

新编21世纪高等职业教育电子信息类规划教材



机电一体化技术专业

单片机技能与实训

宋国富 主 编
袁依凤 副主编
洪 应 主 审
王玉叶



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

新编 21 世纪高等职业教育电子信息类规划教材·机电一体化技术专业

单片机技能与实训

宋国富 主 编

王玉叶 副主编
袁依凤

洪 应 主 审

電子工業出版社·

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书主要介绍了单片机微处理器结构及系统、汇编语言及 C 语言程序设计、存储器、输入/输出和中断技术、接口技术、单片机的系统应用等内容。本书以案例为教学中心,将 51 单片机的理论体系贯穿到实践教学,特意引入了当今世界上是最优秀的单片机硬件仿真工具“Proteus”作为案例实施的主要载体,可以使单片机实践教学的实施直接在普通的微机室实现。

本书可作为高职高专电气类专业的教材,也可作为中职类相关专业教材或参考书,同时也可作为从事电气类专业工作的工程技术人员自学或参考用书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

单片机技能与实训/宋国富主编. —北京:电子工业出版社,2010.2
新编 21 世纪高等职业教育电子信息类规划教材·机电一体化技术专业
ISBN 978-7-121-10164-9

I. 单… II. 宋… III. 单片微型计算机—高等学校:技术学校—教材 IV. TP368.1

中国版本图书馆 CIP 数据核字(2009)第 243406 号

策 划:陈晓明

责任编辑:陈晓明 特约编辑:张凯贤

印 刷:涿州市京南印刷厂

装 订:涿州市桃园装订有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1092 1/16 印张:15.75 字数:403 千字

印 次:2010 年 2 月第 1 次印刷

印 数:4 000 册 定价:25.00 元

凡所购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010)88254888。

质量投诉请发邮件至 zlts@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010)88258888。

前 言

为满足国家对高技能型人才培养的要求,配合职业院校在教学过程中加大实践教学环节的力度,更好地进行基于项目化教学模式的改革,并结合编者多年的单片机实践教学经验,特编写了这套《单片机技能与实训》教材。

本教材在编写过程中,主要是围绕着解决职业院校学生“愿意学、学得会、用得上”等方面的问题,进行有效的内容组织,编写原则主要以注重能力培养,内容也主要以项目和任务为载体。

教材打破传统的以理论为主导的内容安排模式,而是以项目为导向,以“够用为度”。本书以单片机的实际应用为出发点,将传统单片机系统原理进行了整合,并以案例的形式体现出来,使学生通过对案例的实操,引导其主动寻求对原理知识的追寻,即按需学习,从而提高了学生学习的主观能动性。

教材内容组织以突出实践操作技能为主线,案例实施以单片机设计工作现场为背景,教学现场且以学生自己操作为主导,教师讲授为辅,更好地体现了对学生实践技能的培养。

教材针对于 MCS-51 系列单片机的理论体系,安排了二十三个针对性案例和一个综合案例,为使学习更贴近单片机系统实际开发现场,特意安排了 C51 的知识内容,并安排了 Keil C51 开发工具的操作技能,另外,考虑到单片机实验实训环节在具体实施过程中,总会由于诸多硬件方面的问题而使案例项目无法实现,从而影响教学效果,本书特意引入了当今世界上是最优秀的单片机硬件仿真软件“Proteus”作为案例实施的主要载体,可以使单片机实践教学的实施直接在普通的微机室实现。

本教材的第 1、2、4、7、11 章内容由安徽职业技术学院王玉叶编写,第 3、5 章内容由淮北职业技术学院袁依凤编写,其余内容由安徽职业技术学院宋国富编写,全书由宋国富担任主编,并负责统稿,安徽职业技术学院洪应主审了全书。

在本书的编写过程中,编者也参考了一些同类教材和同行的研究成果,并得到了有关院校领导的大力支持,在此深表感谢,特别要感谢安徽职业技术学院程周的支持与帮助。当然,由于编者水平有限,错误和不足之处在所难免,还请各位专家同行批评指正。

