

福建省高校计算机等级考试规划教材(三级)

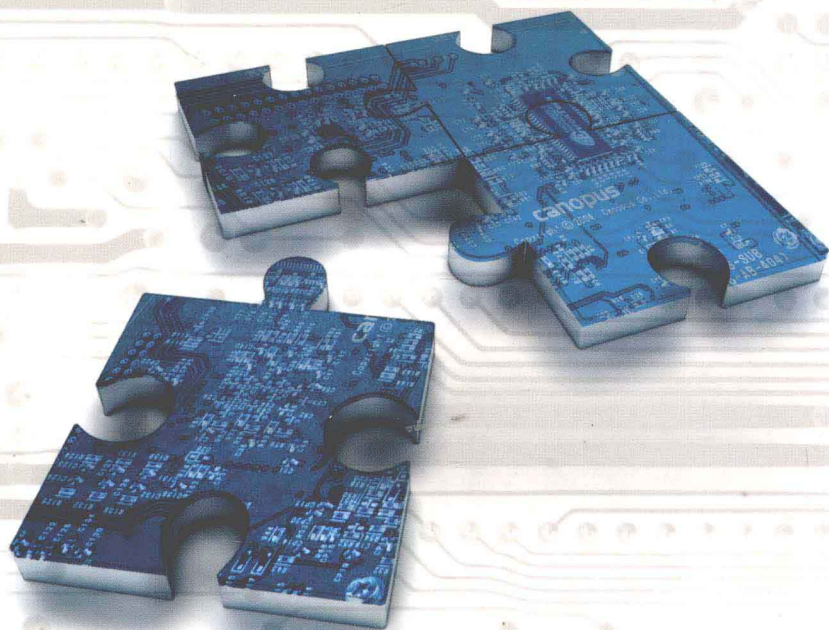
微机应用技术基础

Computer Application Technology

福建省高校计算机等级考试三级(偏硬)考试指导书·第一分册

福建省高校计算机教材编写委员会 组织编写

陈庆强 编著



厦门大学出版社
XIAMEN UNIVERSITY PRESS

国家一级出版社
全国百佳图书出版单位

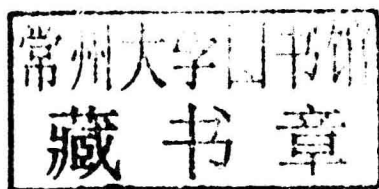
福建省高校计算机等级考试规划教材

微机应用技术基础

——福建省高校计算机等级考试三级(偏硬)

考试指导书·第一分册

陈庆强 编著



厦门大学出版社

图书在版编目(CIP)数据

微机应用技术基础/陈庆强编著. —厦门:厦门大学出版社, 2012. 4
福建省高校计算机等级考试规划教材
ISBN 978-7-5615-3841-8

I. ①微… II. ①陈… III. ①微型计算机-高等学校-教材 IV. ①TP36

中国版本图书馆 CIP 数据核字(2012)第 035856 号

厦门大学出版社出版发行

(地址:厦门市软件园二期望海路 39 号 邮编:361008)

<http://www.xmupress.com>

xmup@public.xm.fj.cn

三明市华光印务有限公司印刷

2012 年 4 月第 1 版 2012 年 4 月第 1 次印刷

开本:787×1092 1/16 印张:22.75

字数:553 千字 印数:1~2000 册

定价:40.00 元

如有印装质量问题请与承印厂调换

内容简介

本书是根据福建省高校计算机等级考试指导委员会审核批准的《福建省高等学校计算机应用水平等级考试三级(偏硬)考试大纲》(2010年修订版)要求编写而成的。内容包括微机系统概述、计算机中常用的基本逻辑器件、接口与中断基础、计算机网络基础、数据结构与算法基础、微机在测控系统中的应用以及嵌入式系统基本知识等。

