换问题；二是带符号数与无符号数的表示方法及其运算问题；三是数字和字符的表示，即编码问题。

1.1.1　数制及其转换

数制就是计数方式，按照进位方式计数的数制叫进位计数制。在日常生活中，人们常用十进制数，它的特点是逢十进一。此外，还有七进制，如七天为一星期，十二进制，如十二个月为一年等等。在计算机内部，一切信息（包括数值、字符、指令等）的存放、处理和传送均采用二进制的形式。但是在输入输出或书写时，为了用户的方便，经常用到八进制和十六进制。

1．进位制中数的表示

（1）十进制数

十进制数由 0～9，共十个不同的数字符号组成，且自左向右，由高位到低位依次排列。我们把字符个数“10”称为十进制数的基数，即十进制数就是以 10 为基数的计数体制。

每个数字符号处在不同数位所代表的数值是不同的。例如，数字 3 在个位数位置上时表示 3，即 3×10^0；在十位数位置上时表示 30，即 3×10^1；在百位数位置上时表示 300，即 3×10^2；在小数点后第一位则表示 0.3，即 3×10^{-1}。可见每个数码所表示的数值等于该数码乘以一个与数码所在位置有关的常数，这个常数叫做位权。位权的大小是以基数为底、数码所在位置的序号为指数的整数次幂。以十进制数为例，其个位数位置上的位权为 10^0；十位数位置上的位权为 10^1；百位数位置上的位权为 10^2；…；小数点后第一位的位权为 10^{-1}；小数点后第二位的位权为 10^{-2}；…。例如，十进制数 573.46 可以表示成：

$$573.46=5\times10^2+7\times10^1+3\times10^0+4\times10^{-1}+6\times10^{-2}$$

相邻位的权值关系是：左边的位权是相邻右边位权的 10 倍。

根据以上分析，可以将任意一个十进制数表示为：

$$(N)_{10}=\pm\sum_{i=m}^{n}K_i\times10^i$$

式中，K_i 为基数 10 的 i 次幂的系数，它可为 0～9 中任一个数字。n、m 可为$-\infty\sim+\infty$之间的任一整数。

一般而言，对于任意 R 进制数，可以表示为以下和式：

$$(N)_R = \pm\sum_{i=m}^{n} K_i \times R^i$$

式中，K_i为基数 R 的 i 次幂的系数，它可为 0～R–1 中任意一个数字。n、m 可为–∞～+∞之间的任一整数。

（2）二进制数

二进制数只有“0”和“1”两个数字符号，所以其基数为“2”。它同十进制数一样，自左向右作高位到低位的排列，计数时按“逢二进一”的原则进行计数。

根据位权表示法，每一个数字符号在不同的位置上具有不同的值。例如：

$$(10111)_2=1\times2^4+0\times2^3+1\times2^2+1\times2^1+1\times2^0=(23)_{10}$$

$$(110.101)_2=1\times2^2+1\times2^1+0\times2^0+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}=(6.625)_{10}$$

任意二进制数可表示为：

$$(N)_2 = \pm\sum_{i=m}^{n} K_i \times 2^i$$

其中，K_i为基数 2 的 i 次幂的系数，它可以为 0，也可以为 1。n、m 可为–∞～+∞之间的任一整数。

由于二进制数只有两个数字符号，因此很容易用电路元器件的状态来表示，这些器件具有两种不同的稳定状态（如电平的高和低、晶体管的导通和截止等）且能互相转换，既简单又可靠。

在计算机内，二进制的位（bit）是数据的最小单位，如 111 是 3 位，11101 为 5 位。通常，计算机中将 8 位二进制数编为一组叫做一个字节（Byte），作为数据处理的基本单位。Byte 也可以简写为大写的英文字母“B”。1024 个字节称为 1KB，1024KB 称为 1MB，1024MB 称为 1GB，1024GB 称为 1TB。计算机存储器的容量就是用字节来计算和表示的。例如，内存容量为 256MB，即表示内存容量是 256 兆字节。

计算机中也经常用字（Word）表示数据或信息的长度，一个字由若干字节组成。通常将组成一个字的位数叫做该字的字长，即字长是指以多少位二进制位表示一个二进制数。例如，一个字由 4 个字节（即 32 位）组成，则该字字长为 32 位。一个字可以用来存放一条指令或一个数据，不同的计算机系统其字长是不同的。较长的字长可以处理数位较多的信息，字长是衡量计算机功能的一个重要的指标。

（3）十六进制

十六进制数使用 0～9、A～F，共 16 个字符，所以其基数为“16”。其中 0～9 与十进制数 0～9 相同，A～F 分别表示十进制的 10～15。十六进制的计数原则是“逢十六进一，借一当十六”。