参加“新编 21 世纪高等职业教育电子信息类规划教材”

编写的院校名单(排名不分先后)

桂林工学院南宁分院

江西信息应用职业技术学院

江西蓝天职业技术学院

吉林电子信息职业技术学院

保定职业技术学院

安徽职业技术学院

杭州中策职业学校

黄石高等专科学校

天津职业技术师范学院

福建工程学院

湖北汽车工业学院

广州铁路职业技术学院

台州职业技术学院

重庆科技学院

济宁职业技术学院

四川工商职业技术学院

吉林交通职业技术学院

连云港职业技术学院

天津滨海职业技术学院

杭州职业技术学院

重庆电子工程职业学院

重庆工业职业技术学院

广州大学科技贸易技术学院

湖北孝感职业技术学院

江西工业工程职业技术学院

四川工程职业技术学院

广东轻工职业技术学院

广东技术师范职业技术学院

西安理工大学

辽宁大学高职学院

天津职业大学

天津大学机械电子学院

九江职业技术学院

包头职业技术学院

北京轻工职业技术学院

黄冈职业技术学院

郑州工业高等专科学校

泉州黎明职业大学

浙江财经学院信息学院

南京理工大学高等职业技术学院

南京金陵科技学院

无锡职业技术学院

西安科技学院

西安电子科技大学

河北化工医药职业技术学院

石家庄信息工程职业学院

三峡大学职业技术学院
桂林电子工业学院高职学院
桂林工学院
南京化工职业技术学院
湛江海洋大学海滨学院
江西工业职业技术学院
江西渝州科技职业学院
柳州职业技术学院
邢台职业技术学院
漯河职业技术学院
太原电力高等专科学校
苏州经贸职业技术学院
金华职业技术学院
河南职业技术师范学院
新乡师范高等专科学校
绵阳职业技术学院
成都电子机械高等专科学校
河北师范大学职业技术学院
常州轻工职业技术学院
常州机电职业技术学院
无锡商业职业技术学院
河北工业职业技术学院
天津中德职业技术学院
安徽电子信息职业技术学院
合肥通用职业技术学院
安徽职业技术学院
浙江工商职业技术学院
河南机电高等专科学校

深圳信息职业技术学院
河北工业职业技术学院
湖南信息职业技术学院
江西交通职业技术学院
沈阳电力高等专科学校
温州职业技术学院
温州大学
广东肇庆学院
湖南铁道职业技术学院
宁波高等专科学校
南京工业职业技术学院
浙江水利水电专科学校
成都航空职业技术学院
吉林工业职业技术学院
上海新侨职业技术学院
天津渤海职业技术学院
驻马店师范专科学校
郑州华信职业技术学院
浙江交通职业技术学院
江门职业技术学院
广西工业职业技术学院
广州市今明科技公司
无锡工艺职业技术学院
江阴职业技术学院
南通航运职业技术学院
山东电子职业技术学院
潍坊学院
广州轻工高级技工学校

