

大学经营管理专业教材

计算机概论

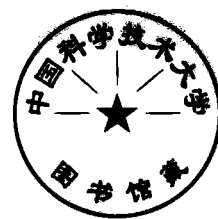
蒺志华 王裕明

上海科学普及出版社

大学经营管理专业教材

计 算 机 概 论

黄志华 王裕明



上海科学普及出版社

(沪)新登字第 305 号

责任编辑 陈泽加

大学经营管理专业教材

计算机概论

黄志华 王裕明

上海科学普及出版社出版

(上海曹杨路 500 号 邮政编码 200063)

新华书店上海发行所发行 上海长鹰印刷厂印刷

开本 787×1092 1/16 印张 16.25 字数 415000

1996 年 8 月第 1 版 1996 年 8 月第 1 次印刷

ISBN 7-5427-1057-5/TP·264 定价:22.00 元

前 言

今天,计算机已大踏步地走进寻常百姓家,很少有人对计算机这个名称感到陌生。但大部分人对汹涌而来的计算机应用浪潮,多少感到有些惶惑不安,在林林总总的计算机软件面前感到有点目不暇接,无所适从。在原有的软件还未充分理解、使用时,又有新的、功能更强的综合性软件推出。许多计算机用户往往不知道选择怎样的软件去完成他们自己的工作。有的用户是只知其然,不知其所以然。

人人都清楚,现在的科技时代,不会用计算机是不行的,可是怎么学?是不是学会操作一二种,或者更多的软件就算学会计算机了?这个问题困惑着大多数莘莘学子和想学会计算机的人。

任何实用性技术都有一个共同点,那就是技术越先进,对使用这种技术的人来讲就越方便,操作也就越简便。计算机软件的开发也正在朝这个方向发展。要掌握一种新推出的软件,对操作使用来说,也会越来越简单。

那么,什么是应该学的呢?编者认为,从工作需要出发,从研究课题出发,应该掌握必要的软件知识与操作方法,但更重要的是从基础知识和基础理论学起,形成计算机的总体概念,在此基础上再学习计算机的各种系统软件与应用软件。如果感兴趣,还可以学习更高深的理论。

本书的宗旨是讲述计算机理论的基础知识,为读者进一步学会和掌握计算机的应用打下基础。

本书适合高等学校、中等专业学校作为计算机课程的基础教材使用,也可作计算机等级考试或计算机应用能力考试的补充教材;计算机使用者也可以把它当作案头的常备参考书。

本书的第一、二、四章由王裕明编写;第三、五章由于毅红编写;第六章由康光毅编写;第七章由黄志华编写,最后一章由江滨编写。

编 者

1996年6月

大学经营管理专业教材编辑委员会

名誉主编：陆新葵

主 编：汪 泓

副 主 编：黄志华 徐克绍 陈泽加

编 委：王秀兰 王裕明 吴 清 陈 明

陈星德 杭 育 顾惠云 龚 平

崔晓明 葛开明 蒋仲刚

目 录

第一章 计算机的数制和码制	1
§ 1-1 数制及其转换	1
§ 1-1-1 十进制数	1
§ 1-1-2 二进制数	1
§ 1-1-3 八进制数和十六进制数	3
§ 1-2 原码、反码和补码	5
§ 1-2-1 机器数与字长	5
§ 1-2-2 机器数的几种码制表示	5
§ 1-3 数的定点表示和浮点表示	7
§ 1-3-1 数的定点表示	7
§ 1-3-2 数的浮点表示	7
§ 1-4 字符编码	8
习题一	9
第二章 计算机基本原理和组成	10
§ 2-1 计算机的特点和应用	10
§ 2-1-1 计算机的特点	10
§ 2-1-2 计算机的应用	10
§ 2-2 存储程序和程序控制原理	12
§ 2-2-1 程序和指令	12
§ 2-2-2 存储程序和程序控制原理	13
§ 2-2-3 冯·诺依曼型计算机的组成	13
§ 2-3 计算机硬件的基本组成	14
§ 2-3-1 中央处理器 CPU	14
§ 2-3-2 存储器	14
§ 2-3-3 输入输出设备	18
§ 2-4 计算机软件系统	19
§ 2-4-1 系统软件	19
§ 2-4-2 应用软件	20
§ 2-5 计算机系统	20
§ 2-6 计算机病毒防范和软件产权保护	20
§ 2-6-1 计算机病毒	20
§ 2-6-2 计算机病毒的防范	22
§ 2-6-3 计算机软件的产权保护	23

