

新世纪高职高专教改项目成果教材

单片机控制技术的应用

缴瑞山 韩全立 主编



高等教育出版社

HIGHER EDUCATION PRESS

新世纪高职高专教改项目成果教材

单片机控制技术的应用

主编 缴瑞山 韩全立

高等教育出版社

内容提要

本书是新世纪高职高专教改项目成果系列教材之一,是面向 21 世纪课程教材。全书以目前较为流行的 89C5X 单片机为核心,系统介绍了单片机控制技术的应用,内容包括:单片机基本结构和工作原理、单片机指令系统、单片机内部功能部件及应用、单片机系统扩展、单片机控制技术的应用以及单片机应用技术实验等。

本书可作为高等职业学校机电一体化及相关专业教学用书,亦可供相关技术人员参考。

图书在版编目(CIP)数据

单片机控制技术的应用/缴瑞山,韩全立编著. —北京:
高等教育出版社,2003.7

ISBN 7-04-012018-6

I.单... II.①缴...②韩... III.单片微型计算机-
计算机控制-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(2003)第 047647 号

责任编辑 李宇峰 封面设计 吴 昊 责任印制 蔡敏燕

书 名 单片机控制技术的应用
编 著 缴瑞山 韩全立

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100011
总 机 010-82028899
传 真 021-56965341

购书热线 010-64054588
021-56964871
免费咨询 800-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
<http://www.hepsh.com>

排 版 南京理工排版校对公司
印 刷 商务印书馆上海印刷股份有限公司

开 本 787×1092 1/16
印 张 18
字 数 440 000

版 次 2003 年 6 月第 1 版
印 次 2003 年 6 月第 1 次
定 价 24.40 元

凡购买高等教育出版社图书,如有缺页、倒页、脱页等质量问题,请在所购图书销售部门联系调换。

版权所有 侵权必究

出版说明

前 言

本书以目前较为流行的 89C5X 系列(MCS-51 兼容)单片机为核心,介绍了单片机基本原理、系统结构、指令系统、程序设计及单片机控制系统的设计方法,并以较大篇幅讲述了单片机在实际使用中的控制技术、应用技巧及新型外围元器件的使用。

本书突出实用性,突出新技术的应用及新产品的使用,以使教材具有新颖、实用、通俗等特点,教材以一种全新的内容展现给读者。

全书从内容上分为四大部分:单片机系统基本知识、单片机系统扩展技术、单片机控制应用技术、单片机系统基本实验。

第一章至第四章讲述单片机系统基本知识,内容包括计算机基础知识、数制和编码、最新常用单片机介绍、MCS-51 单片机的基本结构和原理、MCS-51 单片机的指令系统和基本编程方法。

第五章讲述单片机系统扩展技术,内容包括存储器扩展、并行输入/输出接口,串行输入/输出接口、A/D 和 D/A 接口以及系统扩展的新技术。

第六章是单片机控制技术的应用,主要内容包括构建单片机控制系统的基本方法、抗干扰设计、单片机的选型、传感器与前向通道的应用技术、控制系统与后向通道的应用技术、人机对话通道的应用技术、单片机的综合应用技术(看门狗、加密等)等。

第七章的实验包括了各类指令的练习;各种程序结构的练习;键盘识别和数码管的驱动训练;1-Wire、I²C、Three-Wire、并行接口等接口技术的使用;A/D、D/A 等方面的实验,训练类别比较齐全。

本教材在内容和编写上注重新技术的运用,并且把主要的课时也安排在这些部分,主要的思想是让读者在单片机应用方面多接触这些新且广泛应用的技术,开拓眼界。

本书是自动化、电子、仪表和机电等专业的主要课程教材,同时可供工程技术人员参考使用。

本书由辽宁机电职业技术学院缴瑞山、河南工业职业技术学院韩全立主编。

本书第一章和第二章由河南工业职业技术学院韩全立编写,第三章由包头轻工职业技术学院赵金考编写,第四章由河南工业职业技术学院史志强编写,前四章由韩全立统稿;第五章由辽宁机电职业技术学院缴瑞山编写,第六章和第七章由辽宁机电职业技术学院许连阁编写,后三章由缴瑞山统稿。

