全国计算机等级考试应试培训与指导教程

新大纲

二级适用

全国计算机等级考试二级基础知识应试培训教程

应试培训的首选
轻松过关的导师

- 努力贯彻新的考试大纲，按照新的考试大纲来组织内容
- 兼顾课堂教学和考生考前系统自学和复习的需要
- 这是一个集教师教学、学生自学、考前系统复习为一体的新思维教材
- 本教材特别适合以自学为主的初学者的学习需求

北京工业大学出版社

全国计算机等级考试应试培训与指导教程

二 级 适 用

全国计算机等级考试二级基础知识
应 试 培 训 教 程

本书编写组

北京工业大学出版社

内容提要

本书是《全国计算机等级考试应试培训与指导教程》系列丛书的一册，该书的主要内容是二级考试所涉及到的基础知识，它也是新的等级考试大纲颁布后增加的一门新课程。本书主要内容包括：数制转换的基础知识、计算机系统的组成与应用、DOS 操作系统与 Windows 操作系统的基本操作、计算机安全的基础知识及常用的杀病毒软件的使用、多媒体与网络基础知识。本书既适合有关学校课堂教学使用，也可作为广大考生参加等级考试的学习辅导书。

图书在版编目（CIP）数据

全国计算机等级考试二级基础知识应试培训教程/《全国计算机等级考试二级基础知识应试培训教程》编写组编. 北京：北京工业大学出版社，1998. 12

全国计算机等级考试应试培训与指导教程

ISBN 7-5639-0747-5

I. 全… II. 全… III. 电子计算机-水平考试-教材 IV. TP3

中国版本图书馆 CIP 数据核字（98）第 33734 号

书　　名	**全国计算机等级考试二级基础知识应试培训教程**
编 著 者	本书编写组
责任编辑	廖晨钟
出 版 者	北京工业大学出版社出版（北京市朝阳区平乐园 100 号　100022）
发 行 者	北京工业大学出版社发行部
印　　刷	徐水宏远印刷厂
开　　本	787 mm×1092 mm　1/16　9.25 印张　220 千字
书　　号	ISBN 7-5639-0747-5/T・104
版　　次	1999 年 1 月第 1 版　1999 年 1 月第 1 次印刷
印　　数	0001～5000
定　　价	**13.50 元**

编写说明

计算机等级考试始于1994年。原国家教育委员会于1994年1月颁布了《全国计算机等级考试考试大纲》,《大纲》的颁布带来了全社会学习计算机，使用计算机的高潮。时至今日，计算机应用技能已经成为下个世纪人们的基本技能之一。尤其对于当代大学生，计算机知识已经成为他们知识结构的重要组成部分。目前普遍开展的计算机等级考试正有效地推动这一目标的实现，与此有关的教材和参考资料的需求与日俱增。

然而，计算机科学在不断地发展，计算机等级考试也以更加猛烈之势席卷全国，各系统、行业、组织的等级考试已有10余种之多，如今，奔腾II计算机已成为主流，多媒体、Windows操作系统、计算机网络等原来的考试大纲中没有的内容已成为学习计算机的基本要求，汇编语言的应用更加普遍，数据库技术的发展也导致了FoxBASE与FoxPro成为基本要求之一。计算机硬件和计算机软件的理论也成为计算机应用人员的必学科目。许多教育工作者都认为：跟上时代前进的步伐，保持计算机等级考试的权威性、全面性、时效性，是刻不容缓的。

我们很高兴地看到：教育部已于1998年9月颁布了新的计算机等级考试大纲，为此，本书的主编约请国内部分高等学校从事计算机等级考试教学第一线工作的教师和一些对计算机普及教育有经验的同仁，根据新的大纲编写了《全国计算机等级考试应试培训与指导教程》。本丛书有让人耳目一新的感觉，它浅显易懂、循序渐进、深入浅出。在编写的过程中我们主要注意了以下几点：

一、新大纲与新思维的结合

在努力贯彻新的考试大纲，按照新的考试大纲来组织内容的同时，编者兼顾课堂教学和考生考前系统自学或复习的需要，在讲解基本知识的同时，注意分析难点，着力解决易混淆的概念，纠正错误的观点——这是一个集教师教学、学生自学、考前系统复习为一体的新思维教材。

