

21世纪高等学校计算机规划教材

大学计算机 基础

武兆辉 主编

张欣 张海霞 刘冬 陈华 马占有 麦海娟 李娜 王珊珊 编著

21st Century University
Planned Textbooks of Computer Science

 人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等学校计算机规划教材

大学计算机 基础

武兆辉 主编

张欣 张海霞 刘冬 陈华 马占有 麦海娟 李娜 王珊珊 编著

21st Century University
Planned Textbooks of Computer Science

人 民 邮 电 出 版 社

北 京

图书在版编目 (C I P) 数据

大学计算机基础 / 武兆辉主编 ; 张欣等编著. --
北京 : 人民邮电出版社, 2012.10
21世纪高等学校计算机规划教材
ISBN 978-7-115-19342-1

I. ①大… II. ①武… ②张… III. ①电子计算机—
高等学校—教材 IV. ①TP3

中国版本图书馆CIP数据核字(2012)第211062号

内 容 提 要

本书是一本讲述计算机基础知识和应用的教材,是我们多年教学实践和探索教学改革结晶。

本书根据现代社会对大学生计算机基础知识和能力的要求而编写,包括大学计算机基础部分和全国计算机等级考试二级公共基础部分。主要内容有信息表示,计算机系统及使用,计算机网络基础与 Internet 应用,办公软件 Word、Excel 及 PowerPoint 的使用,数据结构基础,程序设计基础,软件工程基础和数据库原理基础等。

本书适合大学本科非计算机专业学生以及准备参加全国计算机等级考试二级的人员使用。

21 世纪高等学校计算机规划教材

大学计算机基础

-
- ◆ 主 编 武兆辉
编 著 张 欣 张海霞 刘 冬 陈 华 马占有
麦海娟 李 娜 王珊珊
责任编辑 武恩玉
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京隆昌伟业印刷有限公司印刷
- ◆ 开本: 787×1092 1/16
印张: 12.25 2012 年 10 月第 1 版
字数: 321 千字 2012 年 10 月北京第 1 次印刷

ISBN 978-7-115-19342-1

定价: 28.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223
反盗版热线: (010)67171154

前 言

本书是参照《中国高等院校计算机基础教育课程体系 2006》和《高等学校文科类专业大学计算机教学基本要求（2008 年版）》，结合我们多年的教学实践和探索教学改革的经验编写的。目前的计算机基础教材很多都不包括全国计算机二级等级考试的内容，学生学完计算机基础，还是没法参加计算机等级考试；而本书在计算机基础教材内容的基础上，增加了全国计算机等级考试二级公共基础部分的内容。学生再学一门后续的程序设计课程，就可以参加全国计算机二级等级考试了。

本书既注重计算机基础理论、基本概念的教学，又加强了实践能力、动手能力的培养；既强调提高学生的计算机素养和信息技术素养，又重视提高学生计算机应用技术的掌握。教材中计算机应用部分的内容操作性强，力求用新颖、实用的案例来讲解，既能提高学生的学习兴趣，又能增强可操作性，便于学生尽快掌握。

本书主要内容包括信息表示、计算机系统的基本组成、计算机网络及 Internet 的应用、数据结构与算法基础、程序设计基础、软件工程基础、数据库原理基础、中文 Windows 操作系统的使用、文字处理软件 Word 的使用、表格处理软件 Excel 的使用以及演示文稿制作软件 PowerPoint 的使用等知识。

本书第 1 章与第 9 章由武兆辉编写，第 2 章由马占有编写，第 3 章由王珊珊编写，第 4 章由张欣编写，第 5 章由麦海娟编写，第 6 章由李娜编写，第 7 章由刘冬编写，第 8 章由陈华编写，第 10 章由张海霞编写。第 2 章与第 3 章由张海霞审核，第 4 章与第 5 章由张欣审核，第 7 章、第 8 章和第 10 章由刘冬、武兆辉审核，其他章节由武兆辉审核。全书由武兆辉统稿。

