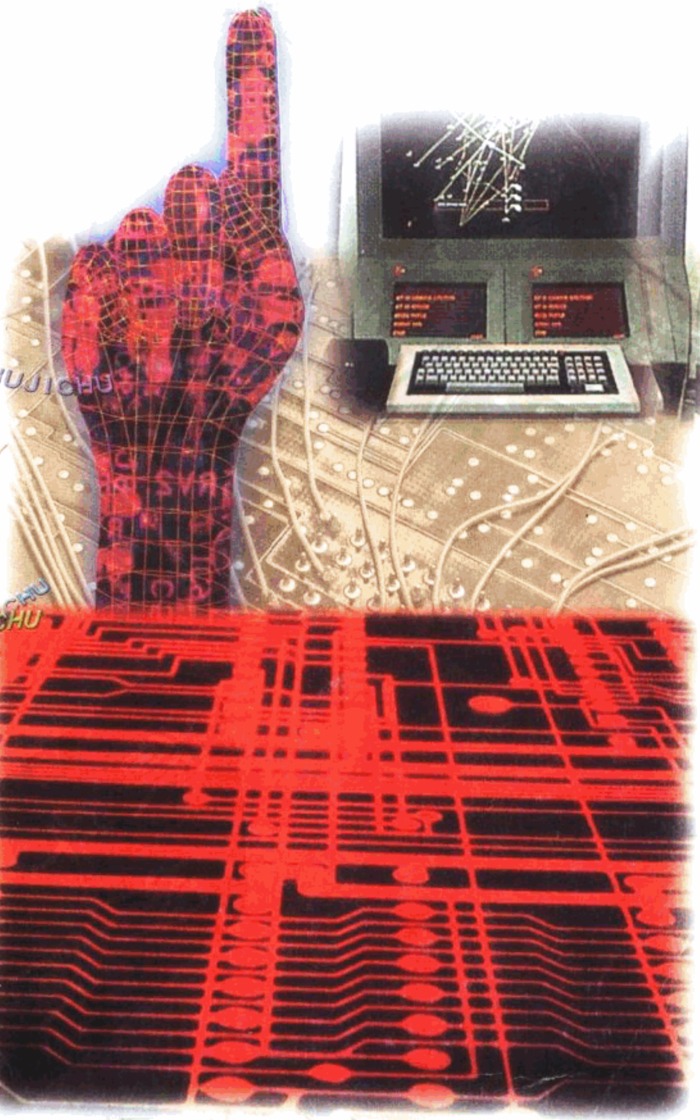
·21世纪高等学校教材·

电子技术基础

DIANZIJISHUJICHU

·夏路易 阎宏印 王永强 主编·

上册



兵器工业出版社

出版说明

目前，高校正处于新的一轮教学改革时期，许多院校进行了重组和专业调整，以适应新世纪对不同类型高技术人材的需求。这样，原来的教材体系结构很难适应新的要求，不能满足调整后专业的需要，必须进行相应的改革。为此，教育部启动了“面向21世纪高等工程教育教学内容和课程体系改革计划”，许多高等院校都进行了有益的探索研究和实践，并取得了一定的成果。

计算机、自动化、电子信息专业是当前发展最快的专业，课程内容更新快，变化大。为了满足各类高校以上专业教学改革和课程建设的需要，在有关部门和专家的大力支持下，我们组织有关高校的教师编写了计算机、自动化、电子信息类专业系列教材。

我们认为，新的教材必须是教学改革力度大，有创新精神，既要反应当前教学改革的需要，还应有一定的特色。在确保基本内容概念清晰、准确，深入浅出的前提下，还要反应出本门学科的最新发展。为此，我们组织有关高校的专家对这批教材进行了认真的审阅。

由于我们的水平有限，这批教材在编审和出版工作中还会存在一些缺点和不足，希望使用本教材的老师和同学以及广大读者提出宝贵意见，以使今后进一步提高教材的出版质量，使我们的工作能做的更好。