目 录

第 1 章 数制与编码	(1)
1.1 不同进位计数制及其转换	(1)
1.1.1 进位计数制	(1)
1.1.2 数制的转换	(2)
1.2 计算机中数的表示方法及运算	(4)
1.2.1 机器数的特点	(4)
1.2.2 原码、反码和补码	(4)
1.3 BCD 码及文字符号代码	(6)
1.3.1 8421BCD 编码	(6)
1.3.2 ASCII 编码	(6)
1.3.3 实训 1 Keil C51 的使用方法	(7)
习题 1	(20)
第 2 章 MCS-51 单片机的硬件配置	(21)
2.1 单片机硬件系统的组成	(21)
2.1.1 微型机硬件系统的组成	(21)
2.1.2 单片微型计算机系统组成	(22)
2.2 MCS-51 单片机的引脚功能	(23)
2.2.1 MCS-51 单片机系列	(23)
2.2.2 MCS-51 单片机的内部组成及信号引脚	(23)
2.2.3 实训 2 并行端口特性	(26)
2.3 单片机内部资源分配	(28)
2.3.1 MCS-51 内部数据存储器	(28)
2.3.2 MCS-51 内部程序存储器	(33)
2.3.3 实训 3 Keil 对单片机各存储器单元的检查	(33)
2.3.4 实训 4 工作寄存器组的操作	(37)
2.3.5 实训 5 位寻址区的操作	(39)
习题 2	(41)
第 3 章 MCS-51 指令系统及汇编语言程序设计	(43)
3.1 指令概述	(43)
3.1.1 基本概念	(43)
3.1.2 指令格式	(44)
3.2 寻址方式	(44)
3.2.1 MCS-51 的寻址方式	(44)
3.2.2 实训 6 寻址方式	(46)
3.3 指令系统	(47)
3.3.1 内部 RAM 间的数据传送(16 条)	(48)

3.3.2	ACC 与外部数据存储器(或扩展 I/O 口)传递数据	(49)
3.3.3	查表指令	(49)
3.3.4	堆栈操作	(50)
3.3.5	累加器 A 的数据交换指令	(51)
3.3.6	实训 7 栈的操作	(51)
3.4	算术运算指令	(53)
3.4.1	加法指令	(53)
3.4.2	减法指令	(54)
3.4.3	十进制调整指令	(55)
3.4.4	乘除法指令	(56)
3.5	逻辑运算指令	(57)
3.5.1	基本与、或、异或运算指令	(57)
3.5.2	对 A 简单逻辑指令	(58)
3.6	控制转移指令(22 条)	(58)
3.6.1	无条件转移指令	(58)
3.6.2	条件转移指令	(59)
3.6.3	比较不相等转移指令	(60)
3.6.4	减“1”不为 0 转移指令	(61)
3.6.5	调用指令与返回指令	(61)
3.6.6	空操作指令	(63)
3.7	位操作指令	(63)
3.8	实训 8 MCS-51 指令系统	(64)
3.9	MCS-51 伪指令	(68)
3.9.1	汇编语言语句格式	(68)
3.9.2	常用伪指令	(68)
3.10	汇编语言程序的结构	(69)
3.10.1	概述	(69)
3.10.2	简单程序的设计	(70)
3.10.3	分支程序设计	(70)
3.10.4	循环程序设计	(73)
3.10.5	查表程序设计	(75)
3.11	汇编语言程序设计举例	(76)
3.11.1	控制流程的基本设计方法	(76)
3.11.2	应用程序控制流程设计	(77)
3.12	实训 9 汇编语言程序设计	(78)
	习题 3	(81)
第 4 章	单片机的 C 语言程序设计基础	(82)
4.1	单片机 C 程序设计的一般格式	(82)
4.1.1	单片机 C 语言编程的步骤	(82)
4.1.2	单片机 C 程序的几个基本概念	(82)
4.1.3	单片机 C 语言程序的基本结构	(84)
4.2	单片机 C 语言的数据类型	(84)
4.2.1	常量和符号常量	(85)

