

应用与开发 汉字信息处理的 微型计算机

孙俊逸 李邦畿 冯刚 程正军 编著



汉

人民邮电出版社

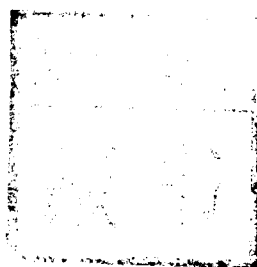
CWS CWS CWS CWS CWS CWS

1.12
Y/1

TP391.12
CTV/1

微型计算机汉字信息处理的 应用与开发

孙俊逸 李邦畿 编著
冯 刚 程正军



025384

人民邮电出版社

登记证号(京)143号

内 容 简 介

本书系统地介绍了微型计算机汉字信息处理应用与开发的基本方法、应用软件及数据库管理系统 FOXBASE⁺。全书包括三个部分共八章。第一章和第二章介绍了微型计算机的组成、DOS 操作系统、计算机软件和数据安全。第三、四章重点介绍了汉字信息处理及汉字软件 C-Wordstar 和 WPS。第五章至第八章介绍汉字 FOXBASE⁺ 程序设计方法与技巧、应用程序设计方法及实例。

本书内容充实,实用性强,应用面广,可作为高等院校、中等学校、微机应用培训班的教材,也可供计算机管理人员学习参考。

52.6/14

微型计算机汉字信息处理的应用与开发

Weixing Jisuanji Hanzi Xinxi Chuli De Yingyong Yu Kaifa

孙俊逸 李邦畿 冯刚 程正军 编著

责任编辑 赵桂珍

*

人民邮电出版社出版发行

北京东长安街 27 号

北京冶金出版社印刷厂印刷

新华书店总店科技发行所经销

*

开本:787×1092 1/16 1994 年 2 月 第 一 版

印张:17.75 1994 年 2 月 北京第 1 次印刷

字数:437 千字 印数:1—8 000 册

ISBN7-115-05104-6/TP·085

定价:15.00 元

前 言

近年来,计算机技术已取得惊人的迅猛发展,使得计算机渗透到各个领域,特别是在企、事业管理及办公自动化方面的应用前景更令人注目。为了适应微型计算机开发与应用的需要,我们根据教学实践编写了这本书。

本书包括三部分内容。第一部分(第一、二章)为微型计算机(以下简称微型机)基本知识,介绍了PC系列微型机DOS、CC-DOS常用操作命令及使用方法,汉字输入方法等内容。第二部分(第三、四章)为汉字信息处理及汉字编辑、排版软件,重点介绍了汉字信息处理技术、汉字编辑软件C-Wordstar和汉字排版软件WPS。其中汉字编辑软件Wordstar可进行多种文书的输入、编辑和打印工作,用户可按屏幕上所提供的功能菜单和提示说明,边阅读边操作。文字处理软件WPS集编辑、排版与打印为一体,深受用户欢迎。WPS的功能与四通MS-2401打字机兼容,初学者可利用帮助信息和功能菜单进行操作。第二部分的最后介绍了微型机排版基础知识。第三部分(第五至八章)为汉字FOXBASE⁺的编程方法和技巧,介绍了应用程序的方法及实例。汉化的FOXBASE⁺可在CC-DOS操作系统下运行,对dBASE⁺熟悉的用户可以方便地将dBASE⁺程序搬到FOXBASE⁺环境下运行。

本书内容充实,实用性强,适应面广,各章、节内容自成体系,读者可根据需要选学有关内容。本书可作为高等院校各专业、中等学校及培训班的教材,也可供计算机管理人员学习参考。

参加本书编写的有冯刚(第一章1.1~1.4、第二章、第三章3.1~3.4)、孙俊逸(前言、第一章1.5~1.6、第三章3.2~3.3、第四章、附录)、李邦畿(第五章、第六章、第八章)、程正军(第七章)。全书由孙俊逸、李邦畿最后修改定稿。