二、应试与实际应用相结合

考试的目的是考核应试者对知识的掌握程度与灵活应用所学知识的能力，本丛书的教材均分为应试培训教材和考试辅导书两部分，配套使用。应试培训教材特别适合初学者又是自学为主的读者的学习要求。全书在培养读者实际上机操作能力方面的指导意义较为突出。考试辅导书则直接针对考试的知识点、题型进行练习与指导，可在临考前阅读，用来检验复习的效果。

三、本书简介

本书是《全国计算机等级考试应试培训与指导教程》系列丛书的一册，该书的主要内容是二级考试所涉及到的基础知识，它也是新的等级考试大纲颁布后增加的一门新课程。本书主要内容包括：数制转换的基础知识、计算机系统的组成与应用、DOS操作系统与Windows操作系统的基本操作、计算机安全的基础知识及常用的杀病毒软件的使用、多媒体与网络基础知识。本书既适合有关学校课堂教学使用，也可作为广大考生参加等级考试的学习辅导书。

虽然编委会做了大量细致的工作，但肯定还有不少谬误之处，欢迎广大读者多提意见，以利再版更正。

编者

1998年10月

目　　录

第 4 章 计算机系统的安全

第 5 章 网络与多媒体技术

第 6 章 Windows 操作系统

第 1 章　数制与编码

二级基础知识是此次计算机等级考试新设置的课程，主要是将程序设计中要用到的基础知识集中起来，以方便学习。其中的数制与编码是二级考试中最基础的预备知识，也是最难的一部分。本章先讲解数制的基本概念，然后讲解二进制数的基本知识，接下来介绍了其他的几种数制（十六进制、八进制）以及它们之间的相互转化的方法，之后介绍了二进制数的算术运算与逻辑运算的规则，最后讲述了字符与汉字的编码。

1.1　计算机的数制

计算机的基本功能之一就是进行计算。计算机由数量巨大的电子元器件与集成电路组成，那么，在这些设备中如何表示数字呢？这就涉及到二进制，它是计算机的数学基础。

1.1.1　什么是进位计数制

数制有非进位计数制和进位计数制两种。

1. 非进位计数制

非进位计数制的特点是表示数值大小的数码与它在数中的位置无关。典型的非进位计数制是罗马数字。例如，罗马数字中，Ⅰ总是代表 1，Ⅱ总是代表 2，Ⅳ总是代表 4，Ⅴ总是代表 5，X 总是代表 10，C 总是代表 100，等等。

非进位计数制表示数据不便、运算困难，现已不常用。

2. 进位计数制

进位计数制的特点是表示数值大小的数码与它在数中所处的位置有关。例如，十进制数 123.45，数码 1 处于百位上，它代表 $1 \times 10^2 = 100$，即 1 所处的位置具有 10^2 权；2 处于十位上，它代表 $2 \times 10^1 = 20$，即 2 所处的位置具有 10^1 权；其余类推，3 代表 $3 \times 10^0 = 3$，而 4 处于小数点后第一位，代表 $4 \times 10^{-1} = 0.4$，最低位 5 处于小数点后第二位，代表 $5 \times 10^{-2} = 0.05$。

十进制运算中，凡是超过 10 就向高位进一位，相邻两位间是十倍的关系，10 称为进位"基数"。可以想象：若是二进制，则进位基数应该是 2，八进制进位基数为 8，十六进制则进位基数应该是 16。

十进制数共有 0～9 十个数码，十进制数就是由这十个数码及其他一些符号（小数点、正

负号)组成。相应地,二进制数的数码为:0与1;八进制数有八个数码:0～7;十六进制数有16个数码:0～15(10至15分别由A～F表示)。

1.1.2 十进制数与二进制数

人们已经习惯使用十进制数0～9。逢十进一,借一当十,这完全是现在人们的习惯。其实,古埃及人与古巴比伦人就曾经使用过六十进制与十二进制。那么为什么在计算机中却偏偏采用古怪的二进制呢?