4.2.2 变量	(85)
4.3 单片机 C 语言的运算符和表达式	(86)
4.4 单片机 C 语言的一般语法结构	(87)
4.4.1 顺序结构	(87)
4.4.2 分支结构	(88)
4.4.3 循环结构	(90)
4.5 51 系列单片机的 C 语言程序	(90)
4.5.1 存储类型及存储区	(91)
4.5.2 特殊功能寄存器(SFR)	(94)
4.5.3 中断标识符	(95)
4.6 单片机 C 语言应用	(96)
4.6.1 C 语言程序编写举例	(96)
4.6.2 实训 10 Proteus 7 Professional 入门	(98)
4.6.3 实训 11 跑马灯	(106)
习题 4	(109)
第 5 章 存储器系统	(111)
5.1 MCS-51 单片机外部总线的扩展	(111)
5.1.1 单片机的三总线结构	(111)
5.1.2 典型 8 位地址总线锁存芯片	(112)
5.2 存储器概述	(112)
5.2.1 存储器的分类	(112)
5.2.2 随机存取存储器	(113)
5.2.3 只读存储器	(115)
5.3 存储器扩展的基本方法	(118)
5.3.1 存储器扩展的基本问题	(118)
5.3.2 单片机存储器的片选技术	(118)
5.3.3 实训 12 单片机存储器的综合扩展	(121)
习题 5	(122)
第 6 章 中断技术	(124)
6.1 中断概述	(124)
6.1.1 中断的概念	(124)
6.1.2 中断的特点	(125)
6.1.3 计算机的中断源	(125)
6.2 MCS-51 中断系统	(126)
6.2.1 MCS-51 中断系统的结构	(126)
6.2.2 中断源和中断控制	(126)
6.2.3 中断处理过程	(129)
6.2.4 外部中断源的扩展	(130)
6.3 中断应用举例	(131)
6.3.1 实训 13 多信号彩灯的中断实现	(131)
6.3.2 实训 14 方波信号发生器的中断实现	(134)
习题 6	(136)
第 7 章 定时/计数器	(138)

7.1	MCS-51 定时/计数器的构成	(138)
7.1.1	MCS-51 定时/计数器的结构与工作原理	(138)
7.1.2	定时/计数器相关寄存器	(139)
7.2	定时/计数器工作方式及应用	(141)
7.2.1	定时/计数器工作方式	(141)
7.2.2	定时/计数器应用	(144)
7.2.3	实训 15 电子钟	(147)
7.2.4	实训 16 计数器	(152)
习题 7		(155)
第 8 章	输入/输出设备与接口	(157)
8.1	计算机 I/O 接口技术概述	(157)
8.1.1	I/O 设备及 I/O 接口	(157)
8.1.2	CPU 与外设之间数据传送方式	(158)
8.2	MCS-51 并行输入/输出口	(160)
8.2.1	P0 口(80H)的结构和功能	(160)
8.2.2	P2 口(A0H)的结构和功能	(162)
8.2.3	P1 口(90H)的结构和功能	(162)
8.2.4	P3 口(B0H)的结构和功能	(163)
8.3	并行 I/O 口的应用	(165)
8.3.1	并行 I/O 口的应用说明	(165)
8.3.2	实训 17 多个灯的智能控制	(165)
8.4	并行 I/O 口的扩展	(167)
8.4.1	简单 I/O 口的扩展	(167)
8.4.2	通用可编程并行接口 8255A 的结构	(168)
8.4.3	MCS-51 和 8255A 的接口方法	(173)
8.4.4	实训 18 8255A 扩展的彩灯控制	(174)
8.5	八段数码显示接口技术	(177)
8.5.1	LED 数码管工作原理及应用	(177)
8.5.2	实训 19 八段数码管的显示控制	(179)
8.5.3	实训 20 数码管动态显示的串行驱动	(181)
8.6	键盘接口技术	(185)
8.6.1	键盘接口技术及原理	(185)
8.6.2	实训 21 矩阵式键盘的按键识别	(187)
习题 8		(190)
第 9 章	串行通信	(192)
9.1	串行通信基本概念	(192)
9.1.1	并行通信和串行通信	(192)
9.1.2	串行通信的数据传送方向	(193)
9.1.3	串行通信的工作方式	(194)
9.1.4	串行通信的波特率(Baud rate)	(194)
9.2	MCS-51 单片机的串行口	(195)
9.2.1	串行口的结构	(195)
9.2.2	串行通信所用的专用寄存器	(195)

9.2.3 MCS-51 串行口的工作方式 (196)

9.2.4 MCS-51 串行口的波特率..... (198)

9.3 MCS-51 单片机串行通信的应用 (200)

9.3.1 串行通信的标准 (200)