在本书的编写过程中,湖北大学计算机科学系及华中师范大学计算机科学系的老师们给予了大力支持,在此表示衷心感谢。

书中的不妥之处,恳请读者及从事计算机教学及应用研究的同仁批评指正。

编 者

目 录

第一章 微型计算机基础知识	1
1.1 计算机系统组成	1
1.1.1 计算机系统的基本构成	1
1.1.2 计算机硬件系统	1
1.1.3 计算机软件系统	2
1.1.4 程序设计语言	3
1.2 计算机运算基础	3
1.2.1 二进制数字系统	3
1.2.2 数制间的转换	4
1.2.3 逻辑运算	6
1.2.4 计算机中数的表示	7
1.2.5 数字编码与字符编码	8
1.3 微型计算机的结构	8
1.3.1 微型计算机系统	8
1.3.2 微型计算机的基本结构	9
1.4 IBM PC 系列微型计算机系统	10
1.4.1 IBM PC 及其兼容机	10
1.4.2 PC 系统的硬件配置	10
1.4.3 PC 系统的软件配置	11
1.5 计算机软件和数据安全	12
1.5.1 实体安全	12
1.5.2 软件安全	12
1.5.3 数据安全	13
1.5.4 运行安全	14
1.6 计算机键盘指法练习	19
1.6.1 计算机键盘结构	19
1.6.2 英文字母的使用频度	20
1.6.3 击键的姿势	20
1.6.4 键盘指法分区	21
1.6.5 键盘指法练习	22
第二章 磁盘操作系统	25
2.1 磁盘操作系统 DOS 概述	25
2.1.1 什么是操作系统	25
2.1.2 DOS 的基本结构	25
2.1.3 DOS 的版本	25
2.2 DOS 的使用	26

2.2.1	DOS 的文件	26
2.2.2	DOS 的启动	29
2.2.3	DOS 键盘的使用	31
2.2.4	DOS 的常用命令	33
2.2.5	DOS 的批命令	43
2.2.6	系统配置文件 CONFIG.SYS	46
2.3	CC-DOS 简介	49
2.3.1	CC-DOS 的主要特性	49
2.3.2	CC-DOS 的使用环境	49
2.3.3	CC-DOS 的安装与启动	50
2.3.4	CC-DOS 的使用	52
第三章	汉字信息处理	55
3.1	汉字代码和汉字字库	55
3.1.1	汉字代码系统	55
3.1.2	汉字输入码	55
3.1.3	汉字内部码	56
3.1.4	汉字交换码	56
3.1.5	汉字地址码	57
3.1.6	汉字字形码	57
3.1.7	汉字库	58
3.2	拼音码输入法	59
3.2.1	拼音码输入法	59
3.2.2	拼音双音输入法	60
3.3	五笔字型输入法	62
3.3.1	五笔字型编码基础	62
3.3.2	五笔字型键盘字根总表	66
3.3.3	五笔字型单字输入编码规则	67
3.3.4	简码、重码、容错码	69
3.3.5	词语输入	72
3.3.6	万能学习键	72
3.4	汉字的输出	73
3.4.1	汉字的显示输出	73
3.4.2	汉字的打印输出	74
第四章	汉字文书处理软件	76
4.1	汉字编辑软件 Wordstar	76
4.1.1	C-Wordstar 的启动	76
4.1.2	文书的编辑和修改	79
4.1.3	编辑排版	87
4.1.4	退出编辑	90
4.1.5	表格的构成	91

4.1.6 文书的打印及字型	94
4.1.7 非文书文件的编辑	99
4.2 汉字排版软件 WPS	101
4.2.1 WPS 的使用	101
4.2.2 文书的编辑和修改	106
4.2.3 文件操作	109
4.2.4 查找与替换	111
4.2.5 块操作	113
4.2.6 文书的格式编排及制表	116
4.2.7 窗口功能	119
4.2.8 设置打印控制符	121
4.2.9 模拟显示与打印输出	126
4.2.10 WPS 与 Wordstar 控制命令比较	128
4.3 微型计算机排版系统	131
4.3.1 系统硬件配置与选型	131
4.3.2 华光 IV 型排版软件的构成	133
4.3.3 华光排版系统的启用和操作	134
4.3.4 微型计算机排版的工艺流程	137
4.3.5 版面与版心尺寸	138
第五章 数据库系统基础知识	140
5.1 数据库系统	140
5.1.1 历史回顾	140
5.1.2 数据库系统	141
5.1.3 数据模型和数据库的分类	142
5.2 数据库的应用前景	144
5.3 汉字 FOXBASE ⁺ 的特点和运行环境	145
5.3.1 汉字 FOXBASE ⁺ 文件	146
5.3.2 汉字 FOXBASE ⁺ 特点	147
5.3.3 系统组成	148
5.3.4 系统的运行环境和安装	149
5.4 汉字 FOXBASE ⁺ 的常量、内存变量、表达式和函数	149
5.4.1 FOXBASE ⁺ 的数据类型	149
5.4.2 常量和变量	150
5.4.3 表达式	151
5.4.4 FOXBASE ⁺ 命令规则	153
5.4.5 汉字 FOXBASE ⁺ 函数	154
第六章 汉字 FOXBASE⁺数据库的建立和操作	164
6.1 数据库文件的建立	165
6.1.1 定义数据库结构	165
6.1.2 数据库文件的数据输入	167