习题二	24
第三章 西文操作系统	25
§ 3-1 PC-DOS 及其发展	25
§ 3-1-1 PC-DOS	25
§ 3-1-2 DOS 的发展	26
§ 3-2 文件及文件目录	26
§ 3-2-1 文件	28
§ 3-2-2 文件目录	29
§ 3-3 键盘的使用	31
§ 3-4 DOS 的结构和启动	31
§ 3-4-1 DOS 的基本结构	32
§ 3-4-2 DOS 启动过程和内存使用状况	34
§ 3-4-3 DOS 的启动方法	35
§ 3-5 DOS 常用命令	35
§ 3-5-1 命令的类型和工作方式	37
§ 3-5-2 目录操作命令	41
§ 3-5-3 文件操作命令	49
§ 3-5-4 磁盘操作命令	53
§ 3-5-5 其他操作命令	56
§ 3-5-6 批处理文件	62
§ 3-6 DOS 常见错误及解决办法	62
§ 3-6-1 操作错误	65
§ 3-6-2 设备错误	67
习题三	72
第四章 高版本 DOS 系统	72
§ 4-1 内存管理	73
§ 4-1-1 扩展内存	74
§ 4-1-2 扩充内存	74
§ 4-1-3 高位内存和上位内存	75
§ 4-2 高版本 DOS 相对于 DOS3.3 的性能特点	76
§ 4-3 DOS Shell	76
§ 4-3-1 DOS Shell 基本知识	77
§ 4-3-2 使用菜单条	78
§ 4-3-3 改变查询方式	79
§ 4-3-4 文件和目录管理	82
§ 4-3-5 程序的管理和运行	85
§ 4-3-6 将程序与文件相联接	85
§ 4-3-7 退出 DOS Shell	86
§ 4-4 DOS 6.0 常用命令	

..... RECOVER & RESTORE	86
§ 4-4-2 UNFORMAT	88
§ 4-4-3 DEFRAGMENTER	88
§ 4-4-4 UNDELETE	89
§ 4-4-5 MEM	92
§ 4-4-6 MSAV	92
§ 4-4-7 EDIT	93
§ 4-4-8 MOVE	94
§ 4-4-9 DELTREE	94
§ 4-5 系统优化	95
§ 4-5-1 常用系统配置命令和设备驱动程序	95
§ 4-5-2 内存优化	98
§ 4-5-3 速度优化	100
习题四	102
第五章 汉字操作系统	104
§ 5-1 汉字的表示形式	104
§ 5-1-1 汉字内码	105
§ 5-1-2 汉字输入码	106
§ 5-1-3 汉字库	106
§ 5-2 Super-CCDOS 汉字操作系统	108
§ 5-2-1 运行环境	108
§ 5-2-2 主要功能模块简介	109
§ 5-3 系统的安装和启动	111
§ 5-3-1 系统的安装	111
§ 5-3-2 系统的启动	112
§ 5-4 Super-CCDOS 系统菜单的使用	114
§ 5-5 国标码、区位码输入法	118
§ 5-6 拼音输入法	119
§ 5-6-1 全拼双音	119
§ 5-6-2 双拼双音	120
§ 5-6-3 多字词汇	121
§ 5-6-4 提高输入速度的方法	121
习题五	122
第六章 WPS 文字信息处理系统	125
§ 6-1 概述	125
§ 6-1-1 硬件环境	125
§ 6-1-2 软件环境	125
§ 6-1-3 WPS 的组成	126
§ 6-2 WPS 的使用	126