兵器工业出版社

前 言

本书是根据国家教育部工科课程教学指导委员会审定的教学大纲，结合我们多年的教学经验编写的专业技术基础课教材，可作为高等院校计算机、通信、自动控制、电力系统及相近专业“数字电子技术”和“模拟电子技术”的教材，也可作为其他专业和有关工程技术人员的参考书。

电子技术是当前发展最快的学科之一。电子器件已从原来的中、小规模集成电路发展到大规模、超大规模集成电路，从而对教学内容、教学方法和教材都提出了新的要求。为此，本书在编写时，在确保基本内容概念清晰、准确，深入浅出的前提下，力求反应出电子技术的最新发展，以适应新世纪对高技术人材的要求。

本书分为上下册，上册为数字电子技术，下册为模拟电子技术。上册分为9章，第1、2、3章分别介绍数制、编码、逻辑代数基础和集成门电路；第4、5、6章分别介绍组合逻辑电路、触发器电路和时序电路；第7、8、9章分别介绍半导体存储器、脉冲波形的产生与整形、数模与模数转换。下册分为12章，第10、11章分别介绍半导体器件基础知识和基本放大电路；第12、13、14、15章分别介绍差动电路、功放电路、集成运算电路和反馈放大电路；第16、17、18章分别介绍放大电路的频率特性、信号运算处理和发生电路；第19、20、21章分别介绍调制解调器、电流模拟技术和直流电源。本书各章均有一定数量的例题和习题，以提高读者分析问题、解决问题的能力。凡是书中有“*”的章节，可根据需要作为选修内容。

本书第1、2章由阎宏印编写，第3、16章由赵庆玲编写，第4、5、9章由秦建中编写，第6章由夏路易编写，第7、8章和数字电子技术附录由丁芳编写，第10、11章由李海芳编写，第12、13、14、18章由王淼编写，第15、17、19章由武娟萍编写，第20、21章由王永强编写。

本书由夏路易、阎宏印、王永强策划、统稿，由太原理工大学谢克明教授主审。在编写过程中，得到了太原理工大学王华奎、段富教授和许多老师的大力支持和帮助，在此表示衷心的感谢。

由于我们水平有限，书中难免存在错误和不妥之处，敬请各位老师、同学和读者提出批评和指正。

编　者

2001年5月

目　录

数字电子技术

此为试读，需要完整PDF请访问：www.ertongbook.com

数字电子技术

第1章　数制和编码

数制和编码是学习计算机、通信、自动化技术必须掌握的重要基础知识。本章将详细介绍进位计数制的基本概念和不同数制之间的转换方法，还将介绍数和字符在数字系统中的编码方法和表示方法。

1.1　进位计数制

计数制是指用一组固定的符号和统一的规则来表示数值的方法。如果按照进位的方法进行计数，则称为进位计数制。

1.1.1　基数和权

在进位计数制中，数的表示涉及到两个基本问题：权和基数。权是一个与相应数位有关的常数，它与该数位的数码相乘后，可得到该数位的数码代表的数值。一个数码处于不同的数位时，代表的数值不相同，因为它拥有的权不同。基数是一个正整数，它等于相邻数位上权的比。对任何一种进制的数，基数和能选用的数码的个数相等，能选用的最大数码要比基数小 1，每个数位能表示的最大数值是最大数码乘以该位具有的权，当超过这个数值时要向高位进位。

1.1.2　几种常用的进位计数制

在日常生活中，人们使用的是十进制计数制，而计算机中使用的是二进制计数制。为了阅读和书写方便，计算机技术中还使用八进制和十六进制计数制。

1．十进制数

采用十进位计数制的数称为十进制数，计数时“逢十进一”。十进制数的基数是 10，每一个数位可选用的数码有 10 个，即：0、1、2、3、4、5、6、7、8、9。对十进制数来说，其整数部分每一位的权，从右到左依次为 10^0，10^1，10^2，10^3，10^4，…，即平常所说的个、十、百、千、万，… 对小数部分每一位的权，从左到右依次为 10^{-1}，10^{-2}，10^{-3}，10^{-4}，…，即平常所说的十分之一、百分之一、千分之一、万分之一，… 可看出，相邻两数位权的比是 10。