对于北方民族大学及教务处给予的大力支持表示感谢！

由于编者水平有限，书中难免存在一定的缺点和错误，敬请读者给予指正。

编 者

2012 年 7 月于北方民族大学

第 1 章 数据在计算机中的表示	1	3.1.1 计算机网络概述	34
1.1 进位计数制	1	3.1.2 计算机网络系统构成	37
1.1.1 计算机中的数制	1	3.1.3 计算机网络的协议	38
1.1.2 不同进制间的转换	2	3.2 局域网基础	38
1.1.3 二进制数的常用单位	4	3.2.1 局域网的特点及组成	38
1.2 数值编码	5	3.2.2 局域网的拓扑结构	39
1.2.1 整数的编码	5	3.2.3 局域网的工作模式	41
1.2.2 实数的编码	7	3.3 Internet 基础	42
1.3 字符编码	8	3.3.1 Internet 的概念与特点	42
1.3.1 ASCII 码	8	3.3.2 Internet 的发展概况	44
1.3.2 汉字编码	8	3.3.3 Internet 的通信协议	45
1.3.3 其他汉字编码	10	3.3.4 Internet 的接入方式	46
小结	11	3.3.5 Internet 提供的服务	47
习题	11	3.4 利用 Internet 进行信息浏览	49
第 2 章 计算机系统	12	3.4.1 浏览器 IE 的基本操作	49
2.1 计算机硬件系统	12	3.4.2 网页浏览技巧	55
2.2 操作系统	14	3.5 利用 Internet 进行信息检索	59
2.2.1 操作系统的功能	14	3.5.1 搜索引擎	59
2.2.2 操作系统的分类	15	3.5.2 使用 Google 搜索信息	59
2.3 计算机语言与语言处理程序	17	3.6 利用 FTP 进行文件传输	60
2.3.1 程序设计语言	17	3.6.1 FTP 概述	60
2.3.2 语言处理程序	18	3.6.2 从 FTP 网站下载文件	62
2.4 计算机的发展	19	3.6.3 从 Internet 下载文件	62
2.5 微型计算机组成	20	3.6.4 文件的压缩与解压缩	63
2.5.1 中央处理器	20	3.7 在 Internet 上收发电子邮件	66
2.5.2 主板系统	22	3.7.1 电子邮件基础知识	66
2.5.3 存储器系统	23	3.7.2 通过 Outlook Express 收发电子 邮件	67
2.5.4 总线 and 接口	27	3.8 Internet 上的其他服务	70
2.5.5 输入/输出设备	29	小结	72
小结	31	习题	73
习题	31	第 4 章 文字处理软件的功能和使用	75
第 3 章 计算机网络基础与 Internet 应用	34	4.1 Word 基本操作	75
3.1 计算机网络技术基础	34	4.2 文档排版	76

4.2.1 字符格式	78	5.6.2 使用记录单	106
4.2.2 段落格式	78	5.6.3 排序	107
4.2.3 页面格式	79	5.6.4 分类汇总	107
4.3 在 Word 中使用对象	80	5.6.5 筛选数据	109
4.3.1 表格	80	5.7 各种图表的制作	111
4.3.2 图片与剪贴画	81	5.7.1 创建图表	112
4.3.3 艺术字	82	5.7.2 管理图表	113
4.3.4 公式	82	小结	115
4.4 页面控制	83	习题	115
4.4.1 页面设置	83		
4.4.2 打印预览和打印	83	第 6 章 演示文稿的制作和使用	119
小结	84	6.1 工作界面和基本概念	119
习题	84	6.2 演示文稿的脚本设计	121
第 5 章 电子表格软件的功能和使用	87	6.2.1 创建演示文稿	121
5.1 工作界面和基本概念	87	6.2.2 应用设计模板	125
5.2 电子表格文件基本操作	88	6.3 在演示文稿中加入对象	127
5.2.1 工作簿管理	89	6.3.1 加入文字	127
5.2.2 管理工作表	89	6.3.2 添加项目符号列表	128
5.3 数据的输入和编辑	91	6.3.3 幻灯片中图片和艺术字的插入	128
5.3.1 输入数据	91	6.3.4 插入表格和图表	129
5.3.2 选取单元格	92	6.3.5 插入剪辑库中的影片和声音	130
5.3.3 输入序列	93	6.4 对象的格式设置	130
5.3.4 单元格的复制与移动	94	6.4.1 文字格式的设置	130
5.3.5 插入和删除单元格	95	6.4.2 图片和艺术字的格式设置	131
5.3.6 调整单元格的列宽和行高	95	6.4.3 表格格式设置	132
5.4 格式设置	96	6.5 增加动画效果	132
5.4.1 数字格式	96	6.5.1 动态显示文本和对象	132
5.4.2 对齐格式	96	6.5.2 应用动作路径	132
5.4.3 字体设置	97	6.6 设置幻灯片切换方式和交互 功能	133
5.4.4 设置边框	97	6.6.1 设置幻灯片切换	133
5.4.5 设置图案	98	6.6.2 交互功能	134
5.4.6 自动套用格式	98	6.7 演示文稿放映的设置	137
5.4.7 使用批注	99	6.7.1 设置演示文稿的放映 方式	137
5.4.8 打印报表	99	6.7.2 排练计时	137
5.5 公式、函数的使用	101	6.7.3 幻灯片播放快捷键	138
5.5.1 使用公式	101	6.7.4 录制旁白	138
5.5.2 使用函数	103	6.8 其他功能	139
5.6 数据处理	105	6.8.1 打印演示文稿	139
5.6.1 数据清单	105		