§ 6-2-1 WPS 的基本概念	126
§ 6-2-2 WPS 的启动	127
§ 6-2-3 WPS 的主菜单功能	127
§ 6-2-4 WPS 的工作方式	128
§ 6-3 文本编辑	128
§ 6-3-1 进入编辑状态	129
§ 6-3-2 文件编辑状态的屏幕	130
§ 6-3-3 文本编辑状态下的命令菜单的使用	132
§ 6-3-4 菜单命令的结构	133
§ 6-3-5 编辑操作	135
§ 6-3-6 文件操作	138
§ 6-4 编辑技巧	139
§ 6-4-1 块操作	139
§ 6-4-2 寻找与替换	142
§ 6-4-3 制表格	143
§ 6-4-4 打印控制与打印输出	145
§ 6-4-5 多窗口编辑	149
§ 6-4-6 取日期与时间	150
习题六	151
第七章 数据库的原理与基本概念	152
§ 7-1 企业管理过程与管理决策	152
§ 7-1-1 企业管理过程	152
§ 7-1-2 企业的管理决策	154
§ 7-2 数据、信息、管理信息系统	155
§ 7-2-1 数据与信息	155
§ 7-2-2 数据处理技术的发展	156
§ 7-2-3 管理信息系统概述	158
§ 7-3 数据模型与数据库管理系统(DBMS)	158
§ 7-3-1 数据模型	159
§ 7-3-2 建立在数据模型上的数据库管理系统 DBMS	162
§ 7-4 关系数据库的基本理论	166
§ 7-4-1 关系模型的基本概念	166
§ 7-5 关系代数与关系演算	170
§ 7-5-1 关系代数运算	170
§ 7-5-2 关系演算	175
§ 7-6 关系的规范化	179
§ 7-6-1 关系模式的规范化	181
§ 7-7 数据库的保护	184
§ 7-7-1 安全性保护	184

§ 7-7-2 完整性保护	185
§ 7-7-3 并发控制	186
§ 7-7-4 数据库的恢复	186
§ 7-7-5 数据库的维护	187
§ 7-8 数据库系统	187
§ 7-8-1 数据库系统的软件资源	187
§ 7-8-2 数据库系统的结构	189
§ 7-8-3 数据库管理员	189
§ 7-9 数据库设计	190
§ 7-9-1 数据库设计的内容	190
§ 7-9-2 数据库的基本设计方法	191
习题七	193
第八章 关系数据库管理系统 FoxBASE+	195
§ 8-1 FoxBASE+的基本概况	195
§ 8-2 FoxBASE+的操作命令	200
§ 8-2-1 数据库的建立、编辑、使用	200
§ 8-2-2 数据库的重新组织、检索、统计、查询	216
§ 8-2-3 多重数据库的操作	225
§ 8-2-4 数据库的辅助功能	231
§ 8-2-5 函数	235
习题八	239
附录 I 双拼双音二级简码表	242
附录 II FoxBASE+操作过程中常见的错误表	243

第一章 计算机的数制和码制

§ 1-1 数制及其转换

§ 1-1-1 十进制数

人们日常使用的是十进制数,十进制数的特点是:

1. 使用十个不同的数字 0,1,2,...,9;
2. 逢十进一。

各数字在数据中的位置不同,代表数值的大小也不同。例如 123.45 这个数,3 在小数点左面个位,代表的数值是 3×10^0 ;1 在小数点左面百位上,代表的数值是 1×10^2 ;5 在小数点右面百分位上,代表的数值是 5×10^{-2} 。这个数可以写成:

$$123.45 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2}。$$

一般地,任意一个十进制数 $a_n a_{n-1} \cdots a_1 a_0 . a_{-1} \cdots a_{-m}$ 都可以表示成按权展开的多项式:

$$a_n a_{n-1} \cdots a_1 a_0 . a_{-1} \cdots a_{-m} = a_n \times 10^n + a_{n-1} \times 10^{n-1} + \cdots + a_1 \times 10^1 + a_0 \times 10^0 + a_{-1} \times 10^{-1} + \cdots + a_{-m} \times 10^{-m}。$$