6.1.3 数据库文件的打开	168
6.2 数据库结构的操作	169
6.2.1 查看数据库结构	169
6.2.2 拷贝和修改库文件结构	170
6.2.3 修改和更新库文件结构	171
6.3 数据库记录的操作	172
6.3.1 记录的指针及定位	172
6.3.2 记录的输出	173
6.3.3 记录的删除和恢复	174
6.3.4 记录的编辑、插入和翻阅	175
6.3.5 记录的添加、拷贝和修改	176
6.4 索引文件	179
6.4.1 分类	179
6.4.2 索引文件	180
6.4.3 索引文件的打开	181
6.4.4 重建索引	182
6.4.5 数据的查询	183
6.5 统计和汇总	185
6.5.1 求和	185
6.5.2 求平均值	185
6.5.3 统计记录数	186
6.5.4 汇总	186
6.6 多重数据库操作	187
6.6.1 工作区的基本概念	187
6.6.2 工作区的选择和互访	188
6.6.3 建立两个工作区间的联系	189
6.6.4 根据另一数据库进行更新操作	190
6.6.5 数据库间的连接	191
6.7 辅助命令和参数的设置	193
6.7.1 有关内存变量的操作	193
6.7.2 磁盘文件操作命令	197
6.7.3 参数的设置	198
第七章 汉字 FOXBASE⁺ 程序设计	202
7.1 程序设计概述	202
7.2 命令文件的建立和执行	203
7.2.1 命令文件的建立	203
7.2.2 命令文件的执行	204
7.3 顺序程序设计	205
7.4 分支程序设计	205
7.4.1 简单分支	205

7.4.2 多重分支	207
7.5 循环程序设计	209
7.5.1 循环语句用法	209
7.5.2 循环终止	211
7.6 输入输出程序设计	212
7.6.1 简单输入输出程序设计	212
7.6.2 格式输入、输出	214
7.6.3 屏幕输入输出格式设计	217
7.7 过程和过程文件	218
7.7.1 过程	218
7.7.2 过程文件	219
7.8 用户定义函数	220
7.9 屏幕菜单设计	221
7.9.1 菜单制作的一般方法	222
7.9.2 自由显示方式菜单	223
7.9.3 下拉式菜单	224
7.10 报表制作	225
7.10.1 固定格式报表制作	226
7.10.2 自由格式报表制作	228
7.11 其它程序设计技巧	232
7.11.1 用程序方式修改库结构	232
7.11.2 口令的设置	235
7.11.3 如何加快程序的运行速度	236
7.12 FOXBASE+程序调试技巧	237
7.12.1 重现命令执行次序	238
7.12.2 跟踪程序执行	239
7.12.3 设置断点	239
7.13 编译 FOXBASE+程序文件	240
第八章 应用程序开发	242
8.1 应用程序设计基本方法	242
8.1.1 系统分析阶段	242
8.1.2 系统设计阶段	242
8.1.3 系统实施阶段	242
8.1.4 系统运行、维护与评价阶段	242
8.2 应用程序设计实例	242
8.2.1 系统分析和设计	243
8.2.2 程序清单	246
附录一 ASCII 码字符集	259
附录二 FOXBASE+命令清单	260
附录三 FOXBASE+函数	270