对任意一个十进制数，都可以用一个多项式形式表示，其中每一项表示相应数位代表的数值。例如，十进制数 3784.25 可表示成：

$$(3784.25)_{10}=3\times10^3+7\times10^2+8\times10^1+4\times10^0+2\times10^{-1}+5\times10^{-2}$$

2．二进制数

采用二进位计数制的数称为二进制数，计数时"逢二进一"。二进制数的基数是2，每一个数位能选用的数码只有2个，即0和1。对二进制数来说，整数部分每一位的权，从右到左依次为2^0，2^1，2^2，2^3，2^4，… 对小数部分每一位的权，从左到右依次为2^{-1}，2^{-2}，2^{-3}，2^{-4}，… 可看出，二进制数相邻两数位权的比是2。

对任意一个二进制数，也可以用一个多项式形式表示，其中每一项表示相应数位代表的数值。例如，二进制数1101.11可表示成：

$$(1101.11)_2=1\times2^3+1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-2}$$

3．八进制数

采用八进位计数制的数称为八进制数，计数时"逢八进一"。八进制数的基数是8，每一个数位可选用的数码有8个，即：0、1、2、3、4、5、6、7。对八进制数来说，其整数部分每一位的权，从右到左依次为8^0，8^1，8^2，8^3，8^4，… 对小数部分每一位的权，从左到右依次为8^{-1}，8^{-2}，8^{-3}，8^{-4}，… 可看出，相邻两数位权的比是8。

对任意一个八进制数，也可以用一个多项式形式表示，其中每一项表示相应数位代表的数值。例如，八进制数1744.25可表示成：

$$(1744.25)_8=1\times8^3+7\times8^2+4\times8^1+4\times8^0+2\times8^{-1}+5\times8^{-2}$$

4．十六进制数

采用十六进位计数制的数称为十六进制数，计数时"逢十六进一"。十六进制数的基数是16，每一个数位可选用的数码有16个，前10个数码和十进制一样，后6个采用英文字符表示，这16个数码是：0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F。对十六进制数来说，其整数部分每一位的权，从右到左依次为16^0，16^1，16^2，16^3，16^4，… 对小数部分每一位的权，从左到右依次为16^{-1}，16^{-2}，16^{-3}，16^{-4}，… 可看出，相邻两数位权的比是16。

对任意一个十六进制数，也可以用一个多项式形式表示，其中每一项表示相应数位代表的数值。例如，十六进制数1A4.2C可表示成：

$$(1A4.2C)_{16}=1\times16^2+10\times16^1+4\times16^0+2\times16^{-1}+12\times16^{-2}$$

对以上几种进位计数制进行归纳，可对R（R为正整数）进位计数制的特点总结如下：

（1）R进位计数制的基数是R，各数位能选用的数码的个数为基数R，最大的数码应比基数小1。

（2）每一数位都有一个权，权是基数R的整次幂，幂的大小取决于该数码所在的位置，而相邻两数位权的比正好为基数R。

（3）每一个数位的数码代表的数值，等于该数码乘以该数位的权。

（4）计数规则是"逢R进一"。

因此，对任意一个R进制数N都可表示成下面的形式：

$$(N)_R=A_n\times R^n+A_{n-1}\times R^{n-1}+\cdots+A_0\times R^0+A_{-1}\times R^{-1}+\cdots+A_{-m}\times R^{-m}$$

其中，A_i的取值只能是在允许的范围内，第i位的权是R^i。等式左边右括号下角的数表示进制。

1.2 不同进位计数制间的转换

计算机中存储数据和对数据进行运算采用的是二进制数，当把数据输入到计算机中，或者从计算机中输出数据时，要进行不同计数制之间的转换。数制转换的实质是基数之间的转换，转换的原则是：如果两个有理数相等，则两个数的整数部分和小数部分肯定分别相等。

1.2.1 非十进制数到十进制数的转换

非十进制数转换成十进制数一般采用的方法是按权相加。这种方法是按照十进制数的运算规则，将非十进制数各位的数码乘以对应的权再累加起来。一个 R 进制数转换成十进制数的过程可用下式表示：

$$(A_n\cdots A_0.A_{-1}\cdots A_{-m})_R=(A_n\times R^n+\cdots+A_0\times R^0+A_{-1}\times R^{-1}+\cdots+A_{-m}\times R^{-m})_{10}$$