其中 a_i 是 0~9 十个数字中的任意一个, m, n 为正整数,10 是进位基数。

§ 1-1-2 二进制数

虽然我们日常习惯使用十进制数,但它不适宜于计算机使用,计算机采用的是二进制数字系统。

一、数的二进制表示

二进制数的特点是:

1. 使用二个不同的数字 0 和 1;
2. 逢二进一。

计算机中采用二进制数,是因为具有下列好处:

1. 二进制数只有 0 和 1 两个数字,在电子元件中容易实现,如晶体管的导通与截止,电压的正负等。

2. 二进制数的运算规则简单,使得计算机的运算部件结构变得简单。

二进制数加法规则是:

$$0+0=0 \quad 0+1=1 \quad 1+0=1 \quad 1+1=10$$

二进制数乘法规则是:

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

3. 二进制数的二个数字 0 和 1 与逻辑代数的取值一致,从而使计算机可以进行逻辑运算。同时,在设计逻辑电路时,可以利用逻辑代数作为工具。

二、二进制数与十进制数的转换

1. 二进制数转换成十进制数

方法:乘权展开。

与十进制数相仿,一个二进制数可以表示成按权展开的多项式,只是二进制的进位基数是 2。

例 1-1-1:将二进制数 1010.01 转换成十进制数。

解: $(1010.01)_2$

$$=1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$$

$$=8+0+2+0+0+0.25$$

$$=10.25$$

2. 十进制整数转换成二进制数

方法:除 2 取余。

假设一个十进制整数 x 已被表示成二进制数 $(a_n a_{n-1} \cdots a_1 a_0)_2$, 其中 a_i 为 0 或 1 ($0 \leq i \leq n$)。则 x 可按二进制数的乘权展开如下式:

$$x = a_n 2^n + a_{n-1} 2^{n-1} + \cdots + a_1 2^1 + a_0 2^0$$

由于展开式除了最后一项 a_0 外,其余各项都为 2 的整数倍,因此 $x/2$ 的商为 $a_n 2^{n-1} + a_{n-1} 2^{n-2} + \cdots + a_1 2^0$, 余数为 a_0 。将上述的商 $a_n 2^{n-1} + a_{n-1} 2^{n-2} + \cdots + a_1 2^0$ 再除以 2, 得到的余数为 a_1 。依此类推,一直被 2 除下去,直到商为 0, 此时余数为 a_n 。

在上述不断被 2 除的时候得到的一组余数 $a_0, a_1, \cdots, a_n$ 即为 x 化为二进制数各位上的数字。

例 1-1-2:将十进制数 35 转换成二进制数。

解: $2 \overline{) 35}$ 余数

$$2 \overline{) 17} \cdots 1 = a_0$$

$$2 \overline{) 8} \cdots 1 = a_1$$

$$2 \overline{) 4} \cdots 0 = a_2$$

$$2 \overline{) 2} \cdots 0 = a_3$$

$$2 \overline{) 1} \cdots 0 = a_4$$

$$0 \cdots 1 = a_5$$

得: $35 = (100111)_2$

3. 十进制小数转换成二进制小数

方法:乘 2 取整。

假设一个十进制小数 y 已被表示成二进制小数 $(0. a_{-1} a_{-2} \cdots a_{-m})_2$, 其中 a_{-i} 为 0 或 1 ($0 \leq i \leq m$)。则 y 可按二进制数的乘权展开如下式:

$$y = a_{-1} \times 2^{-1} + a_{-2} \times 2^{-2} + \cdots + a_{-m} \times 2^{-m}$$

将 y 乘 2 后得到整数部分为 a_{-1} , 小数部分为 $a_{-2} \times 2^{-1} + \cdots + a_{-m} \times 2^{-m+1}$ 。再将小数部分乘 2 后能得到整数部分 a_{-2} , 依此类推, 小数部分反复被 2 乘, 得到的整数 $a_{-1}, a_{-2}, \cdots, a_{-m}$ 即为十进制小数 y 化为二进制小数各位上的数字。