每一个数字符号处在不同数位代表不同的数值，例如十六进制数 7AE.B5 可表示成：

$$(7AE.B5)_{16}=7\times16^2+10\times16^1+14\times16^0+11\times16^{-1}+5\times16^{-2}=(1966.707)_{10}$$

任意十六进制数可表示为：

$$(N)_{16} = \pm\sum_{i=m}^{n} K_i \times 16^i$$

其中，K_i为基数 16 的 i 次幂的系数，它可为 0～F 十六个数字中任一个。n、m 可为：–∞～

+∞之间的任一整数。

为了便于区分不同进制的数字，常在数字后面跟一后缀：二进制数用“B”表示，如 1101B；十进制数用“D”表示，如 1202D，十进制数字也可省略后缀；十六进制数用“H”表示，如 23A4H。

2．不同数制之间的转换

尽管有不同的进制，但在计算机中的数只能用二进制表示，十六进制适合于读写方便需要，而十进制则是日常生活所必需的。为了方便各种应用场合之间的接口，因此必须知道各种进制之间的转换方法。

（1）二进制转换成十进制

任何二进制数均可用按权展开求和的方法转化成十进制数。

【例 1-1】 将二进制数 11010.011B 转换成十进制数。

解：$11010B=1\times2^4+1\times2^3+0\times2^2+1\times2^1+0\times2^0=26D$

【例 1-2】 将二进制数 101.111B 转换成十进制数。

解：$101.111B=1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-1}+1\times2^{-3}=5.875D$

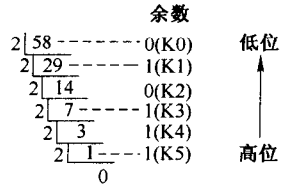
（2）十进制转换成二进制

在将一个十进制数转换成二进制数时，需要将整数部分和小数部分分别进行转换。

十进制整数转换成二进制整数采用“除 2 取余法”。具体做法为：将十进制数除以 2，得到一个商和余数；再将商除以 2，又得到一个商和余数；继续这个过程，直到商等于 0 为止。每次所得的余数（必定是 0 或 1）就是对应二进制数的各位数字。值得注意的是，第一次得到的余数为二进制数的最低位，最后一次得到的余数为二进制数的最高位。

【例 1-3】 将十进制数 58 转换成二进制数。

解：



所以，58D=111010B。

十进制小数转换成二进制小数采用“乘二取整法”。具体做法为：用 2 乘以十进制纯小数，在得到的积中取出整数部分；再用 2 乘余下的纯小数部分，在得到的积中再取出整数部分；继续这个过程，直到余下的纯小数为 0 或满足所要求的精度为止。最后将每次取出的整数部分（必定是 1 或 0）按先后从左到右排列即得到对应的二进制小数。

【例 1-4】 将十进制小数 0.6875 转换成二进制小数。

解：

	取整	
$0.6875\times2=1.3750$	1	(K1 = 1)
$0.3750\times2=0.7500$	0	(K2 = 0)
$0.7500\times2=1.5000$	1	(K3 = 1)
$0.5000\times2=1.0000$	1	(K4 = 1)

至最后余下的纯小数为 0，则结束。所以：0.6875D=0.1011B。

需要指出，一个二进制小数能够准确地转换成十进制小数，但一个十进制小数不一定能完全准确地转换成二进制小数。

【例 1-5】 将十进制小数 0.1 转换成二进制小数。

解：

	取整	
0.1×2 = 0.2	0	(K1 = 0)
0.2×2 = 0.4	0	(K2 = 0)
0.4×2 = 0.8	0	(K3 = 0)
0.8×2 = 1.6	1	(K4 = 1)
0.6×2 = 1.2	1	(K5 = 1)

这个乘 2 的过程将是无限的。

如果小数点后取 4 位，则 0.1D≈0.0001B=0.0625D

如果小数点后取 5 位，则 0.1D≈0.00011B=0.09375D

由此可见，当十进制小数不能完全准确地转换成二进制小数时，转换后二进制小数取的位数越多，越接近十进制小数。在这种情况下，可以根据精度要求转换到小数点后某一位为止。

（3）二进制转换成十六进制

由于十六进制的基数为 16，而 $16=2^4$，因此四位二进制数就相当于一位十六进制数。若将多位二进制数转化为十六进制数，则其整数部分可由小数点起向左推，四位成一组，直至分完最高位，若此时不足四位，可在高位补零；其小数部分可由小数点起向右推，四位组成一组，直至分完最低位，若此时不足四位，可在低位补零。完成上述分组后，将每组的四位二进制数用一位十六进制数表示，就完成了二进制数到十六进制数的转换。