9.3.2 实训 22 串行口的双机通信应用..... (201)

9.3.3 实训 23 主从式多机通信 (207)

习题 9 (215)

第 10 章 A/D、D/A 转换接口 (217)

10.1 D/A 转换接口 (217)

10.1.1 D/A 转换器的原理及主要技术参数 (217)

10.1.2 8 位 D/A 转换器 DAC0832 (218)

10.1.3 实训 24 智能信号发生器 (220)

10.2 A/D 转换接口 (223)

10.2.1 A/D 转换器原理及参数 (223)

10.2.2 逐次逼近式 A/D 转换器 ADC0809 (224)

10.2.3 实训 25 智能电压表 (226)

10.3 串行 A/D、D/A 转换接口 (230)

习题 10 (232)

第 11 章 综合应用案例——交通灯 (233)

第1章 数制与编码

学习目标

本章主要介绍了计算机中关于数的表示方法，几种常用数制的转换，机器数的表示方法和常用编码等内容。学生通过本章数的基础知识的学习，可以为后续单片机原理的学习打下基础。

知识点

- (1) 二进制、十六进制、十进制表达形式及其相互转换。
- (2) 机器数中关于带符号数的原码、反码、补码的表达形式及其相互转换。
- (3) ASCII 码、BCD 码的表达形式及其相互转换。

1.1 不同进位计数制及其转换

1.1.1 进位计数制

计算机其实就是一种由数字电路演变而来的能进行逻辑运算的机器，其处理的信息就是数字电路所提到的二进制数，而人们常使用的是十进制数，这样，为了能顺利地人与计算机之间进行信息交换，一定要进行不同进制数之间的转换操作，因此我们有必要掌握数制及数制转换的原理。

进位计数制：按进位的原则进行计数的一种方法。

进位计数制有以下两个特点：

- (1) 有一个固定的基数 r ，数的每一位只能取 r 个不同的数字，即所使用的数码为 $0, 1, 2, \dots, r-1$ 。
- (2) 逢 r 进位，它的第 i 个数位对应于一个固定的值 r^i ， r^i 称为该位的“权”。小数点左侧各位的权是基数 r 的正次幂，依次为 $0, 1, 2, \dots, m$ 次幂，小数点右侧各位的权是基数 r 的负次幂，依次为 $-1, -2, \dots, -n$ 次幂。

1. 十进制

十进制的基数为 10，它所使用的数码为 $0 \sim 9$ ，共 10 个数字。十进制各位的权是以 10 为底的幂，即每个数所处的位置不同，它的值是不同的，每一位数是其右边相邻那位数的 10 倍。

例如，数 555.55 就是下列多项式的缩写：

$$555.55D = 5 \times 10^2 + 5 \times 10^1 + 5 \times 10^0 + 5 \times 10^{-1} + 5 \times 10^{-2}$$

式中的后缀 D (Decimal) 表示该数为十进制数, 通常对十进制数可不加后缀。

2. 二进制

二进制的基数为 2, 它所使用的数码为 0、1, 共 2 个。二进制各位的权是以 2 为底的幂, 即 $\dots, 2^2, 2^1, 2^0, 2^{-1}, 2^{-2}, \dots$ 。

例如, 二进制数 1011.101 相当于十进制数:

$$1011.101\text{B} = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = 11.625$$

二进制数的运算规则类似于十进制, 加法为逢二进一, 减法为借一为二。利用加法和减法就可以进行乘法、除法以及其他数值运算。上式中的后缀 B (Binary) 表示该数为二进制数。

3. 十六进制

十六进制的基数为 16, 它所使用的数码共有 16 个: 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F, 其中 A~F 相当于十进制数的 10~15。十六进制的权是以 16 为底的幂, 即 $\dots, 16^2, 16^1, 16^0, 16^{-1}, 16^{-2}, \dots$ 。