小结	142	第 8 章 程序设计基础	165
习题	142	8.1 程序设计方法和风格	165
第 7 章 数据结构的概念	145	8.2 结构化程序设计	166
7.1 有关概念和术语	145	8.2.1 结构化程序设计的原则	166
7.2 算法和算法分析	146	8.2.2 结构化程序的基本结构和特点	167
7.2.1 算法特性	147	8.3 面向对象的程序设计	168
7.2.2 算法描述	147	8.3.1 关于面向对象方法	168
7.2.3 算法性能分析与度量	147	8.3.2 面向对象方法的基本概念	170
7.3 线性表	148	小结	174
7.3.1 线性表的顺序存储	148	习题	174
7.3.2 线性表的链式存储和运算 实现	150	第 9 章 软件工程	175
7.3.3 顺序表和链表的比较	153	9.1 软件工程基本概念	175
7.4 栈和队列	153	9.2 软件开发方法	176
7.4.1 栈的定义及基本运算	154	9.3 软件测试	177
7.4.2 队列的定义及基本运算	154	9.4 软件维护	178
7.5 树与二叉树	154	小结	179
7.6 查找	158	习题	180
7.6.1 顺序查找	158	第 10 章 数据库原理基础	181
7.6.2 有序表的折半查找	159	10.1 数据库系统的基本概念	181
7.7 排序	159	10.2 数据模型	183
7.7.1 插入排序	159	10.3 关系代数	186
7.7.2 交换排序	161	10.4 数据库设计方法和步骤	187
7.7.3 选择排序	162	小结	188
小结	163	习题	188
习题	163	参考文献	190

第 1 章

数据在计算机中的表示

当前，计算机已经应用到了各行各业、各个领域；渗透到了我们工作、学习、生活、娱乐等各个方面，成为人们每天不可或缺的工具之一。计算机功能强大，可以处理数字、字符、汉字、图形、图像及声音等多种信息。那么，计算机如何识别多种不同类型的信息呢？这就是本章所要讲述的内容。

1.1 进位计数制

进位计数制也称进制。我们最熟悉的是十进制，而计算机中使用的是二进制。另外还有其他进制可以辅助使用。

1.1.1 计算机中的数制

日常生活中，人们经常会遇到不同进制的数。除了十进制外，还有十二进制、十六进制、二十四进制、六十进制等，例如 1 年有 12 个月，1 打等于 12 个，1 英尺等于 12 英寸，1 斤等于 16 两（旧秤），1 天有 24 小时，1 小时有 60 分钟，1 分钟有 60 秒。

1. 认识二进制

（1）十进制。在学习二进制之前，不妨先回顾一下我们熟悉的十进制。十进制有以下一些规律。

① 有 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 共 10 个基本数码，我们称为“基数”。

② 进位规则为“逢十进一”。

③ 每个数位的位值称为“权”，指的是基数 10 的某次幂。如小数点左边第一位是 10 的 0 次幂，第二位是 10 的 1 次幂，依次类推；而小数点右边第一位的权是 10 的 -1 次幂，第二位是 10 的 -2 次幂等。

例如，十进制数 128.56 可写成：

$$128.56 = 1 \times 10^2 + 2 \times 10^1 + 8 \times 10^0 + 5 \times 10^{-1} + 6 \times 10^{-2}$$

上面的算式叫做“按权展开式”，可以看到每一位表示的数值不仅取决于该位的数码本身，还取决于所在位的位值——权。

（2）二进制。与十进制类似，二进制也有如下规律。

① 只有 0, 1 两个数码。

② 进位规则为“逢二进一”。

③ 每个数位的权是 2 的某次幂。即，小数点向左各位的权依次是 2 的 0 次幂、1 次幂、2 次幂等；小数点向右各位的权依次则是 2 的 -1 次幂、-2 次幂等。

对于二进制数，同样可以按权展开，如二进制数 1100.01 可写成：

$$(1100.01)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$$

2. 二进制的特点

在计算机内部，一律采用二进制表示。这是因为，二进制有以下特点。

（1）物理上容易实现，可靠性强。电子元器件大都具有两种稳定的状态：如电压的高和低，晶体管的导通和截止，电容的充电和放电等，这两种状态分明，工作可靠，抗干扰能力强。两种状态正好表示二进制的两个数码 0 和 1。

（2）运算简单，通用性强。如二进制数乘法运算规则只有 3 种： $1 \times 0 = 0 \times 1 = 0$ ； $0 \times 0 = 0$ ； $1 \times 1 = 1$ 。而十进制的乘法运算法则有 55 种。

（3）便于实现逻辑运算。计算机中二进制数的 0、1 数码与逻辑量“真”和“假”的 0 与 1 吻合，便于表示和运算。

3. 几种常用的数制

由于二进制在表达信息时位数太长，书写麻烦又不易识别，因此在编写计算机程序时，经常将它们写成对应的八进制或十六进制，也经常采用十进制。表 1-1 列出了几种常用的计数制。