对于本书存在的问题,敬请大家多多批评指正。

编 者

2003 年 5 月

目 录

第一章 概述.....	1
第一节 计算机的基础知识.....	1
第二节 计算机中的数制和编码.....	3
第三节 单片机总体介绍	12
第四节 常用单片机系列介绍	16
习题与思考题	21
第二章 单片机的基本结构及工作原理	23
第一节 MCS-51 单片机的性能及结构	23
第二节 AT89 系列单片机的性能及结构	27
第三节 单片机的工作方式	42
习题与思考题	47
第三章 单片机的指令系统	48
第一节 指令系统概述	48
第二节 寻址方式	49
第三节 MCS-51 指令系统	53
第四节 程序设计举例	64
习题与思考题	71
第四章 单片机内部功能部件及应用	73
第一节 AT89 单片机中断系统	73
第二节 片内定时器/计数器(T/C)	78
第三节 外部中断应用举例及扩展	88
第四节 串行通信的基本概念	92
第五节 AT89C51 单片机串行 I/O 接口.....	95
习题与思考题.....	106
第五章 单片机系统扩展.....	107
第一节 系统扩展概述.....	107
第二节 存储器扩展.....	108
第三节 I/O 接口扩展技术	113
第四节 系统扩展新技术.....	128
第五节 模/数转换接口及应用	144

第六节 数/模转换接口及应用	153
习题与思考题	158
第六章 单片机控制技术的应用	160
第一节 构建单片机控制系统的基本方法	160
第二节 单片机控制系统的抗干扰设计	166
第三节 单片机的选型	175
第四节 传感器与前向通道的应用技术	179
第五节 控制系统与后向通道的应用技术	186
第六节 人机对话通道的应用技术	196
第七节 单片机的综合应用技术	219
习题与思考题	231
第七章 单片机应用技术实验	233
第一节 单片机控制系统的仿真调试	233
第二节 软件实验	238
第三节 硬件实验	251
参考文献	277

第一章 概 述

第一节 计算机的基础知识

电子计算机(Electronic Computer)是一种能自动、高速、精确地进行信息处理的现代化电子设备。计算机具有很强的算术运算和逻辑运算能力,并能通过预先编制好的程序自动地完成数据的加工处理,这也是计算机与计算器的根本区别。计算机的应用已为人们带来了巨大的经济效益和社会效益,计算机已经发展成为人类社会生活不可缺少的智能工具,因此,计算机也被誉为 20 世纪最重大的科学技术成就之一。

本章首先对计算机的发展做一概述,然后简要地介绍冯·诺依曼型计算机及微型计算机;计算机中的数制和编码;单片微型计算机简介等,使读者对计算机的结构及单片微型计算机系统有一个基本的认识,为以后各章的学习建立一个总体的概念。

一、计算机发展概述

1946 年 2 月世界上第一台电子数字积分计算机在美国宾夕法尼亚大学诞生,这台计算机的名字称为 ENIAC(Electronic Numerical Integrator and Calculator),中文的意思为“电子数值积分和计算机”。这台计算机使用了 18 000 只电子管,重达 30 t,占地超过 160 m²,功率为 150 kW,价值 40 万美元,但运算速度仅为 5 000 次/秒。它采用了科学家冯·诺依曼(John Von Neumann,美籍匈牙利人)的“存储程序”和“用二进制代替十进制”的设计方案,并构建了计算机由运算器、控制器、存储器、输入设备和输出设备所组成的这一基本结构,这种设计思想一直沿用至今。这是计算机科学技术发展史上的一个重要里程碑,它奠定了现代计算机发展的科学基础,开创了电子计算机时代。

随着科学技术的不断发展,计算机先后经历了电子管、晶体管、集成电路、大规模和超大规模集成电路为主要器件的四个发展时代。其各代计算机的主要特征如下。

第一代计算机出现在 1946 年至 1958 年间,其主要特征为:主机采用电子管器件,主要应用于科学计算,软件采用机器语言和汇编语言。

第二代计算机出现在 1958 年至 1964 年间,其主要特征为:主机采用晶体管器件,应用领域涉及科学计算和数据处理两个方面,软件采用高级(算法)语言和操作系统。