本书内容丰富,通俗易懂,并附有多试模拟题,是参加福建省高等学校计算机应用水平三级(偏硬)考试的必备教材。同时,本书围绕实际应用开发所需的知识结构进行编写,突出实用性,可作为电子工程、自动控制、机电工程等相关专业学生的教学用书,也可供从事微机及其应用系统设计开发的广大工程技术人员学习参考。

前 言

福建省高校计算机等级考试分为一、二、三级三个级别,其中一、二级定位为课程考试,考生通过相应课程的学习,一般都可以顺利通过考试;而三级定位为计算机能力水平考试,且根据考生专业需求的差异,三级考试又分为“三级偏硬”和“三级偏软”两种。其中,“三级偏硬”的考试内容覆盖“计算机组成”、“微机原理及接口技术”、“计算机网络”、“数据结构与算法”、“微机测控技术”和“嵌入式系统”等多门计算机课程。对于电子工程、自动控制、机电工程等非计算机专业类的本科生而言,若能通过这一考试,无疑是如虎添翼。但是,要全面系统地学习和掌握这些课程显然也有较大的难度。正因为如此,多年以来,“三级偏硬”考试的通过率不甚理想,也挫伤了不少学生参加这项考试的积极性。为解决这一问题,福建省高校计算机应用水平等级考试指导委员会组织力量编写了《福建省高校计算机等级考试三级(偏硬)考试指导书》系列,本书是其中的第一分册。本书与《Intel 80X86 微机原理与接口技术》(余朝琨编著,厦门大学出版社 2009 年 10 月出版)结合,全面覆盖三级(偏硬)并选择 Intel 80X86 方向的考纲内容;与《MCS-51 微机原理与接口技术》(吴锤红编著,厦门大学出版社 2009 年 5 月出版)结合,全面覆盖三级(偏硬)并选择 MCS-51 系列单片机方向的考纲内容。同时,本书力图从计算机应用实际出发,以最简明扼要的形式,将相关课程的基础知识和主要内容展示给读者,从而帮助读者花尽量少的时间和精力顺利通过考试,也为读者今后从事计算机控制领域的工作打下坚实的基础。

本书由福建工程学院陈庆强副教授编写。福建工程学院计算机与信息科学系的蔡文培老师、徐翔老师分别为本书的第 6 章和第 4 章的编写提供了部分参考资料,龚小红老师和林芳老师帮助校对了第 4 章和第 5 章的内容。在此,向他们表示感谢。同时,感谢福建省教育厅和福建省高校计算机应用水平等级考试指导委员会的大力支持,也感谢厦门大学出版社在本书编辑和出版过程中给予的大力支持。

由于作者水平有限,所涉及的内容又极其广泛,本书难免存在疏误,恳请读者不吝赐教。

作者

2012 年春

目 录

第 1 章 微型计算机系统概述	1
1.1 计算机中数据的表示及运算	1
1.1.1 计算机中数据的表示	1
1.1.2 BCD 码	6
1.1.3 字符和汉字的编码	6
1.1.4 基本逻辑运算	9
1.2 微型计算机基本概念	9
1.2.1 微处理器、微型计算机、微型计算机系统	9
1.2.2 微型计算机的分类	10
1.2.3 微型计算机系统的主要技术指标	11
1.3 微型计算机的硬件组成	12
1.3.1 总线	12
1.3.2 微处理器	13
1.3.3 存储器	14
1.3.4 输入/输出设备和输入/输出接口	16
1.4 微型计算机工作基本原理	17
1.4.1 微型计算机的工作过程	17
1.4.2 微处理器的定时和基本操作	19
第 2 章 计算机中常用的基本逻辑器件	21
2.1 逻辑值与电平	21
2.2 TTL 与 CMOS 晶体管电路特点	21
2.2.1 TTL 系列集成电路型号	21
2.2.2 CMOS 逻辑门电路系列	22
2.2.3 CMOS 数字电路的特点	22
2.3 简单门电路	23
2.3.1 与门	24
2.3.2 或门	25
2.3.3 非门(反相器)	25
2.3.4 与非门	26
2.3.5 或非门	26