表 1-1 几种常用的数制

数制	基数	数 码	位权	进位规则
二进制	2	0、1	2^i	逢二进一
八进制	8	0、1、2、3、4、5、6、7	8^i	逢八进一
十进制	10	0、1、2、3、4、5、6、7、8、9	10^i	逢十进一
十六进制	16	0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F	16^i	逢十六进一

1.1.2 不同进制间的转换

1. 二进制数转换成十进制数

把一个二进制数转换成十进制，只需使用前面讲述的“按权展开法”，即，将各数码乘以各自的权再相加。例如，将二进制数 110101.11 转换成十进制：

$$\begin{aligned} (110101.11)_2 &= 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^2 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} \\ &= 32 + 16 + 4 + 1 + 0.5 + 0.25 \\ &= 53.75 \end{aligned}$$

即： $(110101.11)_2 = (53.75)_{10}$

2. 十进制数转换成二进制数

十进制数要转换成二进制，应先将十进制数分成整数部分和小数部分，再分别进行转换，最后加起来即可。

（1）十进制整数转换成二进制数。

转换方法：除 2 倒序取余。即对该十进制整数逐次除以 2，直到商为 0，然后逆序取每次的余数（最后得到的余数为最高位），就可得到对应的二进制数。例如，将十进制整数 125 转换成二进制，如图 1-1 所示。

即： $(125)_{10} = (1111101)_2$

（2）十进制小数转换成二进制数。

转换方法：乘 2 顺序取整。即对十进制纯小数逐次乘以 2，直到乘积的小数部分为 0，取每次

乘积的整数，就可得到对应的二进制小数。需要注意的是，每次相乘时只应乘前次乘积的小数部分。例如，将十进制小数 0.234375 转换成二进制，如图 1-2 所示。

2		125	1
2		62	0
2		31	1
2		15	1
2		7	1
2		3	1
2		1	1
		0	

图 1-1 除 2 取余法

0.234375	
×	2
0.468750	整数部分：0
×	2
0.937500	整数部分：0
×	2
1.875000	整数部分：1
×	2
1.750000	整数部分：1
×	2
1.500000	整数部分：1
×	2
1.000000	整数部分：1

图 1-2 乘 2 取整法

即： $(0.234375)_{10}=(0.001111)_2$



用上述方法把十进制数转换为二进制时，其整数部分可用有限的二进制整数表示，而小数部分则不一定能用有限位二进制小数来表示，这时，可根据精度要求进行取舍，会有一定的误差。

3. 二进制、八进制、十六进制数间的相互转换

(1) 八进制数。八进制数的基数为 8，由 0、1、2、3、4、5、6、7 八个数码组成，进位规则是“逢八进一”。由于 8 是 2 的 3 次方，因此，一位八进制数对应 3 位二进制数，相互转换很方便。

例如，将八进制数 527.34 转换为二进制为：

5 2 7. 3 4
101 010 111. 011 100

即： $(527.34)_8=(101010111.0111)_2$

二进制数转换为八进制时，以小数点为基准，向左、向右每 3 位为一组合并为八进制数（整数部分不足 3 位，高位补 0；小数部分不足 3 位，低位补 0）。

例如，将二进制数 10110101.11 转换为八进制数为：

010 110 101.110
2 6 5. 6

即： $(10110101.11)_2=(265.6)_8$

(2) 十六进制数。十六进制数的基数为 16，由 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F 十六个数码组成，进位规则是“逢十六进一”。由于 16 是 2 的 4 次方，因此，一位十六进制数对应 4 位二进制数，与二进制数相互转换也很容易。

例如，将十六进制数 DA85.9 转换为二进制为：

D A 8 5. 9

1101 1010 1000 0101. 1001

即：(DA85.9)₁₆=(1101101010000101.1001)₂

二进制数转换为十六进制时，以小数点为基准，向左、向右每 4 位为一组合并为十六进制数（整数部分不足 4 位，高位补 0；小数部分不足 4 位，低位补 0）。

例如，将二进制数 10110101.11 转换为十六进制数为：

1011 0101.1100

B 5. C

即：(10110101.11)₂=(B5.C)₁₆

表 1-2 所示为几种常用计数制的对照表。

表 1-2 几种常用进制数对照表

十 进 制 数	二 进 制 数	八 进 制 数	十六进制数
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
16	10000	20	10

1.1.3 二进制数的常用单位

在计算机中，一切数据都是以二进制数的编码来表示。为了衡量计算机中数据的量，规定了一些二进制数的常用单位，如位、字节、字等。

（1）位。位（bit）又称为比特，用来表示一个二进制数，是计算机中数据的最小单位。通常将位的英文单词 bit 简写成 b。

（2）字节。将 8 个二进制位组成一组，称作字节（byte）。它是计算机中数据处理和存储的基本单位。在书写时，常将字节英文单词 byte 简写成 B。即：

1B = 8b

常用的单位还有 KB、MB、GB、TB 等，它们与字节的关系是：

1KB = 1 024B

1MB = 1 024KB = 1 024 × 1 024B

1GB = 1 024MB = 1 024 × 1 024 × 1 024B

1TB = 1 024GB = 1 024 × 1 024 × 1 024 × 1 024B

(3) 字。字 (或者叫字长, Word) 是计算机一次能传输和处理的二进制位的长度。一个字一般由一个或几个字节组成, 它是衡量计算机性能的一个重要指标。计算机的字长越长, 其运算速度越快, 计算精度越高。计算机字长与系统有关, 目前常见的有 32 位和 64 位等。通常我们说多少位的计算机, 就是指计算机的字长是多少位。