第三代计算机出现在 1964 年至 1971 年间,其主要特征为:主机采用集成电路,其应用领域更加广泛,计算机设计思想已逐步形成标准化、模块化和系列化。

第四代计算机从 1971 年出现至今,其主要特征为:主机采用中、大和超大规模集成电路,应用领域已涉及各行各业,在系统结构方面向着多机系统、分布式系统、计算机网络和微型计算机方向发展,系统软件也朝着智能化方向发展。

预计在不久的将来,将诞生以超导器件、电子仿真、集成光路等技术支撑的第五代计算

机。目前,计算机总的发展趋势是朝着巨型化、微型化、网络化、智能化的方向发展。

二、冯·诺依曼型计算机

现代计算机经过了半个多世纪的发展,计算机系统结构发生了重大变化。但冯·诺依曼提出的“存储程序”思想,即程序和数据都被存放在内存中的工作方式,仍然被现代计算机所采用。因此,现在的计算机基本都可被称为冯·诺依曼型计算机。

冯·诺依曼型计算机的主要特点如下:

1. 采用二进制代替十进制运算

计算机内部的信息都是用二进制编码的。这样设计的主要原因是二进制的 0 和 1 两种状态易于用电路来实现,且与逻辑代数的真(True)和假(False)相对应,运算规则简单等。而十进制的 10 种状态不易用电路来实现,且运算规则相对复杂等。因此,对于输入计算机的各种信息,都是被转化为二进制的代码后,再进行运算的。比如,键盘上的每一个字符键都对应一组二进制代码,每次按键操作都意味着向计算机输入了一组二进制代码。目前,键盘上普遍采用的代码是美国信息交换用标准代码(American Standard Code for Information Interchange),简称 ASCII 码。

2. 存储程序工作方法

将计算机要处理的数据和计算方法、步骤,事先按计算机要执行的操作命令(即指令)和有关原始数据编制成程序(即指挥计算机工作的指令的集合),存放在计算机内部的存储器中,计算机在运行时能够自动地、连续地从存储器中取出指令并执行,不需人工加以干预。

3. 计算机硬件系统的构成

计算机硬件系统由运算器、控制器、存储器、输入设备和输出设备等 5 个部分构成。图 1-1 所示为计算机硬件系统的基本组成框图。

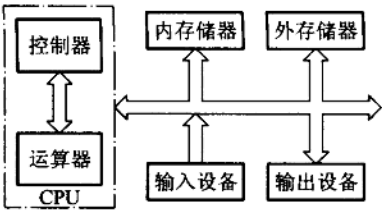


图 1-1 计算机硬件组成框图

为了使计算机按照人们的意图进行运算,必须事先把计算方法和操作步骤编制成可执行的程序,通过输入设备将程序转换成二进制形式的机器语言输入,并存放在计算机的存储器中。启动计算机后,计算机就从存储器的开始地址中取出指令送到控制器中进行分析,控制器会根据指令的操作要求发出一系列相应的执行命令。

程序执行过程中或者执行结束时,可以通过输出设备将中间或最终结果输出。这样,计算机便可以在程序的控制下,按照人们的意图自动工作了。

三、微型计算机的出现

1971 年 1 月,美国英特尔(INTEL)公司的特德·霍夫在与日本商业通讯公司合作研制

台式计算器时,将原始方案的十几个芯片压缩成3个集成电路芯片。其中,2个芯片分别用于存储程序和数据,另1片是将运算器和控制器以及控制逻辑集成在一片集成电路芯片上,称其为中央处理器(CPU)芯片,命名为Intel4004。这就使得计算机的体积大大缩小。因此,人们称之为微处理器。将微处理器再配以存储器、输入/输出接口便构成了微型计算机。

以微型计算机为主体,再配以系统软件、输入设备和输出设备便构成了完整的微型计算机系统。IBM 等公司相继推出了 IBM PC、IBM PC/XT 等微型计算机系统,极大地推动了微型计算机在各行各业的应用。