例 1-1-3:将十进制小数 0.375 转换为二进制数。

解: 0.375

$$\begin{array}{r} *) \quad 2 \\ \hline 0.750 \quad a_{-1}=0 \\ *) \quad 2 \\ \hline 1.500 \quad a_{-2}=1 \\ *) \quad 2 \\ \hline 1.000 \quad a_{-3}=1 \end{array}$$

得: $0.375 = (0.011)_2$

值得注意的是一个有限十进制小数并非一定能够转换成一个有限二进制小数,即小数部分可能永远不等于0,这时可按要求进行到某一精度为止。

例 1-1-4: 将十进制小数 0.7 转换成二进制小数。

解: 0.7

$$\begin{array}{r} *) \quad 2 \\ \hline 1.4 \quad a_{-1}=1 \\ *) \quad 2 \\ \hline 0.8 \quad a_{-2}=0 \\ *) \quad 2 \\ \hline 1.6 \quad a_{-3}=1 \\ *) \quad 2 \\ \hline 1.2 \quad a_{-4}=1 \\ *) \quad 2 \\ \hline 0.4 \quad a_{-5}=0 \\ *) \quad 2 \\ \hline 0.8 \quad a_{-6}=0 \end{array}$$

得: $0.7 = (0.101100\cdots)_2$

如果一个十进制数既有整数,又有小数,则可以将整数部分和小数部分分别进行转换,然后再把两部分结果合并起来。

例 1-1-5: 将十进制数 35.375 转换成二进制数。

解: 因为 $35 = (100011)_2$

$$0.375 = (0.011)_2$$

所以 $35.375 = (100011.011)_2$

§ 1-1-3 八进制数和十六进制数

一个数值用二进制表示时,所需位数较多,造成读写不便。因此,在有关计算机的讨论中,人们还经常使用八进制数和十六进制数。

一、八进制数

八进制数的特点是:

1. 使用八个不同的数字:0,1,2,3,4,5,6,7。

2. 逢八进一

二、二进制数与八进制数的转换

三位二进制数,正好完全表示了八进制数的八个数字,它们之间的对应关系如表 1-1。

表 1-1 二进制数与八进制数、十六进制数的对应关系

八进制数	对应二进制数	十六进制数	对应二进制数
0	000	0	0000
1	001	1	0001
2	010	2	0010
3	011	3	0011
4	100	4	0100
5	101	5	0101
6	110	6	0110
7	111	7	0111
		8	1000
		9	1001
		A	1010
		B	1011
		C	1100
		D	1101
		E	1110
		F	1111

二进制数转换成八进制数的方法是:从小数点开始,分别向左、向右,每三位二进制数为一组,转换成一位八进制数。若小数点左侧位数不是 3 的倍数,则最左侧用 0 补足;若小数点右侧位数不是 3 的倍数,则最右侧用 0 补足。

例 1-1-6:将二进制数 10110111.01101 转换成八进制数。

解: $\underbrace{010110111}_{2\ 6\ 7}.\underbrace{011010}_{3\ 2}$

所以 $(10110111.01101)_2 = (267.32)_8$ 。

反过来,八进制数转换成二进制数的方法是:将每位八进制数用三位二进制数来表示,其整数部分最左侧的 0 或者小数部分最右侧的 0 可以删去。

例 1-1-7:将八进制数 32.54 转换成二进制数。

解: $\begin{array}{ccccccc} 3 & 2 & . & 5 & 4 & & \\ \downarrow & \downarrow & & \downarrow & \downarrow & & \\ 0 & 1 & 1 & 0 & 1 & 0 & . & 1 & 0 & 1 & 1 & 0 & 0 \end{array}$

所以 $(32.54)_8 = (11010.1011)_2$

三、十六进制数

十六进制数的特点是:

1. 使用十六个不同的数字:0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F;
2. 逢十六进一。

四、二进制数与十六进制数的转换

四位二进制数正好完全表示十六进制的十六个数字,它们之间的对应关系如表 1.1。

二进制数转换成十六进制数的方法是:从小数点开始,分别向左、向右,每四位二进制数为一组,用一位十六进制数来表示,若小数点左(右)侧位数不是 4 的倍数,则最左(右)侧用 0 补足。