这是因为电子元器件最容易实现的是电路的通断、电位的高低、电极的正负,而在逻辑学中也常常用到二值逻辑,这都是因为两状态的系统具有稳定性(非此及彼),以及抗干扰性等特性。为了保证在计算机中进行数据传送,运行中不产生差错和减少计算机硬件的成本,都必须采用二进制。

二进制数只有"0"和"1"两个数码,而且由低位向高位进位时逢二进一。像101、110、110.011等都是二进制数,但是以上三个数也可以认为是十进制数,为了表示它们的区别,可以给这些数字加上括号和下标,标明是几进制的数,例如:

$(101)_2$、$(110)_2$表示二进制的数;$(110)_{10}$、$(101)_{10}$表示十进制的数。

一个十进制数525,在十进制中说它是5个百、2个十、5个一的和,也就是:

$$525 = 500 + 20 + 5 = 5 \times 100 + 2 \times 10 + 5 \times 1 = 5 \times 10^2 + 2 \times 10^1 + 5 \times 10^0$$

又如:

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

所以任意的一个十进制数都可以表示成:

$$N = d_m \times 10^m + d_{m-1} \times 10^{m-1} + \cdots + d_0 \times 10^0 + d_{-1} \times 10^{-1}$$

$$+ d_{-2} \times 10^{-2} + \cdots + d_{-n} \times 10^{-n} = \sum_{i=-n}^{m} d_i 10^i \qquad (m,n \geqslant 0) \qquad (1\text{-}1)$$

上式中,$\sum$是求和符号;d_i表示各个位上的数字;m表示10的次幂。

对于第一个例子的十进制数525:

$$n = 0, m = 2, d_2 = 5, d_1 = 2, d_0 = 5;$$

对于第二个例子的十进制数123.45:

$$n = 2, m = 2, d_2 = 1, d_1 = 2, d_0 = 3, d_{-1} = 4, d_{-2} = 5。$$

这里把10叫做权,把式(1-1)叫做十进制数的按权展开式。基数实际上表明了每一位上可取的数字的个数,如十进制每位上可以取0,1,2,…,9十个数字中的一个;二进制每一位上可以取0、1两个数字中的一个。于是,可得到一个结论:对于任意r进制数,可能出现的数字是0,1,2,…,$r-1$,共r个。

把式(1-1)中的10用r来代替:

$$N = d_m \times r^m + d_{m-1} \times r^{m-1} + \cdots + d_0 \times r^0 + d_{-1} \times r^{-1}$$

$$+ d_{-2} \times r^{-2} + \cdots + d_{-n} \times r^{-n} = \sum_{i=-n}^{m} d_i r^i \qquad (m \geqslant 0, n \geqslant 0) \qquad (1\text{-}2)$$

式(1-2)是任意进制的按权展开式。式中r若等于2,那么每一位上可取的数字就只有0和1,这就是计算机科学中广泛使用的二进制数。对于二进制数,可以把式(1-2)写成:

$$N_2 = b_m \times 2^m + b_{m-1} \times 2^{m-1} + \cdots + b_0 \times 2^0 + b_{-1} \times 2^{-1}$$

$$+ b_{-2} \times b^{-2} + \cdots + b_{-n} \times b^{-n} = \sum_{i=-n}^{m} b_i 2^i \qquad (m, n \geqslant 0) \qquad (1\text{-}3)$$

那么,上面提到的几个二进制数可以表示成:

$$(101)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$(110)_2 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

$$(110.011)_2 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3}$$

事实上,每一个十进制数都能找到相对应的二进制数,一些简单数字的二进制和十进制对照见表 1-1 所示。

表 1-1　十进制与二进制对照

十进制	10	9	8	7	6	5	4	3	2	1	0	0.5	0.25	0.125	0.0625
二进制	1010	1001	1000	111	110	101	100	11	10	1	0	0.1	0.01	0.001	0.0001

1.1.3　八进制数与十六进制数

二进制数的缺点是书写较长,不便于阅读,为此人们常用八进制数与十六进制数来表示二进制数。

对于式(1-2),取 r=8,就得到八进制数的展开形式。八进制数有 0,1,2,3,…,7 八个数字,运算规则是"逢八进一,借一当八"。

因为八与十六都是二的整数倍,所以它们在计算机科学中也有广泛的应用。八进制与十六进制书写容易、易读、易记,这是通常一些二进制代码都用八进制和十六进制来表示的原因。

对于十六进制数,按照式(1-2),取 r=16,就得到十六进制数的展开形式。但是十六进制数要有十六个数字,而常用的阿拉伯数字只有 0~9 十个数字,另外的几个数字怎么表示呢?我们采用 A~F 来表示其余的 6 个数字。

表 1-2 给出了十进制数、二进制数、八进制数和十六进制数的对照情况。

表 1-2　十进制数、二进制数、八进制数和十六进制数的对照

十进制数	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
二进制数	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
八进制数	000	001	002	003	004	005	006	007	010	011	012	013	014	015	016	017
十六进制数	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	000A	000B	000C	000D	000E	000F