微型计算机的发展是以微处理器的发展为特征的。自 Intel4004 以后,Intel 公司相继推出了 Intel8080、Intel8086、Intel8088、Intel80286、Intel80386、Intel80486、Pentium、Pentium Pro、Pentium II、Pentium III、Pentium V 等各种微处理器。反映微处理器性能的计算机的字长也由4位发展到32位,并继续向64位发展;时钟频率已经达到2.50 GHz以上。此外,计算机的存储器和输入/输出设备的性能也已达到了较高的水平。

在微型计算机发展的同时,1975年世界上第一块单芯片微型计算机 TMS-1000 问世。单芯片微型计算机(Single Chip Microcomputer)简称单片机,又称 MCU(Micro Controller Unit 微控制器),属于第四代电子计算机(以大规模集成电路为特征)。它把中央处理器(CPU)、存储器(RAM 和 ROM)、输入/输出接口电路以及定时器/计数器等集成在一块芯片上,形成了真正意义上的“单片机”。单片机技术作为一个非常有前途的计算机技术,其应用已遍及工业控制、智能化仪表和家用电器等众多领域。

第二节 计算机中的数制和编码

由上节所述,输入计算机系统内部的各种信息,包括使用的所有指令或数据都要采用二进制代码。二进制是计算机惟一能识别的机器语言,因此,所有需要计算机处理的数字、字母、符号等信息,都必须采用二进制表示。但人们日常生活中使用的计数法是十进制数,二进制与十进制间是如何转换的?数字、字母、符号等在计算机中是如何编码的?下面就解决这些有关计算机的基础问题。

一、数制简介

人们最熟悉的是十进制数,这是一种基数为10的数制,逢十进一。除此之外,人们还用到的有八进制、十二进制和十六进制等。在计算机中常用的有二进制、八进制、十进制和十六进制,为了区别不同的进位计数制,一般在数字后面加上数制,如:2代表二进制、16代表十六进制等;也可以用字母来表示数制,B(Binary)代表二进制、O(Octal)代表八进制、D(Decimal)代表十进制、H(Hexadecimal)代表十六进制。

1. 十进制

十进制中,一个数由一组有序的数码组成,数码可以用0、1、2、3、4、5、6、7、8、9等

10 个数字表示。数码在数中的位置称为数位,不同数位上的数码代表不同的数值,每个数码所表示的数值等于数码乘以与其所在的数位的位权,十进制计数采用“逢十进一”的规则,所以位权表示为以 10 为底的幂,所以任意一个十进制数都可以用多项式表示,例如:

$$156.87 = 1 \times 10^2 + 5 \times 10^1 + 6 \times 10^0 + 8 \times 10^{-1} + 7 \times 10^{-2}$$

因此,由十进制数 156.87 的各数位可知,这个数是由 1 个 $100(10^2)$ 、5 个 $10(10^1)$ 、6 个 $1(10^0)$ 、8 个 $0.1(10^{-1})$ 和 7 个 $0.01(10^{-2})$ 组成的。由此可知,这个表达式是每个十进制数的位表示的权值,和与权对应的数字位的乘积之和。

2. 二进制

二进制与十进制很相似。二进制的数码只有 2 个,用 0、1 表示,计数采用“逢二进一”的规则,位权表示为以 2 为底的幂。例如,有一个二进制数,它的多项式表示形式如下。同时也可求得与它等价的十进制数。

$$\begin{aligned} (10101.01)_2 &= 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} \\ &= (16 + 0 + 4 + 0 + 1 + 0 + 0.25)_{10} = (21.25)_{10} \end{aligned}$$

将一个二进制位称为位或比特(Bit),且对特定位数的二进制数有如下定义:

4 位二进制数称为半个字节;

8 位二进制数称为一个字节;

16 位二进制数称为一个字;

32 位二进制数称为一个双字。

一般将数最左边的位,称为最高有效位,记为 MSB(Most Significant Bit);最右边的位称为最低有效位,记为 LSB(Least Significant Bit)。表 1-1 给出了二进制数各位的权值。