例 1-1 将$(11011.101)_2$转换成十进制数。

解：
$$\begin{aligned}(11011.101)_2&=(2^4+2^3+2^1+2^0+2^{-1}+2^{-3})_{10}\\&=(16+8+2+1+0.5+0.125)_{10}\\&=(27.625)_{10}\end{aligned}$$

例 1-2 将$(10101.01)_2$转换成十进制数。

解：
$$\begin{aligned}(10101.01)_2&=(2^4+2^2+2^0+2^{-2})_{10}\\&=(16+4+1+0.25)_{10}\\&=(21.25)_{10}\end{aligned}$$

在将二进制数转换成十进制数时，若 $A_i=1$，则该位代表的十进制数值是 2^i。若 $A_i=0$，则该位代表的十进制数值为 0，计算时，这一项可以省略，不必参加累加。

在二进制数到十进制数的转换过程中，要频繁的计算 2 的整次幂。下面列出了常用的 2 的整次幂和十进制数的对应关系，记住这些值，对今后的学习是十分有益的。

$2^0=1$	$2^1=2$	$2^2=4$	$2^3=8$
$2^4=16$	$2^5=32$	$2^6=64$	$2^7=128$
$2^8=256$	$2^9=512$	$2^{10}=1024$	$2^{11}=2048$
$2^{-1}=0.5$	$2^{-2}=0.25$	$2^{-3}=0.125$	$2^{-4}=0.0625$

例 1-3 将$(137)_8$转换成十进制数。

解：
$$\begin{aligned}(137.2)_8&=(1\times8^2+3\times8^1+7\times8^0+2\times8^{-1})_{10}\\&=(64+24+7+0.25)_{10}\\&=(95.25)_{10}\end{aligned}$$

例 1-4 将$(A7)_{16}$转换成十进制数。

解：
$$\begin{aligned}(A7.8)_{16}&=(10\times16^1+7\times16^0+8\times16^{-1})_{10}\\&=(160+7+0.5)_{10}\\&=(167.5)_{10}\end{aligned}$$

此为试读，需要完整PDF请访问：www.ertongbook.com

1.2.2 十进制数到非十进制数的转换

将十进制数转换成非十进制数时，整数部分和小数部部分要分别进行转换，整数部分的转换一般采用除基取余法，小数部分的转换一般采用乘基取整法。

1．十进制整数转换成非十进制整数

设十进制整数 N 转换成 R 进制整数的表示形式为 $(A_nA_{n-1}\cdots A_1A_0)_R$，由于每个 R 进制整数都可写成按权展开的多项式，所以有下式成立：

$$(N)_{10}=(A_n\times R^n+A_{n-1}\times R^{n-1}+\cdots+A_1\times R^1+A_0\times R^0)_R$$

那么，转换的关键是寻找多项式每一项的系数 A_n、A_{n-1}、… A_1、A_0，如果对上式两边同除以基数 R 可得：

$$(N/R)_{10}=(A_n\times R^{n-1}+A_{n-1}\times R^{n-2}+\cdots+A_1\times R^0)_R+A_0/R$$

在这个等式的右边，括号中为商的整数部分，A_0/R 为小数部分。由于等式两边整数和小数应分别对应相等，所以 N/R 的余数就是 A_0。同理，再用 N/R 的整数部分除以 R，余数就是 A_1。依次类推，直到整数商为 0，求出 A_n 为止。

例 1-5　将$(288)_{10}$转换成二进制数。

解：
288/2＝144	余数为 0，	$A_0=0$
144/2＝72	余数为 0，	$A_1=0$
72/2＝36	余数为 0，	$A_2=0$
36/2＝18	余数为 0，	$A_3=0$
18/2＝9	余数为 0，	$A_4=0$
9/2＝4	余数为 1，	$A_5=1$
4/2＝2	余数为 0，	$A_6=0$
2/2＝1	余数为 0，	$A_7=0$
1/2＝0	余数为 1，	$A_8=1$