1.2 数值编码

在计算机内部, 系统能直接处理的只是二进制数。这就意味着, 计算机所表达的数据、字符、图像、声音等信息, 都是二进制数 0、1 的某种编码。因此, 进入计算机中的各种数据都要进行编码以及编码的转换; 同样, 从计算机中输出的数据也必须进行逆向的转换。转换关系如图 1-3 所示。

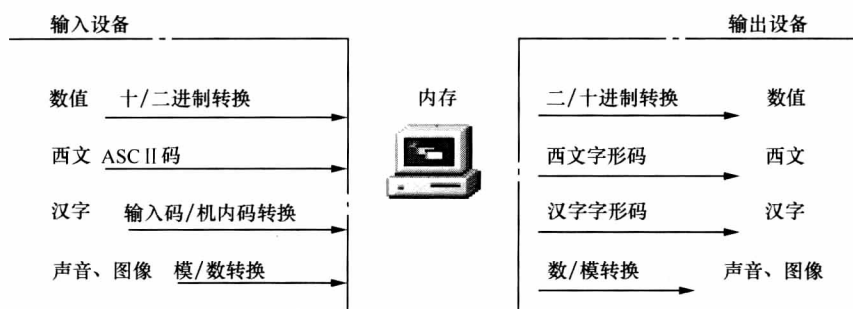


图 1-3 各类数据的转换关系

1.2.1 整数的编码

在计算机中, 因为只有 0 和 1 两种形式, 为了表示数的正 (+) 和负 (-), 就要将数的符号用 0、1 编码。通常把一个数的最高位定义为符号位, 用 0 表示正, 1 表示负, 称为数符, 其余位表示数值。

一个数在计算机内的表示形式称为机器数, 其值称为真值。例如, 一个 7 位的二进制数 -1011001, 它在计算机中表示为 11011001 (其中, 最高位为符号位, 其他位为数值位), 该机器数如图 1-4 所示。



图 1-4 机器数

机器数 -5 与机器数 +4 的和为 -1, 但在计算机中若按照上面讲的符号位同时和数值参加运算, 则运算如图 1-5 所示。

若要考虑符号位的处理, 则运算变得复杂。为了解决此类问题, 引入了多种编码表示方式, 常用的是: 原码、反码和补码, 其实质是对负数表示的不同编码。

$$\begin{array}{rcl}
 10000101 & \dots & -5 \text{ 的机器数} \\
 + 00000100 & \dots & 4 \text{ 的机器数} \\
 \hline
 10001001 & \dots & \text{运算结果为 } -9
 \end{array}$$

图 1-5 表达式 “(-5) + 4”

1. 原码

整数 X 的原码就是指 X 的机器数。即最高位为符号位, 用 0 表示正, 1 表示负, 其余位为数值位, 表示数 X 的绝对值, 通常写为 $[X]_{\text{原}}$ 。

$$[X]_{\text{原}} = \begin{cases} 0X, & X \geq 0 \\ 1|X|, & X \leq 0 \end{cases}$$

$[+7]_{\text{原}}: 00000111$ $[+0]_{\text{原}}: 00000000$ $[+127]_{\text{原}}: 01111111$

$[-7]_{\text{原}}: 10000111$ $[-0]_{\text{原}}: 10000000$ $[-127]_{\text{原}}: 11111111$

由此可见, 8 位原码表示的最大值为 2^7-1 , 即 127, 最小值为 -127。表示的数的范围为 -127~127。当采用原码表示法时, 编码简单, 与真值转换方便。但原码也存在如下问题。

① 在原码表示法中, 0 有两种形式, 即:

$[+0]_{\text{原}}: 00000000$ $[-0]_{\text{原}}: 10000000$

② 用原码做四则运算时, 符号位需要单独处理, 增加了运算规则的复杂性。当两个数做加法运算时, 如果两数符号相同, 则数值相加, 符号不变。如果两数符号不同, 则数值部分应该相减, 这时, 必须比较两个数的绝对值大小, 来决定运算结果的符号位及值。

原码运算的这些不足之处, 促使人们去寻找更好的编码方法。

2. 反码

整数 X 的反码是指: 对于正数, 反码与原码相同; 而对于负数, 其符号位为 1, 数值位为其绝对值取反, 即, 负数反码符号位为 1, 数值位为该负数的原码取反, 用 $[X]_{\text{反}}$ 表示。

$$[X]_{\text{反}} = \begin{cases} 0X, & X \geq 0 \\ 1|X|, & X < 0 \end{cases}$$

$[+7]_{\text{反}}: 00000111$ $[+0]_{\text{反}}: 00000000$ $[+127]_{\text{反}}: 01111111$