表 1-1 二进制各位的权值

2^0	1	2^7	128	2^{14}	16 384
2^1	2	2^8	256	2^{15}	32 768
2^2	4	2^9	512	2^{-1}	0.5
2^3	8	2^{10}	1 024	2^{-2}	0.25
2^4	16	2^{11}	2 048	2^{-3}	0.125
2^5	32	2^{12}	4 096	2^{-4}	0.062 5
2^6	64	2^{13}	8 192	2^{-5}	0.031 25

3. 八进制

同理,八进制数的数码有 8 个,它们是 0、1、2、3、4、5、6、7,计数规则为“逢八进一”,位权为以 8 为底的幂。

4. 十六进制

十六进制数的数码有 16 个,用 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F 表示,

计数规则采用“逢十六进一”，位权为以 16 为底的幂。

对于以上各数制，通常称其各位的权分别为 0 权位、1 权位、2 权位等。

二、不同数制间的转换

上述介绍的 4 种进位数制中，十进制数转换成二进制、八进制或十六进制数的方法比较麻烦：整数部分转换采用基数连除法，而小数部分则要采用基数连乘法。因此，一个带小数的十进制数的转换，应该分别作整数和小数部分的转换后，再相加组合成转换的结果编码。二进制、八进制或十六进制数转换成十进制数，则直接采用按权值表示的展开式就可求得。

1. 十进制转换成其他进制

(1) 十进制整数部分的转换

十进制数整数部分的转换采用基数连除法，即逐次除以基数，得出的余数，就是这个基数的转换编码。转换的过程如下：

- ① 用基数除以十进制数整数部分。
 - ② 保留余数（第一次求得的余数，作为转换编码的最低数字）。
 - ③ 重复执行上述过程，直到商为 0。最后，从低开始向上读取余数，即得转换结果。
- 这种转换过程，由于连续使用基数作除法运算，故称为基数连除法。

基数连除法的转换过程可表述为：作除法，要余数，从低向高排。

例如，若将十进制整数 265 转换成十六进制，可写成如下形式：

$$\begin{array}{r} 16 \overline{) 265} \cdots \text{余 } 9 \\ 16 \overline{) 16} \cdots \text{余 } 0 \\ \underline{1} \cdots \text{余 } 1 \end{array}$$

$$\text{则 } (265)_{10} = (109)_{16}$$

若将十进制整数 265 转换成八进制，可写成如下形式：

$$\begin{array}{r} 8 \overline{) 265} \cdots \text{余 } 1 \\ 8 \overline{) 33} \cdots \text{余 } 1 \\ \underline{4} \cdots \text{余 } 4 \end{array}$$

$$\text{则 } (265)_{10} = (411)_8$$

若将十进制整数 265 转换成二进制，可写成如下形式：

$$\begin{array}{r} 2 \overline{) 265} \cdots \text{余 } 1 \\ 2 \overline{) 132} \cdots \text{余 } 0 \\ 2 \overline{) 66} \cdots \text{余 } 0 \\ 2 \overline{) 33} \cdots \text{余 } 1 \\ 2 \overline{) 16} \cdots \text{余 } 0 \\ 2 \overline{) 8} \cdots \text{余 } 0 \\ 2 \overline{) 4} \cdots \text{余 } 0 \\ 2 \overline{) 2} \cdots \text{余 } 0 \\ \underline{1} \cdots \text{余 } 1 \end{array}$$

$$\text{则 } (265)_{10} = (100001001)_2$$

(2) 十进制小数部分的转换

十进制数小数部分转换的方法,是采用逐次乘以基数,每次乘积的整数部分,就是这个基数的转换编码。转换的过程如下:

① 用基数乘以十进制小数部分。

② 保留乘积的整数部分(0也包括在内)作为结果的1位。

③ 重复执行上述过程,直到小数部分为0或达到转换要求保留小数的精度为止。这种转换过程,由于连续使用基数作乘法运算,故称为基数连乘法。

基数连乘法的转换过程可表述为:作乘法,取整数,从前向后排。

例如,若将十进制小数 0.125 转换成十六进制,可写成如下形式:

$$0.125 \times 16 = 2.0 \quad \text{积的整数部分} = 2$$

此时,小数部分已经为0,可得转换结果:

$$(0.125)_{10} = (0.2)_{16}$$

若将十进制小数 0.125 转换成八进制,可写成如下形式:

$$0.125 \times 8 = 1.0 \quad \text{积的整数部分} = 1$$

此时,小数部分已经为0,可得转换结果:

$$(0.125)_{10} = (0.1)_8$$

若将十进制小数 0.125 转换成二进制,可写成如下形式:

$$0.125 \times 2 = 0.25 \quad \text{积的整数部分} = 0$$

$$0.25 \times 2 = 0.5 \quad \text{积的整数部分} = 0$$

$$0.5 \times 2 = 1.0 \quad \text{积的整数部分} = 1$$

此时,小数部分已经为0,可得转换结果:

$$(0.125)_{10} = (0.001)_2$$

2. 任意进制转换成十进制

任意进制转换成十进制数时,取决于每一位对应的权值。其作法是,先将每一位数与该位对应的权相乘,然后将这些乘积加起来就得到等价的十进制数。

例如,将二进制数 100111.11 转换成十进制,可写成如下形式:

$$\begin{aligned} (100111.11)_2 &= 1 \times 2^5 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} \\ &= 32 + 4 + 2 + 1 + 0.5 + 0.25 \\ &= (39.75)_{10} \end{aligned}$$

若将八进制数 367.12 转换成十进制,可写成如下形式:

$$(367.12)_8 = 3 \times 8^2 + 6 \times 8^1 + 7 \times 8^0 + 1 \times 8^{-1} + 2 \times 8^{-2} = (247.15625)_{10}$$

若将十六进制数 3A.19 转换成十进制,可写成如下形式:

$$(3A.19)_{16} = 3 \times 16^1 + A \times 16^0 + 1 \times 16^{-1} + 9 \times 16^{-2} = (58.09765625)_{10}$$

3. 任意进制之间的转换

二进制与八进制、二进制与十六进制的位数之间存在着一种特殊规律,即 3 位二进制数对应 1 位八进制数;4 位二进制数对应 1 位十六进制数。也就是说:1 位八进制数可用 3 位二进制数表示,1 位十六进制数可用 4 位二进制数表示。表 1-2 给出了十进制、二进制、八进制、十六进制间的相互关系。由表 1-2 中可以看出,几种进制间的互换是很简单的。

表 1-2 几种进制数对照表

十进制	八进制	十六进制	二进制	十进制	八进制	十六进制	二进制
0	0	0	0000	9	11	9	1001
1	1	1	0001	10	12	A	1010
2	2	2	0010	11	13	B	1011
3	3	3	0011	12	14	C	1100
4	4	4	0100	13	15	D	1101
5	5	5	0101	14	16	E	1110
6	6	6	0110	15	17	F	1111
7	7	7	0111	16	20	10	10000
8	10	8	1000	17	21	11	10001

(1) 二进制与八进制间的互换

转换的方法:从二进制(八进制)小数点开始分别向左、向右,以 3(1)位为一组来分组,然后用八进制(二进制)符号记下每组对应的值。

例如,若将二进制数 100111.11 转换成八进制,可写成如下形式:

$$(100111.11)_2 = \underline{100} \underline{111} \underline{110} = (47.6)_8$$

若将八进制数 67.4 转换成二进制,可写成如下形式:

$$(67.4)_8 = \underline{110} \underline{111} \underline{100} = (110111.1)_2$$

(2) 二进制与十六进制间的互换

转换的方法:从二进制(十六进制)小数点开始分别向左、向右,以 4(1)位为一组来分组。然后用十六进制(二进制)符号记下每组对应的值。

例如,若将二进制数 100111.11 转换成十六进制,可写成如下形式

$$(100111.11)_2 = \underline{0010} \underline{0111} \underline{1100} = (27.C)_{16}$$

若将十六进制数 39.F 转换成二进制,可写成如下形式:

$$(39.F)_{16} = \underline{0011} \underline{1001} \underline{1111} = (11100.1111)_2$$