八进制数可用括号加上下标 8 来表示,如:$(123)_8$、$(376)_8$ 等,以便与十进制数区别开。

十六进制可以用相同的方法来表示,如:$(3FD)_{16}$、$(068E)_{16}$等。

由于十进制数的英文是"Decimal",所以有的书上也在十进制数字后加上英文"d"或"D"来表示,如:$126=(126)_{10}=126D=126d$。

二进制数的英文是"Binary",所以在二进制数后加上"B"或"b"来表示,如:$(11000)_2=11000B=11000b$。同样,十六进制数可以在数字后加上"H"或"h"来表示,八进制数可以在数字后加上"O"或"o"来表示,如:$(3FD)_{16}=3FDH=3FDh$,$(321)_8=321O=321o$。

1.2 不同数制间的转换

1.2.1 二进制数与十进制数的转换

1. 二进制数转换为十进制数

由公式(1-3)不难得出二进制数转换为十进制数的规则：

【规则 1】先将二进制数按权展开成式(1-3)的形式，然后再把式(1-3)的各项相加，即得二进制数的十进制表示形式。

这个规则显而易见，在此不做证明。例：

$$(101.11)_2 = 1\times 2^2 + 0\times 2^1 + 1\times 2^0 + 1\times 2^{-1} + 1\times 2^{-2} = (5.75)_{10}$$

$$\begin{aligned}(101.11101)_2 &= 1\times 2^2 + 0\times 2^1 + 1\times 2^0 + 1\times 2^{-1} \\ &\quad + 1\times 2^{-2} + 1\times 2^{-3} + 0\times 2^{-4} + 1\times 2^{-5} \\ &= (5.90625)_{10}\end{aligned}$$

2. 十进制数转换为二进制数

十进制数转换为二进制数要分整数部分和小数部分来转换。

(1) 整数部分的转换。

由

$$S = K_n 2^n + K_{n-1}2^{n-1} + \cdots K_0 2^0$$

得

$$\frac{S}{2} = (K_n 2^{n-1} + K_{n-1}2^{n-2} + \cdots + K_1 2^0) + \frac{K_0}{2}$$