【例 1-6】 将二进制数 1011011.101101B 转换为十六进制数的形式。

解：先将 1011011.101101B 分组：0101 1011. 1011 0100B

然后将各组四位二进制数化成十六进制数，所以：1011011.101101B=5B.B4H

（4）十六进制转换成二进制

由于一位十六进制数相当于四位二进制数，因此要将一个十六进制数转换成二进制数，只需将每一位十六进制数用四位二进制数对应表示即可。

【例 1-7】 将十六进制数 7E.6AH 转换为二进制数。

解：分别将 7、E、6、A 化成相应的二进制数：0111、1110、0110、1010

再按位数的高低依次排列，就可得相应的二进制数：7E6AH=1111110.0110101B

（5）十六进制转换成十进制

可按权展开相加法将十六进制数转换为十进制数。

【例 1-8】 将十六进制数 7A.5CH 转换为十进制数。

解：$7A.5CH=7\times16^1+10\times16^0+5\times16^{-1}+12\times16^{-2}=122.35938D$

（6）十进制转换成十六进制

可将十进制数用前面介绍的方法先转换成二进制数，再将二进制数转换成十六进制数。

【例 1-9】 将十进制数 18.625D 转换成十六进制数。

解：先将 18D 化成二进制数：18D=10010B

再将纯小数 0.625D 化成二进制小数：0.625D=0.101B

将整数与小数合并：18.625D=10010.101B

然后把所得的二进制数再转化成十六进制：10010.101B=12.AH。

即：18.625D=12.AH

当然也可由十进制数的整数部分直接除以 16 取余，纯小数部分不断乘以 16 取整来转换，但过程比较麻烦，请同学们自己练习。

（7）其他进位制数之间的相互转换

前面介绍了几种最常用的进位制之间的相互转换的方法，利用这些方法也可以推出其他进位制之间的相互转换方法。

十进制向任意 R 进制转换的方法是：整数部分采用除 R 取余法；小数部分采用乘 R 取整法，然后将获得的 R 进制整数部分和小数部分相加。

R 进制向十进制转换的方法为：按权展开相加即可。

1.1.2 二进制数的编码

1. 机器数

前面在讲二进制数时，没有提到符号问题，实际上是一种无符号数的表示。但在计算机中，数显然是有正、负之分的。通常一个有符号数的最高位为符号位（0 表示正数，1 表示负数），即数的符号在机器中也数码化了。我们把一个数在计算机中的表示形式叫做机器数，而这个数本身就称为这个机器数的真值。

例如：假设两个数 N_1 和 N_2 的真值分别为：N_1= +100 1011B；N_2= −101 0100B

则所对应的机器数分别为：N_1 = 0100 1011B；N_2 = 1101 0100B

一个有符号数，由于编码的不同，可有几种机器数。反之一个机器数，由于解释方法不同，也可代表几种真值。下面要介绍的原码、反码和补码就是常用的几种机器数的表示形式。

2. 原码

如上所述，正数的符号位用“0”表示，负数的符号位用“1”表示，符号位之后表示数值的大小，这种表示方法称为原码。

例如：x = +1110010B 和 y = −1110010B 的原码可分别表示为：

$$[x]_{原}=01110010B$$

$$[y]_{原}=11110010B$$

原码的表示方法很简单，缺点是“0”的原码有两种形式：

$$[+0]_{原}=00000000B$$

$$[-0]_{原}=10000000B$$

3. 反码

正数的反码等于原码；负数的反码其最高位（符号位）为“1”，数值位等于原码数值位按位求反（0 变为 1，1 变为 0）。

【例 1-10】 求 x = +0101000B 和 y = −0101000B 的原码和反码。

解：$[x]_{原}$=00101000B，$[x]_{反}$=00101000B

$[y]_{原}$=10101000B，$[y]_{反}$=11010111B

“0”的反码也有两种表示方法：

$$[+0]_{反}=00000000B$$

$$[-0]_{反}=11111111B$$

8 位带符号二进制反码可表示的数值范围为：

10000000B(−127D)～01111111B(+127D)

4．补码

正数的补码等于原码；负数的补码其最高位（符号位）为“1”，数值位等于反码数值位的低位加“1”。

【例 1-11】 求 x = +0101000B 和 y =−0101000B 的补码。

解：$[x]_{原}$=00101000B，$[x]_{反}$=00101000B，$[x]_{补}$=00101000B

$[y]_{原}$=10101000B，$[y]_{反}$=11010111B，$[y]_{补}$=11011000B

8 位带符号二进制补码可表示的数值范围为：

10000000B(−128D)～01111111B(+127D)