由此可见,用二进制表示数据时,其位数长,人们不容易记忆和阅读。而利用二进制与八进制或二进制与十六进制数之间一一对应的特点,十分容易地克服了这个问题。不同数制间的相互转换,在汇编语言中是很常用的。

三、计算机中正、负数的表示法

在计算机中,参加运算数值的“+、-”(正或负)符号也是用二进制表示的,并规定用 **0** 表示正数,用 **1** 表示负数,符号位被放置在数值最高位 MSB(即最左边)。此外,对于负数还采用反码或补码表示,这样表示的目的是将负数转化为正数,使减法操作转变为单纯的加法操作。目前,在计算机系统中均采用补码来表示负数。下面对计算机中的码制作些简要介绍。

1. 原码

用原码表示一个带符号的二进制数,其最高位为符号位,用 **0** 表示正数,用 **1** 表示负数,其余各位表示数值。

下面是两个二进制数及其在计算机中的原码表示。

$$\begin{array}{lll} \text{正数} & X = +1001100 & [X]_{\text{原}} = 01001100 \\ \text{负数} & X = -1001100 & [X]_{\text{原}} = 11001100 \end{array}$$

2. 反码

反码在计算机中,对于正数,符号位为“**0**”,数值部分保持不变;对于负数来说,除了在符号位上表示“**1**”外,其数值部分的各位都取它相反的数码,即:“**0**”变“**1**”、“**1**”变“**0**”。

下面是两个二进制数及其在计算机中的反码表示。

$$\begin{array}{lll} \text{正数} & X = +1001100 & [X]_{\text{反}} = 01001100 \\ \text{负数} & X = -1001100 & [X]_{\text{反}} = 10110011 \end{array}$$

3. 补码

补码在计算机中,对于正数,符号位为“**0**”,数值部分保持不变;对于负数来说,除了在符号位上表示“**1**”外,其数值部分的各位都取它相反的数码,然后在最低位加“**1**”。

下面是两个二进制数及其在机器中的补码表示。

$$\begin{array}{lll} \text{正数} & X = +1001100 & [X]_{\text{补}} = 01001100 \\ \text{负数} & X = -1001100 & [X]_{\text{补}} = [X]_{\text{反}} + 1 = 10110011 + 1 = 10110100 \end{array}$$

表 1-3 列出了 8 位带符号二进制数的原码、反码、补码对照表。注意表中的 $[+0]$ 、 $[-0]$ 的反码、补码及补码是 $[10000000]$ 的原码。

表 1-3 8 位带符号二进制数的原码、反码、补码对照表

十进制数	二进制数	原 码	反 码	补 码
+0	+0000000	00000000	00000000	00000000
+1	+0000001	00000001	00000001	00000001
+2	+0000010	00000010	00000010	00000010
⋮	⋮	⋮	⋮	⋮
+126	+1111110	01111110	01111110	01111110
+127	+1111111	01111111	01111111	01111111
-0	-0000000	10000000	11111111	00000000
-1	-0000001	10000001	11111110	11111111
-2	-0000010	10000010	11111101	11111110
⋮	⋮	⋮	⋮	⋮
-126	-1111110	11111110	10000001	10000010
-127	-1111111	11111111	10000000	10000001
-128	-10000000	无法表示	无法表示	10000000

四、带符号二进制数的运算

二进制数和十进制数的运算规则基本相同,所不同的是前者二进制运算是“逢 2 进 1”,后者十进制运算是“逢 10 进 1”。借位时,从高位借 1 到低位,前者当 2,后者当 10。但如果是带符号的二进制数运算,则情况就比较复杂,因为带符号二进制数有三种表示方法,其中反码用得较少,下面主要介绍在原码与补码运算中应注意的问题。

在原码运算时,首先要将符号与数值分开。例如,两数相加,先要判断两数的符号,如果同号,可以做加法,如果异号,实际要做减法,减后的差作为两数之和,和数的符号与绝对值较大的数的符号相同。两数相减也是一样,首先要判断两数符号,然后决定是相加还是相减,还要根据两数的大小与符号决定两数之差的符号。