2.3.6 异或门	26
2.3.7 三态门与 OC 门	27
2.4 缓冲器与锁存器	28
2.4.1 缓冲器	28
2.4.2 锁存器	29
2.5 多路模拟开关与采样/保持器	30
2.5.1 多路模拟开关	30
2.5.2 采样/保持器	31
第3章 系统扩展与接口基础	33
3.1 I/O 接口概述	33
3.1.1 I/O 接口的基本功能	33
3.1.2 CPU 与 I/O 设备之间的接口信息	35
3.1.3 I/O 接口的分类	37
3.2 I/O 传送方式	37
3.2.1 无条件传送方式	38
3.2.2 查询传送方式	38
3.2.3 中断传送方式	39
3.2.4 存储器直接存取(DMA)方式	40
3.2.5 专用 I/O 处理机方式	42
3.3 I/O 端口编址方式	43
3.3.1 存储器映象编址方式	43
3.3.2 I/O 端口独立编址方式	43
3.3.3 PC 机 I/O 端口地址分配	44
3.4 I/O 端口的地址译码	45
3.4.1 门电路译码法	46
3.4.2 译码器译码法	46
3.4.3 比较器译码法	46
3.4.4 通用逻辑阵列 GAL 译码法	46
3.5 中断系统	46
3.5.1 概述	46
3.5.2 中断基本概念	47
3.5.3 中断优先权与中断嵌套	48
3.5.4 中断处理过程	49
3.5.5 中断响应的条件	49
3.6 定时计数器应用基础	50
3.6.1 概述	50

3.6.2 可编程定时/计数器	50
3.7 并行接口	51
3.7.1 概述	51
3.7.2 可编程并行输入/输出接口芯片 8255A	52
3.8 串行通信	64
3.8.1 串行通信基本知识	65
3.8.2 串行接口标准	67
3.9 IIC 总线	73
3.9.1 IIC 总线概述	73
3.9.2 IIC 总线工作原理	74
3.9.3 IIC 总线基本操作	75
第4章 计算机网络基础	78
4.1 计算机网络概述	78
4.1.1 计算机网络的基本概念	78
4.1.2 计算机网络的分类	80
4.1.3 计算机网络拓扑结构	81
4.1.4 网络传输介质	83
4.2 ISO/OSI 网络体系结构	85
4.2.1 协议和体系结构的概念	85
4.2.2 ISO/OSI 参考模型	86
4.3 计算机局域网基础	90
4.3.1 局域网概述	90
4.3.2 局域网体系结构	90
4.3.3 以太网技术	92
4.3.4 令牌传递网	94
4.3.5 高速局域网技术	96
4.3.6 无线局域网	98
4.3.7 交换式局域网	99
4.3.8 虚拟局域网	100
4.4 网络互联	101
4.4.1 网络互联类型	101
4.4.2 网络互联设备	102
4.5 TCP/IP 协议	107
4.5.1 TCP/IP 模型	107
4.5.2 网络接口层	108
4.5.3 网际层(互联网络层)	108

4.5.4 传输层	113
4.5.5 应用层	119
4.6 Internet 和 Intranet 基本知识	119
4.6.1 Internet 及其应用	120
4.6.2 Intranet 及其应用	125
4.7 C/S 结构与 B/S 结构	128
4.7.1 C/S 结构概述	128
4.7.2 B/S 结构概述	129
4.7.3 C/S 结构与 B/S 结构的比较	130
4.8 网络安全基础	131
4.8.1 网络安全概述	131
4.8.2 防火墙	132
4.8.3 密码技术	136
第 5 章 数据结构与算法基础	140
5.1 数据结构与算法的概念	140
5.1.1 数据结构基本概念	140
5.1.2 算法	141
5.1.3 算法分析	142
5.2 线性表	143
5.2.1 线性表的定义及操作	143
5.2.2 线性表的顺序存储结构	144
5.2.3 线性表的链式存储结构	147
5.3 栈	153
5.3.1 栈的定义及操作	153
5.3.2 栈的表示和实现	154
5.4 队列	155
5.4.1 队列的定义及操作	155
5.4.2 队列的表示和实现	156
5.5 树与二叉树	159
5.5.1 树的基本概念	159
5.5.2 二叉树及其存储表示	160
5.5.3 二叉树的遍历	163
5.5.4 霍夫曼树及其应用	166
5.6 查找	168
5.6.1 顺序查找	169
5.6.2 折半查找	170