显然，括号内为商，K_0 是余数，$K_0 = 0$ 或 $K_0 = 1$。

继续以商为被除数，令：

$$S_1 = K_n 2^{n-1} + K_{n-1}2^{n-2} + \cdots + K_1 2^0$$

$$\frac{S_1}{2} = (K_n 2^{n-2} + \cdots + K_2 2^0) + \frac{K_1}{2}$$

这样进行 n 次后

$$\frac{S_n}{2} = \frac{K_n}{2}$$

就得到了一系列的数字：

$$K_0, K_1, K_2, \cdots, K_n$$

$$K_i = 0 \text{ 或 } K_i = 1, \qquad (i = 0, 1, \cdots, n);$$

将这些数字反序排列就能得到：

$$K_n K_{n-1} \cdots K_0$$

这便是所要求的二进制数。

于是我们得到将十进制数转换为二进制数的规则：

【规则 2-1】十进制整数转化为二进制数时，该十进制数除以 2，并记录余数，然后继续

用所得的商数除以 2，并记录余数，如此反复下去一直到商数为零，将所得余数反序排列，就得到该十进制数的二进制表示形式。这种转换的方法叫做“除基倒取余法”。

例如：

$(326)_{10}=(101100110)_2$

$326\div2=163$ ……0

$163\div2=81$ ……1

$81\div2=40$ ……1

$40\div2=20$ ……0

$20\div2=10$ ……0

$10\div2=5$ ……1

$5\div2=2$ ……1

$2\div2=1$ ……0

$1\div2=0$ ……1

又如：

$(31)_{10}=(11111)_2$

$31\div2=15$ ……1

$15\div2=7$ ……1

$7\div2=3$ ……1

$3\div2=1$ ……1

$1\div2=0$ ……1

（2）小数部分的转换。

那么，十进制的纯小数应如何转换为二进制数的表示形式呢？设：

$$S=K_{-1}2^{-1}+K_{-2}2^{-2}+\cdots+K_{-m}2^{-m}$$

于是：

$$2S_0=K_{-1}+(K_{-2}2^{-1}+\cdots+K_{-m}2^{-m+1})$$

$$K_{-1}=0 \quad 或 \quad K_{-1}=1;$$

令括号中的 $K_{-2}2^{-1}+K_{-3}2^{-2}+\cdots+K_{-m}2^{-m+1}=S_1$，得到：

$$2S_1=K_{-2}+(K_{-3}2^{-1}+K_{-4}2^{-2}+\cdots+K_{-m}2^{-m+2});$$

反复 m 次以后：

$$2S_m=K_{-m}$$

于是得到一组数字：

$$K_{-1},K_{-2},\cdots,K_{-m}$$

$$K_{-i}=0 \quad 或 \quad K_{-i}=1,(i=0,1,\cdots,m);$$

由上面的推导可以得到将十进制的纯小数转化为二进制的小数的规则：

【规则 2-2】十进制小数转换为二进制小数时，将十进制小数乘以 2，把积的整数部分记录下来，再将积的小数部分继续乘以 2，如此下去，直到小数部分为零或二进制小数部分达到精度要求。这种方法叫做“乘基取整法”。

例如：

$(0.8125)_{10}\rightarrow(0.1101)_2$

$0.8125\times2=1.625$ …………1

$0.625\times2=1.25$ …………1

$0.25\times2=0.5$ …………0

$0.5\times2=1.0$ …………1

又如：

$(0.6)_{10}\rightarrow(0.1001\cdots\cdots)_2$

$0.6\times2=1.2$ ………1

$0.2\times2=0.4$ ………0

$0.4\times2=0.8$ ………0

$0.8\times2=1.6$ ………1

$0.6\times2=1.2$ ………1

有些十进制小数转换为二进制小数时，可能无法用有限长的位数表示，这时往往按照要求精确到小数点后若干位，具体精确的位数应由实际需要或机器的字长决定。

(3) 整体的转换。

将整数与小数两部分的转换合起来,即可得到整体的转换值。如将$(326.8125)_{10}$转换为二进制数,则:

$(326)_{10}=(101100110)_2$

$(0.8125)_{10}=(0.1101)_2$

故

$$(326.8125)_{10}=(326)_{10}+(0.8125)_{10}=(101100110)_2+(0.1101)_2$$
$$=(101100110.1101)_2$$

1.2.2 其他进制的数制转换

1. 其他进制的数转换为十进制数

与二进制数转换为十进制数的方法一样,八进制、十六进制的数都可以按照权展开的方法来转换为十进制数。

例如:

$$(2B30)_{16}=2\times16^3+11\times16^2+3\times16^1+0\times16^0=(11056)_{10}$$
$$(24)_{16}=2\times16^1+4\times16^0=(36)_{10}$$
$$(81)_8=8\times8^1+1\times8^0=(65)_{10}$$
$$(134)_8=1\times8^2+3\times8^1+4\times8^0=(92)_{10}$$

2. 十进制数转换为其他进制的数

与十进制数转换为二进制数相类似,将十进制数转换为十六进制数或八进制数时,也应分成整数部分和小数部分来实现,对于整数部分采用**"除基倒取余"**的方法,而对小数部分采用**"乘基取整"**的方法;只不过这里的基不是2,而是8或16。

例如:

$(266)_{10}=(10A)_{16}$

266÷16=16 ……A

16÷16=1 ……0

1÷16=0 ……1

又如:

$(72)_{10}=(110)_8$

72÷8=9 ……0

9÷8=1 ……1

1÷8=0 ……1

由此可得将十进制数转换为任意r进制数的方法是**"除基倒取余法"**。

3. 二进制数、十六进制数、八进制数之间的转换

下面详细讲述二进制数、十六进制数、八进制数之间的转换。

前面已述及,由于八和十六都是二的整数倍,这就使得二进制数与八进制数、十六进制数之间的转换相对要容易得多。

显然,一位十六进制数需要用四位二进制数来表示,而一位八进制数要用三位二进制数来表示。

例如:十六进制的A用二进制数表示是1010;$(7)_8$用二进制数来表示是111。

那么,我们就可以用很简单的方法来实现二进制数、八进制数、十六进制数之间的转换。

【规则3】将二进制数转化为十六进制数可以将该二进制数从低位起，每四位为一组，最高一组不足四位的前面用零补齐，每组分别对应一个十六进制数字，将这些数字由低位向高位排列就得到该数的十六进制表示形式。