所以，$(288)_{10}=(100100000)_2$

例 1-6　将$(62)_{10}$转换成八进制数。

解：
62/8＝7	余数为 6，	$A_0=6$
7/8＝0	余数为 7，	$A_1=7$

所以，$(62)_{10}=(76)_8$

例 1-7　将$(108)_{10}$转换成十六进制数。

解：
108/16＝6	余数为 12，	$A_0=C$
6/16＝0	余数为 6，	$A_1=6$

所以，$(108)_{10}=(6C)_{16}$

2．十进制小数转换成非十进制小数

设十进制小数 N 转换成 R 进制小数的表示形式为 $(0.A_{-1}A_{-2}\cdots A_{-(n-1)}A_{-n})_R$，也可写成按权展开的多项式，因此有下式成立：

$$(N)_{10}=(A_{-1}\times R^{-1}+A_{-2}\times R^{-2}+\cdots+A_{-(n-1)}\times R^{-(n-1)}+A_{-n}\times R^{-n})_R$$

转换的关键同样是确定多项式中每一项的系数 A_{-1}、A_{-2}、… 、$A_{-(n-1)}$、A_{-n}，如果将上式的两边同乘以 R，可得：

$$(R\times N)_{10}=A_{-1}+(A_{-2}\times R^{-1}+\cdots+A_{-(n-1)}\times R^{-(n-2)}+A_{-n}R^{-(n-1)})_R$$

根据等式两边整数部分和小数部分应对应相等的原则，$R\times N$ 的整数部分就是 A_{-1}。同理，再对等式两边的小数部分乘以 R，可求得 A_{-2}。依次类推，直到求得 A_{-n} 为止。

例 1-8　将$(0.625)_{10}$转换成二进制数。

解：　$0.625\times2=1+0.25$　　$A_{-1}=1$

　　　$0.25\times2=0+0.5$　　$A_{-2}=0$

　　　$0.5\times2=1+0$　　$A_{-3}=1$

所以，$(0.625)_{10}=(0.101)_2$

例 1-9　将$(0.8125)_{10}$转换成二进制数。

解：　$0.8125\times2=1+0.625$　　$A_{-1}=1$

　　　$0.625\times2=1+0.25$　　$A_{-2}=1$

　　　$0.25\times2=0+0.5$　　$A_{-3}=0$

　　　$0.5\times2=1+0$　　$A_{-4}=1$

所以，$(0.8125)_{10}=(0.1101)_2$

例 1-10　将$(0.8125)_{10}$转换成八进制数。

解：　$0.8125\times8=6+0.5$　　$A_{-1}=6$

　　　$0.5\times8=4+0$　　$A_{-2}=4$

所以，$(0.8125)_{10}=(0.64)_8$

例 1-11　将$(0.8125)_{10}$转换成十六进制数。

解：　$0.8125\times16=13+0$　　$A_{-1}=\mathrm{D}$

所以，$(0.8125)_{10}=(0.\mathrm{D})_{16}$

由于不是所有的十进制小数都能用有限位 R 进制小数来表示，因此，在转换过程中可根据精度要求取一定的位数即可。若要求误差小于 R^{-n}，则转换取小数点后 n 位就能满足要求。

例 1-12　将$(0.7)_{10}$转换成二进制数，要求误差小于 2^{-6}。

解：　$0.7\times2=1+0.4$　　$A_{-1}=1$

　　　$0.4\times2=0+0.8$　　$A_{-2}=0$

　　　$0.8\times2=1+0.6$　　$A_{-3}=1$

　　　$0.6\times2=1+0.2$　　$A_{-4}=1$

　　　$0.2\times2=0+0.4$　　$A_{-5}=0$

　　　$0.4\times2=0+0.8$　　$A_{-6}=0$

所以，$(0.7)_{10}=(0.101100)_2$

由于最后剩下未转换的部分，即误差在转换过程中扩大了 2^6，所以真正的误差应该是：0.8×2^{-6}，满足精度要求。

1.2.3　非十进制数之间的转换

非十进制数之间的转换主要是指二进制数、八进制数、十六进制数之间的转换。从理论上讲，任意两种数制的数都可以直接转换，但所用的一些计算规则却是一般人不熟悉的。如在八进制运算中，$(3\times7)_8=(25)_8$，而在十六进制运算中，$(3\times7)_{16}=(15)_{16}$。由于二进制数和八进制数、十六进制数的基数之间有整次幂的关系，所以二进制数和八进制数、十六进制数之间的转换可以采用一种十分简单的方法。