5.6.3 分块查找	172
5.6.4 哈希表查找	173
5.7 排序	176
5.7.1 插入排序	177
5.7.2 选择排序	179
5.7.3 交换排序	180
5.7.4 归并排序	184
第6章 微机在测量和控制系统中的应用	187
6.1 微机测控系统概述	187
6.1.1 微机测控系统的特点	187
6.1.2 微机测控系统的组成	187
6.2 PC 机测控系统应用举例	191
6.3 MCS-51 单片机测控系统应用举例	193
6.3.1 系统要求	194
6.3.2 控制系统总体方案的确定	194
6.3.3 硬件设计	194
6.3.4 软件设计	197
6.4 微机测控系统抗干扰技术	201
6.4.1 硬件抗干扰技术	201
6.4.2 软件抗干扰技术	205
第7章 嵌入式系统简介	208
7.1 嵌入式系统概述	208
7.1.1 嵌入式系统的定义	208
7.1.2 嵌入式系统的特点	209
7.1.3 嵌入式系统的分类	209
7.1.4 嵌入式系统的应用	210
7.1.5 嵌入式系统的发展趋势	211
7.2 嵌入式系统的组成	212
7.2.1 硬件系统	212
7.2.2 软件系统	213
7.3 典型嵌入式微处理器简介	214
7.3.1 PIC 单片机简介	214
7.3.2 AVR 单片机简介	215
7.3.3 Freescale 单片机	216
7.3.4 ARM	217
7.4 嵌入式 Linux 操作系统概述	218

7.4.1 Linux 的特点	219
7.4.2 Linux 的内核体系结构	220
7.4.3 典型嵌入式 Linux 系统	220
附录 1 习题集	222
附录 2 福建省高等学校计算机应用水平等级考试三级(偏硬)考试大纲 (2010 年修订)	346

第 1 章 微型计算机系统概述

同普通计算机一样,微型计算机也是由运算器、控制器、存储器、输入设备和输出设备五大部分组成。通常,运算器和控制器合称为中央处理器(CPU),处理器和内存储器则合称为主机,而输入设备、输出设备和外存储器则统称为外部设备,或称为输入/输出设备。由于微型计算机具有体积小、质量轻、耗电量小、价格低、可靠性高等优点,所以它得到迅猛的发展,目前已经成为人类生产、生活乃至社会活动必不可少的工具。

本章介绍计算机中数据的表示和运算,微型计算机的基本概念、组成、工作过程及应用等。

1.1 计算机中数据的表示及运算

1.1.1 计算机中数据的表示

计算机最主要的功能是处理数值、文字、声音、图形和图像等信息,因此,计算机首先必须解决如何表示这些信息的问题。二进制中只有 0 和 1 两个符号,使用有两个稳定状态的物理器件就可以表示二进制的每一位,而制造有两个稳定状态的物理器件要比制造有多个稳定状态的物理器件容易得多,同时二进制的运算规则简单,所以为了能可靠、稳定地工作,同时便于设计与制造,计算机内部使用二进制代码作为信息的编码。二进制编码就是采用 0 和 1 两个基本符号,选用一定的组合原则,以表示大量复杂多样的信息。