第一章 微型计算机基础知识

1.1 计算机系统组成

1.1.1 计算机系统的基本构成

计算机系统包括硬件系统和软件系统两大部分。其基本构成如图 1-1 所示。

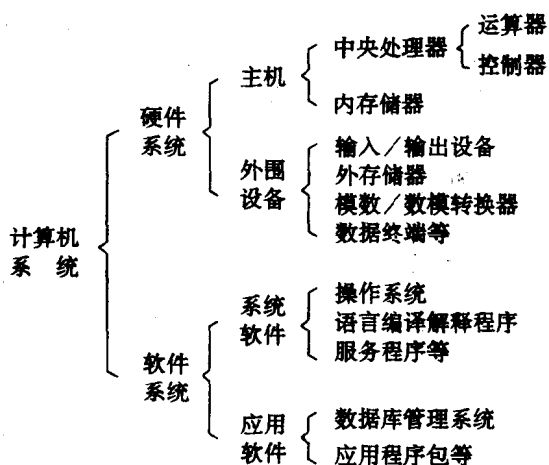


图 1-1 计算机系统组成

1.1.2 计算机硬件系统

计算机硬件系统主要由运算器、控制器、存储器、输入设备和输出设备五大部分组成。存储器一般又分为内存储器和外存储器。运算器、控制器和内存储器构成了主机部分，其中运算器和控制器通常称为中央处理器(CPU)。输入设备、输出设备和外存储器等是计算机的外围设备。它们之间的关系如图 1-2 所示。

计算机硬件的主要功能是根据计算机程序来实现数据的输入、加工和输出等一系列根本性操作。在计算机中，基本上有两种信息在流动：一种是数据信息，如原始数据、中间结果和程序，由输入设备送到内存储器中，经过加工，得到的结果或者仍然储存在内存，或者存入外存，或者经过输出设备输出。另一种是控制信息，由控制器根据程序的规定，发出控制信息以协调各部件的工作。

输入设备： 负责把外界的程序和原始数据输入到计算机中。常用的输入设备有终端键盘和鼠标器等。

输出设备： 负责把运算结果输出供用户查看。常用的输出设备有终端显示器、打印机和绘图仪等。

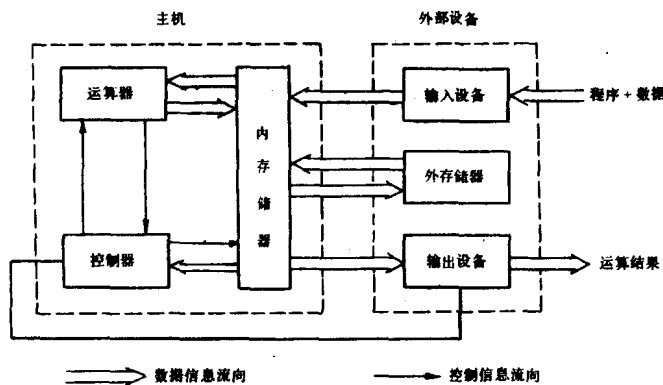


图 1-2 计算机硬件系统

运算器： 负责数据的加工处理。数据的处理,不仅指加、减、乘、除等算术运算,还包括逻辑运算。所有这些运算都是基本运算,也就是说,运算器只会作简单运算,复杂的问题在计算机中只能用简单的运算一步一步实现。然而,由于计算机的速度很快,因此它的能力是靠高速获得的。

控制器： 负责控制并协调各部件的工作,使整个计算机能自动地执行内存中的程序。控制器从内存中按一定的次序取出指令(程序由指令组

成),每取出一条指令就分析这条指令,然后根据指令的功能向各部件发出控制命令,控制它们执行这条指令中规定的任务。当各部件执行完控制器发出的命令之后,会向控制器发出执行情况的反馈信息。控制器一旦得知一条指令执行完之后,就自动去取下一条要执行的指令,然后重复上面的工作过程。