$[-7]_{\text{反}}: 11111000$ $[-0]_{\text{反}}: 11111111$ $[-127]_{\text{反}}: 10000000$

在反码表示中, 0 也有两种表示形式。

8 位反码表示的最大值、最小值和表示数的范围与原码相同, 反码运算也不方便, 一般用作求补码的中间码。

3. 补码

整数 X 的补码是指: 对于正数, 与原码、反码相同; 对于负数, 为该数的反码加 1, 用 $[X]_{\text{补}}$ 表示。

$$[X]_{\text{补}} = \begin{cases} 0X, & X \geq 0 \\ [X]_{\text{反}} + 1, & X < 0 \end{cases}$$

$[+7]_{\text{补}}: 00000111$ $[+0]_{\text{补}}: 00000000$ $[+127]_{\text{补}}: 01111111$

$[-7]_{\text{补}}: 11111001$ $[-0]_{\text{补}}: 00000000$ $[-127]_{\text{补}}: 10000001$

在补码表示中, 0 有唯一的编码。可以用多出来的一个编码 10000000 表示 -128, 既与补码的规则一致, 又扩大了数的表示范围, 不浪费一个编码。

例如 $(-5) + 4$ 的运算:

$[-5]_{\text{补}} = 1111\ 1011$

$[4]_{\text{补}} = 0000\ 0100$

$-5 + 4 = 1111\ 1111$

运算结果的符号位为 1, 表示负数。已知负数的补码, 要求其真值, 只要将数值位再求一次补码 (即取反再加 1), 就可得到其原码。该数的原码为 -1, 运算结果正确。

又如, $(-5) + (-4)$ 的运算:

$[-5]_{\text{补}} = 1111\ 1011$

$[-4]_{\text{补}} = 1111\ 1100$

$(-5) + (-4) = 1111\ 0111$

丢掉高位的 1，运算结果为-9。

利用补码可以方便地实现正、负数的加法运算，其运算规则简单。在数的有效范围内，符号位与数值一样可以参与运算，也允许产生最高位的进位，使用广泛、方便。

当运算结果超出其所能表示的范围时，会产生不正确的结果。

例如，计算 $60+68$ ：

$$60 = 0011\ 1100$$

$$68 = 0100\ 0100$$

$$60+68 = 1000\ 0000$$

两个正数相加，结果为负数，原因是结果超出了该数的有效范围。要存储、处理这类数，应采用实数表示。

1.2.2 实数的编码

计算机中对于小数的处理采用定点数和浮点数。所谓定点数，是指把小数点隐含在了某一位置。小数点位置固定在数值的最后称为定点纯整数；小数点位置固定在符号位之后、数值位之前称为定点纯小数。定点纯整数和定点纯小数分别如图 1-6 和图 1-7 所示。

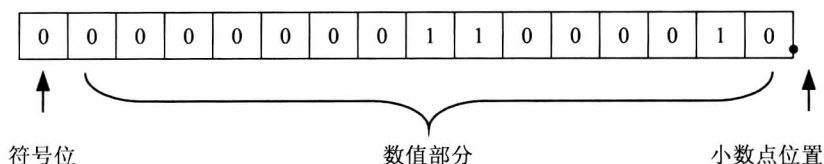


图 1-6 定点纯整数

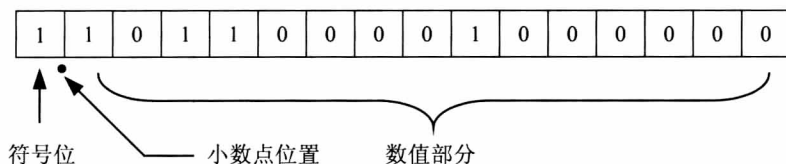


图 1-7 定点纯小数

但在实际存储数据和计算时，定点数是不方便的，范围也有限。因此通常采用浮点数，也就是指数形式，比如，一个实数可以用指数形式表示为

$$N = \pm d \times 10^{\pm p}$$

其中， d 是尾数，前面的“ $\pm$ ”表示数符； p 是阶码，前面的“ $\pm$ ”表示阶符。

为便于计算机中小数点的表示，规定将浮点数写成规格化的形式，即尾数的绝对值小于 1，且大于等于 0.1，从而唯一地规定了小数点的位置。例如：

$$3.14159 = 0.314159 \times 10^1$$

同样，任意二进制规格化浮点数的表示形式为

$$N = \pm d \times 2^{\pm p}$$

它在计算机内的存储形式如图 1-8 所示。

阶符	阶码	数符	尾数
----	----	----	----

图 1-8 浮点数的存储形式

阶码为整数，尾数表示数的有效部分，是纯小数。在浮点数的表示中，数符和阶符都各占一位，阶码的位数表示了数的范围，尾数的位数表示了数的精度。阶码、尾数的位数与机器及字长有关。