【规则4】将二进制数转换为八进制数可以将该二进制数从低位算起，每三位为一组，最高一组不足三位的，前面用零补齐，每组分别对应一个八进制数，将这些数字由低位向高位排列就得到该数的八进制表示形式。

例如：

$(1110101101)_2=(3AD)_{16}$

0011，1010，1101

↓ ↓ ↓

3 A D

$(100000001000)_2=(808)_{16}$

1000，0000，1000

↓ ↓ ↓

8 0 8

$(1100101101)_2=(1455)_8$

001，100，101，101

↓ ↓ ↓ ↓

1 4 5 5

$(100000001000)_2=(4010)_8$

100，000，001，000

↓ ↓ ↓ ↓

4 0 1 0

相反地，要把一个十六进制数或八进制数转换为二进制数，可以把该十六进制数或八进制数的每一位分别用四位（或三位）二进制数来表示，不足四位（或三位）时，前面应补零凑满位。

【规则5】将十六进制数转化为二进制数时，每个十六进制数与四位二进制数相对应，若不足四位时应在前面补零，这样就得到该十六进制数的二进制表示。

【规则6】将八进制数转化为二进制数时，每一个八进制数与三位二进制数相对应，若不足三位应在前面补零，这样就得到该八进制数的二进制表示。

例如：

$(ABC)_{16}=(101011001101)_2$

A B C

↓ ↓ ↓

1010 1100 1101

$(87F)_{16}=(100001111111)_2$

8 7 F

↓ ↓ ↓

1000 0111 1111

$(321)_8=(011010001)_2$

3 2 1

↓ ↓ ↓

011 010 001

$(567)_8=(101110111)_2$

5 6 7

↓ ↓ ↓

101 110 111

1.3 二进制数的运算

1.3.1 计算机为什么采用二进制和十六进制

1. 采用二进制的原因

二进制并不符合人们的习惯，但是计算机内部仍采用二进制表示信息。其主要原因有以

下四点：

(1) 电路简单。

计算机是由逻辑电路组成的，逻辑电路通常只有两个状态。例如开关的接通与断开，晶体管的饱和与截止，电压电平的高与低等。这两种状态正好与二进制数的两个数码 0 和 1 相似，可由这两个数码分别代表这两种状态。若是采用十进制，则需表示十个数码，这是很困难的。

(2) 工作可靠。

两个状态表示的二进制两个数码，数字传输和处理不容易出错，因而电路更加可靠。

(3) 运算简单。

二进制运算法则简单。例如，求和法则只有 3 个，求积法则也只有 3 个。若采用十进制，则其运算法则在电路上实现是很困难的。

(4) 逻辑性强。

计算机的工作机理是建立在逻辑运算的基础上，逻辑代数是逻辑运算的理论依据。二进制只有两个数码，正好代表逻辑代数中的“真”与“假”。

2. 采用十六进制的原因

至于计算机为什么采用十六进制，如果你已理解计算机采用二进制的原因，则此问题就迎刃而解了。认真分析一下二进制数和十六进制数，你就会发现二进制数与十六进制数有太多的共同点，甚至可以说十六进制数是二进制数的简写形式。下面看一个例子。

有这么一个二进制数，如下：

011111011110010110100011110010001111

这密密麻麻的一长串 1 和 0，对阅读与记忆是十分不便的，于是人们采用十六进制数来缩短它，得如下数：

7DE5A3C8F

1.3.2　二进制数的算术运算

1. 二进制数的运算

因为二进制数只有 0 和 1 两个数字，所以它的四则运算特别简单。其运算规则如表 1-3(a)、表 1-3(b)、表 1-3(c)与表 1-3(d)所示：

表 1-3(a)　加法

+	0	1
0	0	1
1	1	10

表 1-3(b)　减法

−	0	1
0	0	1
1	1	0

表 1-3(c)　乘法

×	0	1
0	0	0
1	0	1

表 1-3(d)　除法

·	0	1
0	无意义	0
1	无意义	1

对于加法运算，遵循“逢二进一”的法则，对于减法运算，遵循“借一当二”的法则；对于乘

法运算，由于二进制乘数中只有 1 和 0 两种情况，相乘运算要比十进制数相乘的“九九乘法表”法则简单得多。

例如：

$(1011011)_2+(1010.11)_2=?$

```
     1011011
+)   1010.11
────────────
  1100101.11
```

$(1010110)_2-(1101.11)_2=?$

```
     1010110
-)   1101.11
────────────
  1001000.01
```