1. 进位计数制及其转换

在采用进位计数的数字系统中,若使用 r 个基本符号表示数值,则称其为 r 进制, r 称为该数制的基数。对于不同的数制,它们的共同特点是:

每一种数制都有固定的符号集。例如,十进制计数制的基本符号有 10 个(0,1,2,...,9),二进制计数制的基本符号有 0 和 1 两个。

每一种数制都使用位置表示法。即处于不同位置的数符所代表的值不同,与它所在位置的权值有关。运算时“逢 r 进 1,借 1 当 r ”。

【例 1-1】 十进制数 1234.56 可表示为:

$$1234.56 = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10^1 + 4 \times 10^0 + 5 \times 10^{-1} + 6 \times 10^{-2}$$

即各种进位计数制中的权值就是基数的某次幂。因此,任何一种进位计数制表示的数都可以写成按权展开的多项式之和。常用进位计数制的表示如表 1-1 所示。

表 1-1 常用进位计数制的表示

进位制	二进制	八进制	十进制	十六进制
数符	0, 1	0, 1, 2, ..., 7	0, 1, 2, ..., 9	0, 1, 2, ..., 9, A, B, ..., F
运算规则	逢二进一	逢八进一	逢十进一	逢十六进一
基数	$r=2$	$r=8$	$r=10$	$r=16$
权	2^i	8^i	10^i	16^i
表示符	B	O	D	H

(1) 十进制记数法

在十进制记数制中,基本符号为 0,1,2,...,9,基数 $r=10$ 。无论多大或多小的数,都由这 10 个符号的组合来表示。

(2) 二进制记数法

在二进制记数制中,基本符号为 0 和 1,基数 $r=2$ 。

二进制数转换成十进制数的方法是:将二进制数的每一位数乘以它的权,然后相加,即可得其对应的十进制数值。

【例 1-2】 把二进制数 1101101.101 转换成相应的十进制数。

$$\begin{aligned}
 (1101101.101)_2 &= 1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + \\
 &\quad 0 \times 2^{-2} + 1 \times 2^{-3} \\
 &= 109.625
 \end{aligned}$$

十进制数转换成二进制数的方法是:整数部分和小数部分分别转换,然后再合并。整数部分采用“除 2 取余”法,小数部分采用“乘 2 取整”法。

(3) 八进制计数法

在八进制计数制中,基本符号为:0,1,2,...,7,基数 $r=8$ 。

八进制数转换成十进制数的方法是:将八进制数的每一位数乘以它的权,然后相加,即可得到其对应的十进制数值。

十进制数转换为八进制数的方法是:整数部分和小数部分分别转换,然后再合并。整数部分采用“除 8 取余”法,小数部分采用“乘 8 取整”法。

二进制数转换成八进制数的方法是:对于整数部分,从右到左将二进制数的每三位分为一组,若不够三位时,在最高位左边补 0;对于小数部分,从左到右将二进制数的每三位分为一组,若不够三位时,在最低位右边补 0,然后将每三位二进制数用一位八进制数替换即可。

【例 1-3】 $(1110101.1111)_2 = (001\ 110\ 101.111\ 100)_2 = (165.74)_8$ 。

(4) 十六进制计数法

在十六进制计数制中,基本符号为:0,1,2,...,9,A,B,...,F,基数 $r=16$ 。

十六进制数转换成十进制数的方法是:将十六进制数的每一位数乘以它的权,然后相加,即可得到其对应的十进制数值。

十进制数转换为十六进制数的方法是:整数部分和小数部分分别转换,然后再合并。整数部分采用“除 16 取余”法,小数部分采用“乘 16 取整”法。

二进制数转换成十六进制数的方法是:对于整数部分,从右到左将二进制数的每四位分为一组,若不够四位时,在最高位左边补0;对于小数部分,从左到右将二进制数的每四位分为一组,若不够四位时,在最低位右边补0,然后将每四位二进制数用一位十六进制数替换即可。