例如

$110.011 = 0.110011 \times 2^{11}$

1.3 字符编码

字符是计算机中使用最多的信息形式之一，也是人与计算机通信的重要媒介。将字符变为指定的二进制符号称为编码。在计算机内部，要为每个字符指定一个确定的编码，作为识别与使用这些字符的依据。

一个编码就是一串二进制位“0”和“1”的组合。对一个由 128 个符号构成的符号集进行编码，需要用 7 位二进制数。

1.3.1 ASCII 码

目前计算机中使用最广泛的符号编码是 ASCII 码，即美国标准信息交换码（American Standard Code for Information Interchange）。ASCII 码包括 32 个通用控制字符、10 个数字、52 个英文大小写字母和 34 个专用符号，共 128 个元素，故需要用 7 位二进制数进行编码，以区分每个字符。通常使用一个字节（即 8 个二进制位）表示一个 ASCII 码字符，规定其最高位为 0。ASCII 码见表 1-3：

表 1-3 7 位 ASCII 码编码表

b₇b₆b₅ b₄b₃b₂b₁	000	001	010	011	100	101	110	111
0000	32 个 控制字符		空格	0	@	P	‘	p
0001			!	1	A	Q	a	q
0010			“	2	B	R	b	r
0011			#	3	C	S	c	s
0100			\$	4	D	T	d	t
0101			%	5	E	U	e	u
0110			&	6	F	V	f	v
0111			`	7	G	W	g	w
1000			(	8	H	X	h	x
1001			)	9	I	Y	i	y
1010			*	:	J	Z	j	z
1011			+	;	K	[	k	{
1100			,	<	L	\	l	
1101			-	=	M	]	m	}
1110			.	>	N	^	n	~
1111			/	?	O	_	o	DEL

1.3.2 汉字编码

在我国推广应用计算机，必须使其具有汉字信息处理能力。对于这样的计算机系统，除了配备必要的汉字设备和接口外，还应该装配有支持汉字信息输入、输出和处理的操作系统。汉字信息的输入、输出及其处理远比西文困难，原因是汉字为语素文字，不同于英语的音素文字。所以，

对于汉字的处理，从输入、存储到输出都要有相应的编码。

1. 汉字交换码

汉字也是一种字符。常用的汉字就有几千个，显然无法用一个字节的编码来区分。1981 年，我国公布了《通用汉字字符集（基本集）及其交换码标准》GB 2312-80，共收集了 7 445 个图形字符，其中汉字字符 6 763 个，并分为两级，即常用的一级汉字 3 755 个（按汉语拼音排序）和次常用汉字 3 008 个（按偏旁部首排序），其他图形符号 682 个。

GB 2312-80 编码简称国标码，它规定每个图形字符由两个 7 位二进制编码表示，即每个编码需要占用两个字节，每个字节用低 7 位编码，最高位置 0。例如汉字“啊”的国标码为 3021H，即 00110000 00100001。

2. 汉字输入码

在计算机系统上处理汉字时，首先遇到的问题是输入汉字。若用普通的西文键盘输入汉字，就需要对汉字按某种输入方案进行编码，即对每个需要输入的汉字按照一定的规则用键盘上的若干个字母、数字或其他符号来代表。这种代表符号就称为汉字的输入码，通常也称作汉字的外码。

衡量一个输入编码的好坏通常遵循这么几条原则：首先，编码要短，重码少，这样才能提高输入速度，提高效率；另外，还要好学、易记，便于推广使用。

多年来，人们在计算机输入编码方案的研究上作了大量的工作，提出的方案不下几百种，归纳起来，基本上可分为音码、形码、音形码以及数字编码等几大类。

（1）音码。音码是以汉语读音为基础的编码方法，如智能 ABC、全拼、双拼等。其优点是不需要专门学习，只要有汉语拼音的基础就能使用该方法输入汉字；缺点是由于汉字同音字较多，重码率较高，因此，还必须进行同音字选择，以致输入速度较慢。

（2）形码。形码是根据汉字形状确定的编码。把汉字拆分成部首，用字母或数字进行编码，如五笔字型、四角号码等输入法。它的优点是重码率低，适合专业人员使用；缺点是需要一定时间的学习和训练。

（3）音形码。音形码是根据汉字的读音和字形进行编码。即取音码实施简单、易于接受和形码形象、直观的优点，从而得到较好的输入效果，如自然码等。

（4）数字编码。数字编码是用一串数字来编码一个汉字。最常用的是国标区位码，简称区位码，它实际上是国标码的一种简单变形。把 GB 2312-80 全部字符集分为 94 区，其中 1~15 区是字母、数字和图形符号区，16~55 区是一级汉字区，56~87 区是二级汉字和偏旁部首区。每个区又分为 94 位，编号也是从 01~94。这样，每一个字符便具有一个区号和一个位号。将区号置前，位号置后，组合在一起就成为区位码。