存储器： 负责存储程序和数据。程序和数据都是以二进制代码的形式储存在存储器中。存储器分为内存和外存两部分。内存容量小,但存取速度快,一般采用半导体存储器。外存容量大,但存取速度慢,常用的有磁带和磁盘。磁盘又可分为硬盘和软盘。内存以存储单元来划分。每个单元对应于一个称为单元地址(简称地址)的顺序编号。存储器的容量称为存储量。存储量一般以字节(byte)为单位。1024 个字节叫 1K 字节(1KB),1024K 个字节叫 1M 字节(1MB)。一个字节可以存放 8 个二进制位(bit),一个二进制位称为一个比特。一个存储单元叫一个字,存储单元的长度(以比特为单位)叫字长,各种计算机的字长并不划一。微型计算机的字长多是 16(即两个字节组成一个字),超级微机字长可达 32。存储器具有接受、保存和发送代码的功能。各种类型的信息,如指令、数值、文字、图形等以某种二进制代码储存在存储单元之后,一般是不会改变的。从任何存储单元读出代码后,该代码仍留在原存储单元。但当送入新代码时,就会把存储单元中原有的内容冲写掉而代之以新的代码。外存用来存放“暂时不用”的程序或数据。通常外存不和计算机内其它装置交换数据,只和内存交换数据,并且不是按单个数据进行存取,而是成批数据进行交换。

1.1.3 计算机软件系统

计算机软件包括计算机运行所需的各种程序及有关资料。没有软件的计算机系统是不能做任何有意义的工作的。计算机软件一般可分为系统软件和应用软件。

系统软件是管理、监控和维护计算机各类资源的软件。它主要包括:

操作系统;

各种程序设计语言及其解释程序和编译程序;

机器的监控管理程序、调试程序、故障检查和诊断程序,它们统称为服务程序。

应用软件是指用户利用计算机及系统软件为解决各种实际问题而开发的程序。例如数据库管理系统,文字处理软件和排版软件。

1.1.4 程序设计语言

程序设计语言是指编写计算机程序所用的语言,一般可分为机器语言、汇编语言和高级语言三类。

机器语言: 机器语言是唯一不需要翻译即可被计算机硬件系统识别的程序设计语言。它由计算机的指令系统组成,通常随计算机型号不同而不同。机器语言中的一条语句(即机器指令)实际上是一条二进制形式的指令代码。一般来说,指令是由告诉计算机进行什么操作的操作码和告诉计算机到哪儿去找操作数的地址码组成。由于二进制形式的指令不易读不易写,因此编写、调试、修改机器语言程序难度很大,很少被用户使用,但执行速度最快。

汇编语言: 汇编语言是一种面向机器的程序设计语言。在汇编语言中,用助记符代替操作码,用地址符号代替地址码。使用汇编语言编写的程序需要由一种起翻译作用的程序将其翻译成机器语言程序才可被执行。这一翻译过程称为汇编。汇编语言程序比机器语言程序易读易写易改,同时保持了机器语言程序执行速度快、占存储空间小的优点,但因汇编语言程序依赖于具体的机型,是一种低级语言程序,故难于移植。

高级语言: 同机器语言和汇编语言相比,高级语言是独立于具体的机器系统的程序设计语言。用高级语言编写的程序有较好的通用性和可移植性。而且高级语言的指令(通常称为语句)一般采用自然语汇及与自然语言相近的语法,因此使程序易读易写。自从50年代中末期发表了第一批高级语言的编译程序以来,目前世界上已有几百种高级语言,用的最多的是FORTRAN、BASIC、PASCAL、C、LISP、PROLOG等数十种语言。计算机不能直接执行用高级语言编写的程序,而必须经过“翻译”。所谓翻译,是由一种特殊的程序把高级语言程序(称为源程序)转换成机器语言程序(称为目标程序)。这种特殊的程序叫语言处理程序。高级语言的处理程序有两种:编译程序和解释程序。编译程序把整个高级语言源程序翻译成目标程序,这个过程称为编译。把目标代码连接起来即形成可执行程序。解释程序将高级语言程序逐个语句地翻译(解释)执行,其执行速度比编译的程序慢,但人机会话功能强,调试修改较容易。

1.2 计算机运算基础

1.2.1 二进制数字系统

人们习惯使用的数字表示方法是用十进制来表示的,但计算机采用的是二进制数字系统。

按进位的原则进行计数,称为进位计数制,它有两个基本特点:

逢R进一。 R是指进位计数制表示一位所需要的符号数目,称为基数。例如十进制数,逢十进一,它需要0,1,2,3,4,5,6,7,8,9十个不同的符号表示一个数位,基数为10。二进制数,逢二进一,它由0,1两个数字符号组成,基数为2。

采用位权表示法。 处在不同位置的数字所代表的值不相同,而在固定数位上表示的值是确定的,这个固定位上的值称为位权。位权与基数的关系是:各数位上的权恰巧是基数的若干次幂。因此,任一种数制表示的数都可以写成按位权展开的多项式之和。一般地,R进制数 $(D_m D_{m-1} \dots D_1 D_0 \cdot B_1 B_2 \dots B_n)_R = D_m \times R^m + D_{m-1} \times R^{m-1} + \dots + D_1 \times R^1 + D_0 \times R^0 + B_1 \times R^{-1} + B_2 \times R^{-2} + \dots + B_n \times R^{-n}$ 。

计算机采用二进制的原因主要是二进制有以下优点:

一、用二进制表示数字容易实现

二进制只有 0 和 1 两个符号,一个具有两种不同的稳定状态且能相互转换的器件,就可用来表示一位二进制数。制造具有两个不同的稳定状态的电子元件比制造有十个稳定状态的元件容易,因此在计算机内使用二进制数既简单又可靠。

二、二进制运算规则简单

二进制只有三种求和与三种求积规则。

求和: $0+0=0$; $0+1=1+0=1$; $1+1=10$

求积: $0\times 0=0$; $0\times 1=1\times 0=0$; $1\times 1=1$

十进制有 55 种求和与 55 种求积运算规则, R 进制有 $R(R+1)/2$ 种求和(求积)规则。显然,二进制运算使运算器结构大大简化,控制器也比用十进制简单多了。

三、采用二进制可进行逻辑运算

由于二进制中只有 0 和 1,它们可以用来表示两个不同的逻辑值。二进制的求积(求和)运算规则与逻辑代数中的“与”“或”运算规则类似,因此在计算机内可进行逻辑运算,使计算机具有一定的“思维能力”。

虽然计算机选择了二进制,但人们直接用二进制工作会有许多不便,因此通常使用八进制和十六进制作为二进制的缩写方法。八进制数只有 8 个数字符号:0,1,2,3,4,5,6,7。而十六进制具有 16 个数字符号:0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F。它们分别表示十六进制值 0—15。在十六进制数后加“H”,一般表示这个数是十六进制数。

表 1-1 所示内容是 0—15 在几种常用的进位计数制中的表示方法。

表 1-1 几种常用计数制的表示方法

十进制	二进制	八进制	十六进制
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

1.2.2 数制间的转换

不同数制间转换的基本方法是:将整数部分和小数部分分别进行转换,然后用小数点连

接。

一、二进制数转换为十进制数

使用按权相加法,即把二进制数展成2的多项式就可以得到相应的十进制数。

例1 求 $(10101.101)_2$ 的等值十进制数。

解 $(10101.101)_2$

$$\begin{aligned}&=1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\&=16 + 4 + 1 + 0.5 + 0.125 \\&=21.625\end{aligned}$$

二、十进制数转换为二进制数

整数的转换采用除以2(基数)取余法,即用2重复除被转换的十进制数,直至商为0。每次相除所得余数,即是相应位的二进制数。第一次除以2的余数是二进制数的最低位,最后一次所得余数为最高位。

小数部分的转换采用乘2(基数)取整法,即用2多次去乘被转换的十进制数的小数部分,每次相乘后,所得乘积的整数部分就是相应位的二进制数。第一次所得整数部分是二进制数小数部分的最高位,其次是次高位。十进制有限小数不一定都能转换成等值的二进制有限小数,所以有时要取近似值。