$(1011.01)_2\times(101)_2=?$

```
     1011.01
×)       101
────────────
      101101
     000000
+) 101101
────────────
   111000.01
```

$(100100.01)_2\div(101)_2=?$

```
           111.01
      ───────────
101  / 100100.01
-)     101
      ───────────
        1000
    -)   101
      ───────────
          110
    -)    101
      ───────────
          0101
           101
      ───────────
             0
```

由上式可见，二进制乘法运算可归结为“加法与移位”；二进制除法运算可归结为“减法与移位”。做二进制除法的方法与做十进制除法的方法相同，在列竖式计算时，够除则在商上写 1，不够除则写 0，按此方法依次除下去，直到余数为零为止，在除不尽的情况下，根据需要计算到指定的精度即可。

2. 十六进制数的运算

十六进制数的运算可以采用先把该十六进制数转换为十进制数，经过计算后再把结果转换为十六进制数据的方法，但这样做比较繁琐。其实，按照逢 16 进 1 的规则，直接用十六进制数来计算也是很方便的。

(1) 十六进制加法：当两个一位数之和 S 小于 16 时，与十进制数同样处理，如两个一位数之和 S 大于或等于 16 时，则应用 S 减去 16 所得余数及进位 1 来取代 S。

(2) 十六进制数据的减法也与十进制数类似，够减时可直接相减，不够减时服从向高位借 1 为 16 的规则。

(3) 十六进制数的乘法可以用十进制数的乘法规则来计算，但结果必须用十六进制数来表示。

(4) 十六进制数的除法可以根据其乘法和减法规则处理，这里不再赘述。

1.3.3 二进制数的逻辑运算

逻辑是指“条件”与“结论”之间的关系。因此，逻辑运算是指对“因果关系”进行分析的一种运算，运算结果并不表示数值大小，而是表示逻辑概念成立还是不成立。

计算机中的逻辑关系是一种二值逻辑，二值逻辑很容易用二进制的“0”或“1”表示，例

如:"真"与"假"、"是"与"否"、"成立"与"不成立"等。若干位二进制数组成的逻辑数据,位与位之间无"权"的内在联系,对两个逻辑数据进行运算时,每位之间相互独立,运算是按位进行的,不存在算术运算的进位与借位,运算结果也是逻辑数据。

在逻辑代数中有三种基本的逻辑关系:**与**、**或**、**非**。其他复杂的逻辑关系均可由这三个基本逻辑关系组合而成。

(1)"与"逻辑。

做一件事情取决于多种因素时,当且仅当所有因素都满足时才去做,否则就不做,这种因果关系称为"与"逻辑。用来表示和推演"与"逻辑关系的运算称为"与"运算,常用·、∧、∩或 AND 等运算符来表示,"与"运算规则列于表 1-4 中,两个二进制数进行"与"运算是按位进行的。

两个逻辑变量 a、b 进行"与"运算,在数学上可记为 F=a AND b,F 是 A、B 的逻辑函数。对于 F=a AND b,由"与"运算规则知:当且仅当 A=1、B=1 时,才有 F=1,否则 F=0。

(2)"或"逻辑。

做一件事情取决于多种因素时,只要其中有一个因素得到满足就去做,这种因果关系称"或"逻辑。"或"运算常用+、∨、∪或 OR 等运算符来表示,"或"运算规则列于表 1-5 中,两个二进制数进行"或"运算是按位进行的。

(3)"非"逻辑。

"非"逻辑实现逻辑否定,即进行"求反"运算,常在逻辑变量上面加一横线表示。例如,A 的"非"写成 $\overline{A}$。非运算规则列于表 1-6 中。

表 1-4 AND 运算

a	b	a AND b
1	1	1
1	0	0
0	1	0
0	0	0

表 1-5 OR 运算

a	b	a OR b
1	1	1
1	0	1
0	1	1
0	0	0

表 1-6 NOT 运算

a	NOT a
1	0
0	1

例如:

X=10111001,Y=11110011
求 X AND Y=?

```
          10111001
    AND)  11110011
    ---------------
          10110001
```

X=10100001,Y=10011011
求 X OR Y=?

```
          10100001
    OR)   10011011
    ---------------
          10111011
```

X=10100001,求 NOT X=?
NOT X =01011110

1.4 数据单位、字符编码、汉字编码

1.4.1 计算机中的数据单位

计算机中用到的信息单位主要有位、字节、字等。