【例 1-4】 $(1110111111.11101)_2 = (0011\ 1011\ 1111.1110\ 1000)_2 = (3BF.E8)_{16}$ 。

2. 二进制运算规则

(1)加法运算法则:加法的进位规则是“逢二进一”。

$0+0=0, 0+1=1, 1+0=1, 1+1=0$ (向高位产生进位)。

(2)减法运算法则:减法的借位规则是“借一当二”。

$0-0=0, 1-1=0, 1-0=1, 0-1=1$ (向高位产生借位,借一当二)。

(3)乘法运算法则

$0 \times 0=0, 0 \times 1=0, 1 \times 0=0, 1 \times 1=1$ (二进制乘法运算可归结为“加法与移位”)。

(4)除法运算法则

$0 \div 1=0$ ($1 \div 0$ 与 $0 \div 0$ 无意义), $1 \div 1=1$ (二进制除法运算可归结为“减法与移位”)。

3. 原码、反码和补码

数值在计算机中表示的形式称为机器数,机器数对应的实际数值称为数的真值。我们知道,一个数包含数的符号、数值和小数点,而在计算机内部只能用0和1二个符号来表示一个数,为此,人们发明了原码、反码和补码等来表示数的符号和数值,而使用定点数和浮点数表示形式来解决小数点在机器内的表示问题。

机器数有无符号数和带符号数之分。无符号数即为正数,在机器数中没有符号位,所有二进制数位均表示数值;而对于带符号数,则使用原码、反码和补码等来表示,它们都使用最高位来表示符号位(用“0”表示该数为正数,用“1”表示该数为负数),其余位则表示数值,小数点的位置固定在某一位置。一般微机内存储器器的一个单元可存储8位二进制,所以,通常机器数的长度为8、16、24等8的倍数。为叙述方便,以下假设使用8位二进制数表示原码、反码和补码,并规定小数点的位置在机器数的最低有效数值位之后,即数为纯整数。

(1)原码表示法

在8位原码表示法中,使用最高位(位7)表示符号位,其余7位(位6~位0)表示数的绝对值。

【例 1-5】 写出+62和-62的8位原码。

$$[+62]_{\text{原}} = 00111110\text{B}$$

$$[-62]_{\text{原}} = 10111110\text{B}$$

原码表示法的特点有:

①原码与真值之间的转换简单。

把原码转换为真值只需根据其最高位确定真值的符号,数值部分相同。

如 $[00001111\text{B}]_{\text{原}}$ 对应的真值为: $+0001111\text{B} = 1111\text{B}$; $[10001111\text{B}]_{\text{原}}$ 对应的真值为: $-0001111\text{B} = -1111\text{B}$ 。

②数值0的原码有两种表示:00000000B和10000000B。

③8 位原码表示的数的范围为： $-127 \sim +127 (-2^7 + 1 \sim +2^7 - 1)$ 。

④加减运算复杂。

二个原码表示的数要进行加减运算时,不能直接按照二进制数加减规则进行,即 $[x]_{\text{原}} + [y]_{\text{原}} \neq [x+y]_{\text{原}}$ 。

如： $[+62]_{\text{原}} + [-62]_{\text{原}} = 00111110\text{B} + 10111110\text{B} = 11111100\text{B} \neq [62-62]_{\text{原}}$ 。

由于原码表示法中,加减运算复杂,所以现在已基本不被采用。

(2)反码表示法

在反码表示法中,也使用最高位表示符号位,其余位表示数值,对于正数,其数值部分与真值相同,对于负数,其数值部分为真值数值部分按位求反。

【例 1-6】 写出+62 和-62 的 8 位反码。

$$[+62]_{\text{反}} = 00111110\text{B}$$

$$[-62]_{\text{反}} = 11000001\text{B}$$

反码表示法的特点有:

①反码与真值之间的转换比补码简单,但比原码复杂

把反码转换为真值先根据其最高位确定真值的符号,若为正数,则数值部分相同;若为负数,则必须将反码数值部分按位求反。

如:反码表示的数 00001111B,其对应的真值为:0001111B = 1111B;反码表示的数 10001111B,其对应的真值为:-1110000B。

②0 的反码有两种表示:00000000B 和 11111111B。

③8 位反码表示的数的范围为： $-127 \sim +127 (-2^7 + 1 \sim +2^7 - 1)$ 。

④加减运算比原码简单,但比补码复杂。

二个反码表示的数要进行相加时,先直接按照二进制数相加规则进行,然后将最高位(符号位)产生的进位加到最低位。

如： $[+62]_{\text{反}} + [-60]_{\text{反}} = 00111110\text{B} + 11000011\text{B} = 00000001\text{B}$ (同时产生进位 1)。

修正： $00000001\text{B} + 1(\text{进位}) = 00000010\text{B}$,即为 $[62-60]_{\text{反}}$ 的反码。

反码表示法也已基本不被采用。

(3)补码表示法

在补码表示法中,也使用最高位表示符号位,其余位表示数值,对于正数,其数值部分与真值相同;对于负数,其数值部分为真值数值部分按位求反后在最低位加上 1。

【例 1-7】 写出+62 和-62 的 8 位补码。

$$[+62]_{\text{补}} = 00111110\text{B}$$

$$[-62]_{\text{补}} = 11000010\text{B}$$

补码表示法的特点有:

①补码与真值之间的转换相对复杂些。

把补码转换为真值先根据其最高位确定真值的符号,若为正数,则数值部分相同;若为负数,则必须将补码数值部分按位求反后在最低位加上 1。

如:补码表示的数 00001111B,其对应的真值为:0001111B = 1111B;补码表示的数

10001111B,其对应的真值为: -1110000B。

②0 的补码只有一种表示:00000000B。

③8 位补码表示的数的范围为: $-128 \sim +127 (-2^7 \sim +2^7 - 1)$ 。

④加减运算简单。

两个补码表示的数要进行相加减时,可连同符号位一起直接按照二进制数加减规则进行。

即有:

$$\begin{aligned}[X+Y]_{\text{补}} &= [X]_{\text{补}} + [Y]_{\text{补}} \\ [X-Y]_{\text{补}} &= [X]_{\text{补}} + [-Y]_{\text{补}}\end{aligned}$$

如: $[+62]_{\text{补}} + [-100]_{\text{补}} = 00111110\text{B} + 10011100\text{B} = 11011010\text{B} = [62-100]_{\text{补}}$ 。

这一特点意味着,在补码系统中,两个数要进行减法运算时可以转化为两个数进行加法运算操作,这样,在硬件内部实现时只需一套二进制加法器电路即可,相加时也不必考虑符号位,从而提高了数据处理的效率,降低了硬件的复杂性。

由于补码具有这些优点,所以目前被普遍采用,在没有特殊说明时,计算机中的有符号数都默认采用补码表示。

(4)溢出

对于 n 位二进制数表示的补码,其范围为: $-2^{n-1} \sim +2^{n-1} - 1$ 。两个补码表示的数要进行加减运算时,若结果超出它所能表示的范围,则称为溢出,当产生溢出时,其结果当然就是错误的。如在 8 位补码系统中: $[+100]_{\text{补}} + [+50]_{\text{补}} = 01100100\text{B} + 00110010\text{B} = 10010110\text{B}$,显然,结果是一个负数的补码,它不是 +150 的补码表示了,因为 +150 已超出 8 位补码所能表示的范围。

在实际应用中,要特别注意溢出的发生,判断二个补码表示的数进行相加减是否产生溢出的方法有:

① $OV = C_{n-1} \oplus C_{n-2}$,其中 C_{n-1} 和 C_{n-2} 分别表示运算时最高位和次高位产生的进位值,OV 表示溢出,“ $\oplus$ ”为异或运算符,即若 C_{n-1} 和 C_{n-2} 同时为 0 或同时为 1 时, $OV=0$,没有产生溢出;若 C_{n-1} 和 C_{n-2} 其中有一个为 0,另一个为 1 时, $OV=1$,产生溢出。

②同符号相减或异符号相加,不可能发生溢出;同符号相加或异符号相减,则有可能发生溢出。若两个正数相加或一个正数减去一个负数,结果却得到一个负数,则发生了溢出;若两个负数相加或一个负数减去一个正数,结果却得到一个正数,则也发生了溢出。

可以采用增加机器数的长度来解决溢出的问题。

对于正数,其原码、反码和补码的表示形式完全相同。

4. 定点数和浮点数

(1)定点数

定点数就是小数点的位置固定不变的数,其小数点的位置通常有两种约定方式:一种是约定小数点在最低有效数值位之后,表示一个定点纯整数;另一种是约定小数点在最高有效数值位之前,表示一个定点纯小数。

设用 n 位表示一机器数,则原码、反码和补码表示下的带符号数的范围如表 1-2 所示。

表 1-2 n 位原码、反码和补码表示的带符号数的范围

码制	定点整数	定点小数
原码	$-(2^{n-1}-1) \sim +(2^{n-1}-1)$	$-(1-2^{-(n-1)}) \sim +(1-2^{-(n-1)})$
反码	$-(2^{n-1}-1) \sim +(2^{n-1}-1)$	$-(1-2^{-(n-1)}) \sim +(1-2^{-(n-1)})$
补码	$-2^{n-1} \sim +(2^{n-1}-1)$	$-1 \sim +(1-2^{-(n-1)})$

(2)浮点数

定点数所能表示的数值范围较小,运算中容易发生溢出,因此要引入浮点数。浮点数是小数点位置不固定的数,它能表示更大范围的数。

在十进制表示法中,人们为了表示绝对值比较大和比较小的数,常使用科学计数法,即将数表示为 $m \times 10^e$ 的形式,同样,机器数也可借鉴这种表示形式,即将数表示为 $a = m \times 2^e$,其中 e 为阶码, m 为尾数。用阶码和尾数表示的数叫作浮点数,这种表示数的方法称为浮点表示法。在浮点表示法中,阶码为带符号的纯整数,尾数为带符号的纯小数。浮点数的表示格式如下:

阶符	阶码	数符	尾数
----	----	----	----

其中阶符和数符分别表示阶码和尾数的符号。浮点数所能表示的数值范围主要由阶码决定,所表示数值的精度则由尾数决定。对于 n 位表示的浮点数:阶码长度越长(尾数长度越短),其所能表示的数的范围越大,其精度就越低;阶码长度越短(尾数长度越长),其所能表示的数的范围越小,其精度就越高。

1.1.2 BCD 码

用 4 位二进制代码表示 1 位十进制数,称为二—十进制编码,简称 BCD 编码。因为 $2^4 = 16$,而十进制数只有 0~9 十个不同的数符,故有多种 BCD 编码。常用的 BCD 码使用 8421 码,即 4 个二进制位的权从高到低分别为 8、4、2 和 1,它与十进制数的对应关系如表 1-3 所示。

表 1-3 BCD 码与十进制数的对应关系

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

一般地,内存一个单元可存放 8 位二进制数。若用高四位和低四位分别表示 1 个十进制位,则称为压缩 BCD 码。如,十进制数 38 的非压缩 BCD 码可表示为:00111000B。若以 8 个二进制位中的低 4 位表示 1 个十进制位,高四位不使用,则称为非压缩 BCD 码。如,十进制数 38 的非压缩 BCD 码可表示为:XXXX0011 XXXX1000B(其中 X 可为 0 或 1)。

1.1.3 字符和汉字的编码

在计算机内,数字、字母、符号和汉字等信息最终都必须用二进制表示。