例 1-1-8:将二进制数 10110111.01101 转换成十六进制数。

$$\text{解: } \underbrace{10110111}_B . \underbrace{01101000}_8$$

所以 $(10110111.01101000)_2 = (B7.68)_{16}$ 。

反过来,十六进制数转换成二进制数的方法是:将每位十六进制数用四位二进制数来表示,其整数部分最左侧的0或者小数部分最右侧的0可以删去。

例 1-1-9:将十六进制数 1A.C 转换成二进制数。

$$\begin{array}{ccccccc} \text{解:} & & 1 & & A & & . & & C \\ & & \downarrow & & \downarrow & & & & \downarrow \\ & 0 & 0 & 0 & . & 1 & 1 & 0 & 1 & 0 & . & 1 & 1 & 0 & 0 \end{array}$$

所以 $(1A.C)_{16} = (11010.11)_2$ 。

§ 1-2 原码、反码和补码

§ 1-2-1 机器数与字长

数有正负之分,在计算机中数的符号是用数码来表示的。一般情况下,用0表示正,用1表示负,通常符号位放在数的最高位。例如, $x_1 = (+1011011)_2$, $x_2 = (-1011011)_2$, 它们在机器中表示为:

$$x_1: \boxed{01011011}$$

$$x_2: \boxed{11011011}$$

其中最左边一位代表符号位,连同符号位在一起作为一个数,称为机器数;而它的数值称为机器数的真值。

表示机器数的二进制位的位数称为机器数的字长。字长取决于计算机的硬件,通常情况下,字长有8位、16位、32位、64位等。

§ 1-2-2 机器数的几种码制表示

一、原码

设 x 为整数,以 $[x]_{\text{原}}$ 表示 x 的原码,则

$$[x]_{\text{原}} = \begin{cases} x, & 0 \leq x \leq 2^{n-1} \\ 2^{n-1} - x, & -2^{n-1} < x \leq 0 \end{cases}$$

其中 n 是机器数字长。

例 1-2-1:求 $x = +1010011$ 和 $y = -1010011$ 的原码,设字长 $n = 8$ 。

$$\text{解: } [x]_{\text{原}} = 01010011$$

$$\begin{aligned} [y]_{\text{原}} &= 2^7 - (-1010011) \\ &= 11010011 \end{aligned}$$

求一个数的原码只需在其真值的最高位加上一个符号位,正号加0,负号加1。

原码的优点是直观,与真值转换方便,但做加减法较复杂,当两个异号的数相加时,先要比较两数绝对值的大小,然后相减,结果值的符号取绝对值大的数的符号。

例如, $[-8]_{\text{原}} = 10001000$, $[+6]_{\text{原}} = 00000110$, 由于 $|-8| > |+6|$, 所以 $[(-8) + (+6)]_{\text{原}}$ 的符号位取 $[-8]_{\text{原}}$ 的符号位1。取掉符号位后做减法:

$$0001000-0000110=0000010$$

再加上符号位,得到 $[(-8)+(+6)]_{\text{原}}=10000010$

二、反码

一个整数 x 的反码定义为:

$$[x]_{\text{反}} = \begin{cases} x, & 0 \leq x < 2^{n-1} \\ (2^n - 1) + x, & -2^{n-1} < x \leq 0; \end{cases}$$

其中 n 为机器数字长。

例 1-2-2: 求 $x=+1010011$ 和 $y=-1010011$ 的反码, 设字长 $n=8$ 。

解: $[x]_{\text{反}} = 01010011$

$$[y]_{\text{反}} = 2^8 - 1 + (-1010011)$$

$$= 10101100$$

一个数, 如果为正数, 其反码与原码相同; 如果是负数, 则除了符号位取 1 外, 将真值各位求反(用 $\bar{x}$ 表示对 x 取反), 即 1 换成 0, 0 换成 1。例如:

$$[y]_{\text{反}} = 1\bar{1}\bar{0}\bar{1}\bar{0}\bar{0}\bar{1}\bar{1}$$

$$= 10101100$$

三、补码

一个整数 x 的补码定义为:

$$[x]_{\text{补}} = \begin{cases} x, & 0 \leq x < 2^{n-1} \\ 2^n + x, & -2^{n-1} \leq x < 0 \end{cases}$$