例如, 十六进制数 A3E.8F 相当于十进制数:

$$\text{A3E.8FH} = 10 \times 16^2 + 3 \times 16^1 + 14 \times 16^0 + 8 \times 16^{-1} + 15 \times 16^{-2} = 2622.5059$$

式中的后缀 H (Hexadecimal) 表示该数为十六进制数。十六进制数如是字母打头, 则在使用汇编指令时前面需加一个 0, 例如, 0FFFFH (65535)。

1.1.2 数制的转换

1. 二进制数、十六进制数转换成十进制数

根据定义, 只需将二、十六进制数按权展开后相加即可转换成相应的十进制数。例如,

$$1011\text{B} = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 11$$

$$\text{A4H} = 10 \times 16^1 + 4 \times 16^0 = 164$$

2. 十进制数转换成二进制数、十六进制数

一个十进制整数转换成二进制数 (或十六进制数) 时, 通常采用“除二取余” (或除十六取余) 法, 即用“2” (或 16) 连续除十进制数至商为 0, 逆序排列余数即可得到。

十进制小数部分转换成二进制数 (或十六进制数) 时, 通常采用“乘 2 取整” (或乘十六取整) 的方法得到。即将十进制数的纯小数部分一次一次地乘“2” (或 16), 把每次得到的整数按顺序排列, 就得到用二进制数 (或十六进制数) 表示的小数值。

例 1-1 将 62.6875 转换成二进制数。

解:

(1) 十进制整数 62 转换为二进制数, 采用“除二取余”法。

2 62	余数 = 0	低位  高位
2 31	余数 = 1	
2 15	余数 = 1	
2 7	余数 = 1	
2 3	余数 = 1	
1 ...	余数 = 1	

按余数的逆序排列，即得结果：62D = 111110B。

(2) 十进制小数 0.6875 转换为二进制数，采用“乘二取整”法。

0.6875		
× 2		
1.3750	整数部分 = 1	高位  低位
0.3750		
× 2		
0.7500	整数部分 = 0	
0.7500		
× 2		
1.5000	整数部分 = 1	
0.5000		
× 2		
1.0000	整数部分 = 1	

按整数的顺序排列，即得结果：0.6875 = 0.1011B，最终结果是：62.6875 = 111110.1011B。

3. 二进制数与十六进制数的相互转换

在计算机中引入十六进制数的本质是为了更简化地描述二进制数。即十六进制就是二进制的压缩码，一位十六进制数就等同于 4 位二进制数。所以转换时，只要根据这个原则就很容易实现二者之间的转换操作。

(1) 二进制数转换成十六进制数。转换时，将二进制数整数部分由右向左每 4 位一段，最后不足部分左面补零；小数部分由左向右每 4 位一段，最后不足部分右面补零，然后，每 4 位二进制数用 1 位十六进制数代替，便转换成了十六进制。例如：

$$1011110101.110B = 0010\ 1111\ 0101.1100B = 2F5.CH$$

2 F 5 C

(2) 十六进制数转换成二进制数。转换时，只要将每一位十六进制数用对应的 4 位二进制数代替，便转换成了二进制数。例如：

$$BD5.6H = 1011\ 1101\ 0101.0110B$$

B D 5 6

1.2 计算机中数的表示方法及运算

1.2.1 机器数的特点

1. 机器数的字长

机器数所能表示的数的范围受到计算机字长的限制，如对于 8 位字长的计算机来说，机器数的范围为 $(0000\ 0000)_2 \sim (1111\ 1111)_2$ ，即对应于十进制数为 $0 \sim 255$ 。

为了扩大机器数表示的范围，有时可以用两个字甚至多个字表示一个数，例如对于 8 位机来说，若用两个字来表示一个正数，其数值范围为 $0 \sim 65535$ 。

2. 机器数的符号

机器数只用来表示正数，这时称为无符号数。如果用来既表示正数也可以表示负数时，这时称为带符号数。此时，数的符号在机器中就数码化了，即将一个字的最高位定为符号位，其余各位为数值位。最高位为 0 表示正数，最高位为 1 表示负数。例如：