“0”的补码是惟一的，在 8 位二进制数中：

$$[+0]_{补}=[-0]_{补}=00000000B$$

其实在日常生活中也有求补的例子。如钟表，若当前时间为 10 点，而现在表的指示时间为 8 点，要调整时间，可向前（顺时针）拨 2 小时使时针指到 10 点，也可倒拨（逆时针）10 个小时使时针指到 10 点。这里顺时针拨（+2）与逆时针拨（−10）对 12 互为补数，12 称为模。要求某一个数的补数可用模减去该数即可。对于钟表，如 2 的补数为：

$$[2]_{补}=12-2=10$$

同理，8 位二进制数的模就是其所能表示的最大数加 1，即 256，表示成十六进制形式就是 100H。要求一个 8 位二进制数（负数）的补码就可以用 100H 减去该数。

【例 1-12】 用模求−64H 的补码。

解：$[-64H]_{补}$=100H−64H=9CH

另外，一个数的补码的补码等于原码。

1.1.3 带符号二进制数的运算

原码虽然比较简单、直观，但采用原码进行加、减运算时，要求计算机的运算电路非常复杂；如果采用补码，就可以把减法运算变成加法运算，省略了减法器，使硬件电路大大简化。在单片机中带符号二进制数都以补码的形式出现，所以下面着重讨论 8 位二进制补码的加减运算。

补码加法运算遵循的公式为：

$$[x+y]_{补}=[x]_{补}+[y]_{补}$$

$$[x-y]_{补}=[x]_{补}+[-y]_{补}$$

其运算规则是：

- 按二进制运算规则运算，即逢二进一；
- 两个数的符号位参加运算；
- 由符号位产生的进位自然舍掉。

【例 1-13】 设 x = +24D，y =10D，z = −100D，求：$[x+y]_{补}$和$[x-y]_{补}$。

解：$[x]_{补}$=0001 1000B，$[y]_{补}$=0000 1010B，$[-y]_{补}$=1111 0110B，$[z]_{补}$=1001 1100B

所以：

① $[x+y]_{补}=[x]_{补}+[y]_{补}$=0001 1000B+0000 1010B=0010 0010B，计算过程：

$$\begin{array}{rr} 0001 & 1000B \\ +\ 0000 & 1010B \\ \hline 0010 & 1010B \end{array}$$

$[x+y]_{补}$的原码为：$[[x+y]_{补}]_{补}=[x+y]_{补}$=00100010B，对应真值为 34，结果正确。

② $[x-y]_{补}=[x]_{补}+[-y]_{补}$=0001 1000B+1111 0110B=0000 1110B，计算过程：

$$\begin{array}{rrr} & 0001 & 1000B \\ + & 1111 & 0110B \\ \hline 1 & 0000 & 1110B \end{array}$$

自然丢失

$[x-y]_{补}$的原码为：$[[x-y]_{补}]_{补}=[x-y]_{补}$=00001110B，对应真值为 14，结果正确。

$[x+z]_{补}=[x]_{补}+[z]_{补}$=0001 1000B+1001 1100B=1011 0100B，计算过程：

$$\begin{array}{rr} 0001 & 1000B \\ +\ 1001 & 1100B \\ \hline 1011 & 0100B \end{array}$$

$[x+z]_{补}$的原码为：$[[x+z]_{补}]_{补}=[1011\ 0100B]_{补}$=1100 1100B，对应真值为–76，结果正确。

1.1.4 二–十进制编码

在计算机中，由于所有的字符、数字都以二进制形式出现，所以要表示十进制数必须先将十进制数表示成二进制形式。将十进制数表示成二进制形式的方法有两种：一是直接转换成二进制数，二是用特殊的二进制代码，即 BCD 编码表示。

BCD（Binary Coded Decimal）编码具有二进制数的形式，却又有十进制数的特点，它是一种二进制编码的十进制数，简称 BCD 码。在 BCD 码中用四位二进制代码，给 0～9 这 10 个数字编码。

BCD 码可作为人与计算机联系时的一种中间表示。在某些情况下，计算机也可以对这种形式表示的数进行运算。

BCD 码有多种编码方案，下面分别介绍：

1．8421BCD 码

8421BCD 码是一种使用最广泛的十进制编码。它按照二进制计数顺序选取前十个状态与 0～9 相对应。4 位二进制仍然具有二进制数位所具有的权，从左到右分别为 2^3、2^2、2^1、2^0，即 8、4、2、1。8421BCD 码与十进制数的对应关系如表 1-1 所示。

要注意的是 8421BCD 码中不允许出现 1010～1111 这六个代码，因为十进制数 0～9 中没有哪个数字符号与它们相对应，因此把它们称作“伪码”（或非法 BCD 码）。

8421BCD 码和十进制数之间的转换可以直接按位（或按组）转换。

【例 1-14】 将 1927 转换成四位 8421BCD 码。

解：将 1927 中各位数分别转换成 8421BCD 码，然后按高位到低位依次由左到右排列，可得：1927=（0001 1001 0010 0111）$_{8421BCD\text{码}}$

2．5421 BCD 码

5421BCD 码也是用四位二进制代码表示一位十进制数，这四位二进制数各位的权值不