其中 n 为机器数字长。

例 1-2-3: 求 $x=+1010011$ 和 $y=-1010011$ 的补码, 设字长 $n=8$ 。

解: $[x]_{\text{补}} = 01010011,$

$$[y]_{\text{补}} = 2^8 + (-1010011)$$

$$= 10101101$$

对一个正数, 它的补码与原码相同; 对一个负数, 则除了符号位取 1 外, 将真值各位求反, 然后再加 1。例如:

$$[y]_{\text{补}} = 1\bar{1}\bar{0}\bar{1}\bar{0}\bar{0}\bar{1}\bar{1} + 1$$

$$= 10101101$$

补码具有下面的特点:

$$[x \pm y]_{\text{补}} = [x]_{\text{补}} \pm [y]_{\text{补}}$$

例 1-2-4: 设 $x=2, y=-4$, 利用补码来计算 $x+y$ 的值。设机器数字长为 8。

解: $[x]_{\text{补}} = 00000010$

$$[y]_{\text{补}} = 11111100$$

$$[x]_{\text{补}} + [y]_{\text{补}}$$

$$= 00000010 + 11111100$$

$$= 11111110$$

$$= [-2]_{\text{补}}$$

$$= [x+y]_{\text{补}}$$

即 $x+y=-2$ 。

由该例子可以看出,两个数的补码进行加减法的结果,就是这两个数做加减法结果值的补码。因此用补码表示的数值可以连同符号位直接进行加减法运算。

对一个用补码表示的数要获得它的真值,只需再作一次求补运算。例如, $[-2]_{补} = 11111110$,对它求补为:

$$11111110 + 1 = 10000010$$

即为-2。

§ 1-3 数的定点表示和浮点表示

计算机处理数据时,除了考虑数制和数的各种码制表示外,还要考虑小数点的位置。如果将小数点固定在某一位置,称为定点表示法;如果小数点位置可以任意移动,则称为浮点表示法。

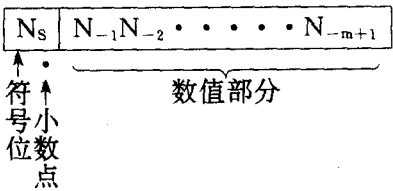
§ 1-3-1 数的定点表示

一、定点小数格式

小数点的位置固定在最高数据位的左边,一般 m 位定点小数(包括符号位)格式为:

$$N = N_s, N_{-1}N_{-2} \cdots N_{-m+1}$$

在计算机中格式为:



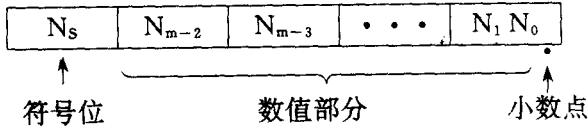
m 位定点小数(补码表示)能表示数的范围是 $-1 \sim 1 - 2^{-m+1}$ 。

二、定点整数格式

小数点的位置固定在最低数据位的右边,一般 m 位定点整数(包括符号位)格式为:

$$N = N_s N_{m-2} N_{m-3} \cdots N_1 N_0$$

在计算机中格式为:



m 位定点整数(补码表示)能表示数的范围是 $-2^{m-1} \sim 2^{m-1} - 1$ 。

在定点数表示中,一旦机器设计好,小数点位置就固定了,因此小数点在机器中是隐含的。

§ 1-3-2 数的浮点表示

定点数所能表示的数的范围,在许多应用领域是不够的。例如电子的质量为 9×10^{-28} 克,用定点数是无法表示的,但用浮点数表示是很方便的。

一个数 N 用浮点形式表示可以写成:

$$N = M \cdot R^E$$

M 表示尾数, E 表示指数, R 表示基数。基数一般取 2, 8, 16。一旦机器定义好基数值,就