$N_1 = (+101\ 1001)_2$ 的机器数可表示为： $N_1 = (0101\ 1001)_2$ ， $N_2 = (-110\ 1011)_2$ 的机器数可表示为 $N_2 = (1110\ 1011)_2$ ，带符号数也可以用两个字来表示一个数，此时，符号位仍定为两个字的最高一位。

1.2.2 原码、反码和补码

1. 原码

正数的符号位用“0”表示，负数的符号位用“1”表示，其余各位表示数值，这种表示法称为原码。例如：

$$X_1 = [+100\ 0001]_2 = +65 \quad [X_1]_{\text{原}} = 0100\ 0001$$

$$X_2 = [-100\ 0001]_2 = -65 \quad [X_2]_{\text{原}} = 1100\ 0001$$

左边表示的数称为真值，即为某数的实际算术值。右边为用原码表示的数，两者的最高位分别用“0”、“1”代替符号位的“+”、“-”。

在原码表示法中，“0”有两种表示法，即：

$$[0]_{\text{原}} = +0 = 0000\ 0000, [0]_{\text{原}} = -0 = 1000\ 0000$$

2. 反码

一个数的反码可由原码求得。如果是正数，则其反码与原码相同；如果是负数，则其反码除符号位为 1 外，其他各数位均按位取反，即 1 转换为 0，0 转换为 1。例如：

$$X_1 = +100\ 0001 \quad [X_1]_{\text{反}} = 0100\ 0001$$

$$X_2 = -100\ 0001 \quad [X_2]_{\text{反}} = 1011\ 1110$$

如果已知一个数的反码，求它的真值，若是正数则可直接求得，若是负数则可将符号位除外的数值部分各位取反得到负数的原码，然后再求真值。例如：

$$[X_1]_{\text{反}} = 0100\ 0001$$

$$[X_1]_{\text{原}} = 0100\ 0001 \quad X_1 = +65$$

$$[X_2]_{\text{反}} = 1011\ 1110$$

$$[X_2]_{\text{原}} = 1100\ 0001 \quad X_2 = -65$$

在反码表示法中，“0”也有两种表示形式，即：

$$[0]_{\text{反}} = +0 = 0000\ 0000$$

$$[0]_{\text{反}} = -0 = 1111\ 1111$$

3. 补码

一个数的补码一般由反码求得。如果是正数，则其补码与原码相同；如果是负数，则其补码为反码加1，即“取反后再加1”。例如：

$$X_1 = +100\ 0001$$

$$[X_1]_{\text{补}} = 0100\ 0001$$

$$X_2 = -100\ 0001$$

$$[X_2]_{\text{补}} = [X_2]_{\text{反}} + 1 = 1011\ 1110 + 1 = 1011\ 1111$$

“求反加1”需要作两步运算，这个过程也可以简化为一步，即符号位不变，只对原码各位中最低一位1以左的各位求反，而最低一位1和右边各位都不变，即可得到负数的补码。

已知X的补码，求X的原码时，可以将X的补码当作X原码形式，再求一次X的补码得到，即 $[[X]_{\text{补}}]_{\text{补}} = [X]_{\text{原}}$ 。

例 1-2 已知 $[X]_{\text{补}} = 1011\ 1111$ ， $[Y]_{\text{补}} = 0001\ 0110$ ，求X、Y的真值。

解：

$$(1) [X]_{\text{原}} = [[X]_{\text{补}}]_{\text{补}} = [1011\ 1111]_{\text{补}} = 1100\ 0000 + 1 = 1100\ 0001$$

因为X为负数，所以X的真值为：

$$X = -100\ 0001\text{B} = -65$$

$$(2) [Y]_{\text{原}} = [[Y]_{\text{补}}]_{\text{补}} = [0001\ 0110]_{\text{补}} = 0001\ 0110$$