如果是补码运算则不存在符号与数值分开的问题。在补码运算时,把符号位也看成数值,一起参加运算,而且加法运算就一定是相加,减法运算就一定是相减,因此在计算机中对带符号的数进行加减时,最好使用补码。

补码的运算规则是: $X_{\text{补}} + Y_{\text{补}} = (X + Y)_{\text{补}}$

设有两个数 x 和 y , $x = -89$, $y = +18$, 用补码表示如下:

$$x = 10100111\text{B} \quad (-89 \text{ 的补码})$$

$$y = 00010010\text{B} \quad (+18 \text{ 的补码})$$

若求 $x + y$, 可不用考虑两数的符号,直接相加,得出的和为 $x + y = 11000111\text{B}$ (-71 的补码),可见直接相加结果是正确的。

若求 $x - y$, 也可以直接相减,即

$$\begin{array}{rcl} x & = & 10100111\text{B} \quad (-89 \text{ 的补码}) \\ -y & = & 00010010\text{B} \quad (+18 \text{ 的补码}) \\ \hline x-y & = & 10010101\text{B} \quad (-107 \text{ 的补码}) \end{array}$$

可见直接相减结果也是正确的。

若求 $y-x$, 同样也用减法, 即

$$\begin{array}{rcl} y & = & 00010010\text{B} \quad (+18 \text{ 的补码}) \\ -x & = & 10100111\text{B} \quad (-89 \text{ 的补码}) \\ \hline y-x & = & \boxed{1}01101011\text{B} \quad (+107 \text{ 的补码}) \end{array}$$

也就是说做减法时, 不论两数符号如何, 其相减结果不论是数值还是符号都是正确的。

在上述 $y-x$ 算式中, 最高位发生的进(借)位只是因为字长为 8 位的计算机中, 若运算结果并未超出补码的记数范围($-128 \sim +127$), 这时的进(借)位被视为自然丢弃。计算机在运算中, 这种自然丢弃并不影响结果的正确性。

但要注意, 如果字长为 8 位的计算机, 其计算结果超过 8 位补码所允许的表示范围(即超出 $-128 \sim +127$), 则其进位称为溢出。溢出和自然丢弃是两种不同的概念。判别属于哪一种, 则要看计算结果第 7 位与第 8 位的进位情况, 如果第 7 位和第 8 位同时产生进位, 即所谓双进位, 则这种进位属于允许的自然丢弃。如果只有第 7 位或者只有第 8 位产生进位, 即只有单进位, 则这种进位属于溢出。溢出表示其数值超出计算机字长所能表示的范围, 运算结果必然是错误的, 因而也是不允许的。

应当注意的是, 溢出的概念主要是指带符号二进制数进行加减运算时可能产生的一种结果。对于无符号数, 第 8 位不是符号位, 只有第 8 位的进位才称为进位, 而不采用溢出这个概念。

对于无符号数的运算还应注意一点, 当两个无符号数相减时, 不允许用小的数去减大的数, 因为小减大它的差一定是负数, 无符号数的前提是没有符号, 显然也不允许有负数, 如果这样做, 减的结果也必然是错误的。

五、字符的编码

计算机除了处理数值信息外, 还需要处理大量的字符数据, 如字母、符号、汉字、图像和语音等, 这些信息在计算机中都是以编码的形式出现的。在计算机中常用的字符编码有 ASCII 码、BCD(Binary Coded Decimal)码、格雷码和奇偶校验码等, 这些编码在计算机内部也是用二进制表示的。

1. ASCII 码

目前, ASCII 码有 7 位和 8 位两种字符编码形式。常用的是 7 位 ASCII 码, 它包括 26 个大写和 26 个小写的英文字母、10 个数字, 以及一些专用字符。7 位编码的 ASCII 码, 实际也采用 8 位二进制, 但最高位置 0 用作校验, 故最多可表示 128 个字符(即 $2^7 = 128$)。表 1-4 中, 0~F 对应 ASCII 编码的低 4 位(表中横向, 编码范围 0000 ~ 1111), 0~7 对应编码的