例2 求十进制数19的等值二进制数。

解 十进制数 基数 商 余数

$$19 \div 2 = 9 \cdots 1 = D_0 (\text{最低位})$$

$$9 \div 2 = 4 \cdots 1 = D_1$$

$$4 \div 2 = 2 \cdots 0 = D_2$$

$$2 \div 2 = 1 \cdots 0 = D_3$$

$$1 \div 2 = 0 \cdots 1 = D_4 (\text{最高位})$$

$$\text{得 } (19)_{10} = (10011)_2$$

例3 求十进制小数0.375的等值二进制小数。

解 按乘2取整进行 整数值

$$0.375 \times 2 = 0.75 \quad 0 = B_1$$

$$0.75 \times 2 = 1.5 \quad 1 = B_2$$

$$0.5 \times 2 = 1 \quad 1 = B_3$$

$$\text{得 } (0.375)_{10} = (0.011)_2$$

例4 求 $(19.375)_{10}$ 的等值二进制数。

解 根据例2和例3的结果,可得

$$(19.375)_{10} = (10011.011)_2$$

三、二进制数与八进制数、十六进制数的相互转换

把二进制数转换为八(十六)进制数时,只需将整数部分从右向左和小数部分从左向右分别按每三(四)位为一组,不是三(四)位用0补齐,用表1.1中对应的八(十六)进制数写出,即为其对应的八(十六)进制数。

反之,把八(十六)进制数转换成二进制数,只要把每位八(十六)进制数用对应的三(四)位二进制数表示即可。

例5 将 $(E05.3BA)_{16}$ 转换为二进制数。

解 $(E05.3BA)_{16} = (1110\ 0000\ 0101.0011\ 1011\ 1010)_2$

例6 将 $(172.365)_8$ 转换成二进制数。

解 $(172.365)_8 = (001\ 111\ 010.011\ 110\ 101)_2$

1.2.3 逻辑运算

计算机中的运算,无论是算术运算还是逻辑运算,实际上都是数字0、1的运算。有三种最基本的逻辑运算:逻辑与、逻辑或、逻辑非。通常把1用作逻辑真(True),0用作逻辑假(False)。

一、逻辑与

逻辑与的运算符号用“ $\cdot$ ”来表示。逻辑与的运算规则是:

$$0 \cdot 0 = 0; 0 \cdot 1 = 0; 1 \cdot 0 = 0; 1 \cdot 1 = 1$$

也就是说,只有当逻辑变量A、B都为真时,逻辑表达式 $A \cdot B$ (读作A与B)才为真。

二、逻辑或

逻辑或的运算符号用“ $+$ ”来表示,其运算规则是:

$$0 + 0 = 0; 0 + 1 = 1; 1 + 0 = 1; 1 + 1 = 1$$

也就是说,只有当逻辑变量A、B都为假时, $A + B$ (读作A或B)才为假。

三、逻辑非

逻辑非的运算符号用“ $-$ ”(上杠)来表示,其运算规则为:

$$\bar{0} = 1 \quad \bar{1} = 0$$

也就是说,如果逻辑变量A为假,则 $\bar{A}$ (读作非A)为真;如果A为真,则 $\bar{A}$ 为假。

逻辑运算满足以下基本规律:

$$\text{结合律: } A \cdot B \cdot C = (A \cdot B) \cdot C = A \cdot (B \cdot C)$$

$$A + B + C = (A + B) + C = A + (B + C)$$

$$\text{交换律: } A \cdot B = B \cdot A$$

$$A + B = B + A$$

$$\text{分配律: } A \cdot (B + C) = A \cdot B + A \cdot C$$

$$A + (B \cdot C) = (A + B) \cdot (A + C)$$

$$\text{多重否定: } \bar{\bar{A}} = A$$

四、真值表

真值表是逻辑变量与逻辑表达式取值的对照表。表1-2是两个逻辑变量取不同值时,逻辑表达式 $A \cdot B, A + B, \bar{A}$ 的值。

表 1-2

真值表

A B	A · B	A + B	$\bar{A}$
0 0	0	0	1
0 1	0	1	1
1 0	0	1	0
1 1	1	1	0

1.2.4 计算机中数的表示

数在计算机中的表示形式称为机器数。机器数分为定点数和浮点数两种。

一、定点数

定点数是小数点位置固定的数。通常,有符号的数用其最高位表示数的符号。因为小数点在计算机中难以表示,所以对小数点的位置要加以规定,使其成为隐含的小数点。如果小数点固定在符号位之后,则符号位右边的第一位即是小数的最高位,称为定点纯小数。如果小数点固定在有效数位的最后,则符号位右边所有数位组成一个整数,称为定点纯整数。