因为Y为正数，所以Y的真值为：

$$Y = 0001\ 0110\text{B} = +22$$

在补码表示法中，0只有一种表示形式，即：

$$[0]_{\text{补}} = 0000\ 0000$$

对于8位有符号数来说，用补码所表示的数的范围为： $-128 \sim +127$ ，其中，“-128”的补码为“1000 0000”。

需要指出的是，在计算机中，所有带符号数均以补码形式表达的。因为计算机在数据处理时，特别是数据运算时，是把符号位看作数值位来运算的，只有补码运算才能保证运算结果的正确性，如下面两个数的加法运算：

$$\begin{array}{r} 1000\ 0001 \\ +\ 0000\ 0001 \\ \hline 1000\ 0010 \end{array}$$

如果把上述两数看做无符号数，则是 $129 + 1 = 130$ ，结果显然是正确的。

如果把上述两数看做有符号数，则“1000 0001”是“-127”的补码，“0000 1000”是“+1”的补码，而结果“1000 0010”正好是“-126”的补码，显然，结果也是正确的。

1.3 BCD 码及文字符号代码

1.3.1 8421BCD 编码

BCD (Binary Coded Decimal) 编码就是用二进制代码表示的十进制数, 即计算机眼中的十进制数。在 8421BCD 码中, 是用 4 位二进制数 0000~1001 给 0~9 这 10 个数字编码, 具体表示见表 1.1。

表 1.1 8421BCD 码与十进制数的关系

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

利用上表可以很容易地实现 8421BCD 码与十进制数之间的转换。例如,

$$(0100\ 1001\ 0111)_{\text{BCD}} = 497$$

8421BCD 码不是二进制数, 它只是一种编码, 如果要转换为二进制数, 要先经过十进制, 然后再转换为二进制, 反之过程类似。

1.3.2 ASCII 编码

字母与字符用二进制编码的方法很多。目前计算机中用得最广泛的字符编码, 是由美国国家标准局制定的 ASCII 码 (American Standard Code for Information Interchange, 美国标准信息交换码), 它已被国际标准化组织 (ISO) 定为国际标准, 称为 ISO 646 标准。适用于所有西文字母, ASCII 码的二进制位数共有 8 位, 也就是所说的一个字节, 但参与编码的位数一般只有 7 位, 在西文编码中, 最高位恒为 “0”。

因为 1 位二进制数可以表示 $(2^1) = 2$ 种状态: 0、1; 而 2 位二进制数可以表示 $(2^2) = 4$ 种状态: 00、01、10、11; 依次类推, 7 位二进制数可以表示 $(2^7) = 128$ 种状态, 每种状态都唯一地编为一个 7 位的二进制码, 对应一个字符, 这些码字可以排列成一个序号 0~127 (00H~7FH)。所以, ASCII 码是用 7 位二进制数进行编码的, 可以表示 128 个字符。具体码列表可参见表 1.2。

在 ASCII 码表中, 第 0~32 (00H~20H) 号及第 127 (7FH) 号共 34 种编码是控制字符或通讯专用字符, 如控制符: LF (换行)、CR (回车)、FF (换页)、DEL (删除)、BEL (振铃) 等; 通讯专用字符: SOH (文头)、EOT (文尾)、ACK (确认) 等。

第 33~126 号 (21H~7EH) 共 94 种编码是字符编码, 其中第 48~57 (30H~39H) 号为 “0~9” 十个阿拉伯数字; 65~90 (41H~5AH) 号为 26 个大写英文字母 “A~Z”, 97~122 (61H~7AH) 号为 26 个小写英文字母 “a~z”, 其余为一些标点符号、运算符等。

例如, 数字 0~9 的 ASCII 码可表示为十六进制数为 30H~39H, 大写字母 A~Z 的 ASCII 码可表示为十六进制数为 41H~5AH, a~z 为 61H~7AH, 回车符为 0DH, 换行符为 0AH 等。