国标码与区位码是一一对应的。区位码通常用十进制数表示。将汉字的区号和位号分别转换成十六进制后再分别加 20H，就得到了相应的国标码。国标码增加 20H 是为了避开 ASCII 码中的控制符。

例如，汉字“啊”在第 16 区第 01 位。它的区位码为 1601（十进制输入就按此码），转换成十六进制为 1001H，分别加上 20H 后便是国标码，即 3021H。

为了提高输入速度，输入方法智能化成为目前研究的主要课题。未来的智能化方向基于模式识别的语音识别输入、手写输入和扫描输入。汉字语音输入法是操作者只要对着计算机口述，计算机就能识别并记录下来，而且能根据不同人的口音特点自动识别。这体现了计算机人性化发展的趋势。

不论操作者使用哪种输入法，汉字在计算机内部都是以机内码的形式表示的。

3. 汉字机内码

汉字机内码是汉字在计算机内部存储、处理和传输用的信息代码。为了实现中、英文兼容，

采用字节的最高位来区分某个码值是代表汉字还是 ASCII 码字符。通常将国标码两个字节的最高位置“1”作为汉字机内码。例如，汉字“啊”的国标码为 3021H，两个字节的最高位均为 0，把它们改为 1 后变为 B0A1H，这就是“啊”的机内码。要求它与 ASCII 码兼容但又不能相同，以便实现汉字和西文的并存。

4. 汉字字形码

汉字字形码是用在输出时产生汉字的字形，即汉字的输出是指汉字字形的输出。输出方式是显示和打印两种。实际上，每一个汉字都是一个特定的图形，可用一个点阵来表示。点阵的每个点位只有两种状态：有点或无点。若用二进制代码来表示，即为该位取值为 1 表示有点，取值为 0 表示无点。

例如，如果用 16×16 点阵来表示一个汉字，则该汉字图形由 16 行 16 列共 256 个点构成。这 256 个点需要 256 个二进制位来描述，也就是需要 32 个字节来存储其点阵信息。这样的二进制代码串称为汉字的“字模”。所有汉字和各种符号的点阵信息就组成汉字的“字模库”（简称字库）。这类点阵字库最大的缺点是不能放大，一旦放大后就会发现文字边缘出现锯齿状。

另外，还有一种字库叫矢量字库。矢量字库是用矢量的方法表述的汉字信息，比如一个笔画的起始、终止坐标，以及方向、弧度等。在显示、打印这一类字库时，要经过一系列的数学运算才能输出结果。但是这一类字库可以实现各种文字字体大小的无级缩放。目前，Windows 系统中普遍采用的矢量字库是 TrueType 字形技术。

5. 各种编码之间的关系

汉字通过输入码，并借助输入设备输入到计算机内，再由汉字系统的输入管理模块进行查表或计算，将输入码（外码）转换成机内码存入计算机存储器中。当存储在计算机内的汉字需要在屏幕上显示或在打印机上输出时，要借助汉字机内码在汉字库中找出相应的字形码，并将其输出。这种代码转换过程如图 1-9 所示。

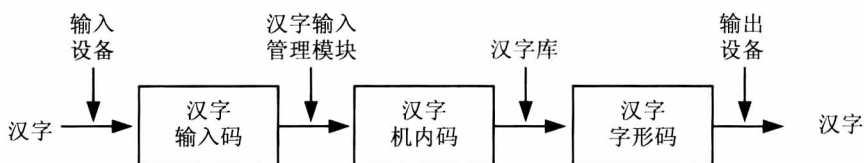


图 1-9 汉字在计算机中的处理过程

1.3.3 其他汉字编码

GB 2312 国标码只能表示和处理 6 763 个汉字。为了统一地表示世界各国、各地区的文字，便于全球范围的信息交流，各种组织公布了不同的汉字编码，最常见的有以下几种。

① Unicode 码是另一种国际编码标准，它为每种语言中的每个字符（包括西文字符）设定了唯一的二进制编码，便于统一地表示世界上的主要文字，以满足跨语言进行文字转换和处理。目前，Windows 的内核已经支持 Unicode 字符集。

② GBK 码（扩展汉字内码规范）是我国针对 GB 2312 国标码的补充，它对多达 21 003 个简、繁汉字进行了编码。该编码标准向下与 GB 2312 编码兼容，向上支持国际标准，起到了承上启下的作用。Windows 3.1 简体中文操作系统使用的就是 GBK 编码。

③ GB 18030 编码是取代 GB K1.0 的正式国家标准。该标准收集了 27 484 个汉字，同时还收录了藏文、蒙文、维吾尔文等少数民族文字，采用单字节、双字节和四字节编码。