有符号定点数有三种表示方式:原码、补码和反码。

原码:原码符号位用 0 表示正号,1 表示负号,数值部分按一般二进制形式表示。

例如,若 $M_1 = +1010111$ $M_2 = -1010111$,

则 $[M_1]_{\text{原}} = 01010111$ $[M_2]_{\text{原}} = 11010111$

补码:正数的补码与其原码相同,负数的补码则先对该数的原码除符号位外各位取反,即原来为 0 的变为 1,原为 1 的变为 0,然后在末位加 1。

例如,若 $N = -1101101$,则 $[N]_{\text{补}} = 10010011$ 。

引入补码之后,加减法都可以用加法来实现,数的符号位也当作数值一样处理,且两数的补码之和等于两数和的补码,这就使得加减法运算大为简化。因此,在近代计算机中加减法多采用补码运算。

例如,若 $P_1 = +0010110$ $P_2 = -1001101$,

则 $P_1 + P_2 = -0110111$ $[P_1]_{\text{补}} = 00010110$

若 $[P_2]_{\text{补}} = 10110011$ $[P_1 + P_2]_{\text{补}} = 11001001$,

则 $[P_1]_{\text{补}} + [P_2]_{\text{补}} = 11001001$ 。

反码:正数的反码同其原码相同,负数的反码是对该数的原码除符号位外各位取反。

例如,若 $S = -1011100$,则 $[S]_{\text{反}} = 10100011$ 。

反码通常用作求补码的中间形式。

二、浮点数

浮点数是小数点位置不固定的数。通常,一个二进制数 R 可用下列形式表示:

$$R = \pm d \times 2^{\pm P}$$

式中 R 、 d 、 P 都是二进制数。 d 称为 R 的尾数,表示全部有效数字(数值小于 1), d 前面的符号称为数符。 P 称为 R 的阶码,它指明了小数点的实际位置, P 前面的符号称为阶符。阶符和阶码构成阶码部分,数符和尾数构成尾数部分。

阶符	阶码 P	数符	尾数 d
----	------	----	------

三、数的范围

机器数能表示的数的范围受到存放机器数的二进制位的数目的限制。如果用一个 16 位字来表示有符号整数,则数的范围是 $-32768 \sim +32767$ 。如果用一个(8 位)字节表示一个无符号整数,则可以表示的数的范围是 $0 \sim 255$ 。如果用四个字节(32 位)表示一个浮点数,阶码占 7 位,尾数占 23 位,则所能表示的数的范围要大的多。但不论用多少二进制位来表示一个数,机器数的表示范围总是有限的。如果数值超出这个范围,就会发生“溢出”。在数的可能表示范围内,整数一定是精确的,而小数的表示可能只是近似值,因为有限十进制小数(例如 0.9)不一定能用有限二进制小数来表示。

1.2.5 数字编码与字符编码

计算机输入输出的是数字、字母、符号以及其它信息(如图象、声音),但计算机内部只能存放二进制代码信息,因此需要对数字、字母、符号等进行特殊的编码。

一、数字编码

计算机在输入输出数字时要进行十—二进制转换和二—十进制转换。为方便起见,通常把一位十进制数写成四位二进制数,称为十进制数的二进制编码,简称二—十进制编码或 BCD(Binary-Coded Decimal)编码。常用的 BCD 编码是 8421 码(8、4、2、1 分别表示四个二进制位上的权)。

例如,(0110 1001.0011 0111)BCD=69.37

表 1-3 列出了 8421 码与十进制数之间的关系。

表 1-3 8421 码与十进制数的关系

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

二、字符编码

普通采用的字符编码是美国标准信息交换码,简称 ASCII 码。它用七位二进制数进行编码,故可以表示 128 个符号,其中包括数字(0—9),大小写英文字母等可打印字符。实际上一个字符占一个字节(8 位),其最高位用作奇偶校验。基本 ASCII 码字符表见附录二。

1.3 微型计算机的结构

1.3.1 微型计算机系统

70 年代初,出现了第一批微处理器,它把计算机的运算器、控制器制作在一片大规模集成