1．二进制数和八进制数之间的转换

二进制数的基数是 2，八进制数的基数是 8，正好有 $2^3=8$。因此，任意一位八进制数可以转换成三位二进制数。当要把一个八进制数转换成二进制数时，可以直接将每位八进制数码转换成三位二进制数码。而二进制数到八进制数的转换可按相反的过程进行，转换时，从小数点开始向两边分别将整数和小数每三位划分成一组，整数部分的最高一组不够三位时，在高位补 0，小数部分的最后一组不足三位时，在末位补 0，然后将每组的三位二进制数转换成一位八进制数即可。

例 1-13　将$(354.72)_8$转换成二进制数。

解：

$$\begin{array}{cccccc}
3 & 5 & 4 & . & 7 & 2 \\
\downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\
011 & 101 & 100 & . & 111 & 010
\end{array}$$

所以，$(354.72)_8=(011101100.111010)_2$

例 1-14　将$(1010110.0101)_2$转换成八进制数。

解：

$$\begin{array}{cccccc}
001 & 010 & 110 & . & 010 & 100 \\
\downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\
1 & 2 & 6 & . & 2 & 4
\end{array}$$

所以，$(1010110.0101)_2=(126.24)_8$

2．二进制数和十六进制数之间的转换

二进制数的基数是 2，十六进制数的基数是 16，正好有 $2^4=16$。因此，任意一位十六进制数可以转换成四位二进制数。当要把一个十六进制数转换成二进制数时，可以直接将每位十六进制数码转换成四位二进制数码。对二进制数到十六进制数的转换可按相反的过程进行，转换时，从小数点开始向两边分别将整数和小数每四位划分成一组，整数部分的最高一组不够四位时，在高位补 0，小数部分的最后一组不足四位时，在末位补 0，然后将每组的四位二进制数转换成一位十六进制数即可。

例 1-15　将$(8E.3A)_{16}$转换成二进制数。

解：

$$\begin{array}{ccccc}
8 & E & . & 3 & A \\
\downarrow & \downarrow & & \downarrow & \downarrow \\
1000 & 1110 & . & 0011 & 1010
\end{array}$$

所以，$(8E.3A)_8=(10001110.00111010)_2$

例 1-16　将$(1011111.101101)_2$转换成十六进制数。

解：

$$\begin{array}{ccccc}
0101 & 1111 & . & 1011 & 0100 \\
\downarrow & \downarrow & & \downarrow & \downarrow \\
5 & F & . & B & 4
\end{array}$$

所以，$(1011111.101101)_2=(5F.B4)_{16}$

3．八进制数和十六进制数之间的转换

八进制数和十六进制数之间的转换，直接进行比较困难，可用二进制数作为转换中介，即先转换成二进制数，再进行转换就比较容易了。

例 1-17　将$(345.27)_8$转换成十六进制数。

解：　3　4　5 . 2　7

↓　↓　↓　↓　↓

011　100　101 . 010　111　　先转换成二进制数

1110　0101 . 0101　1100　　重新分组

↓　↓　↓　↓

E　5 . 5　C　　转换成十六进制数

所以，$(345.27)_8=(E5.5C)_{16}$

例 1-18　将$(2B.A6)_{16}$转换成八进制数。

解：　2　B . A　6

↓　↓　↓　↓

0010　1011 . 1010　0110　　先转换成二进制数

101　011 . 101　001　100　　重新分组

↓　↓　↓　↓　↓

5　3 . 5　1　4　　转换成八进制数

所以，$(2B.A6)_{16}=(53.514)_8$

为了方便初学者，表 1-1 列出了二进制和八进制、十进制、十六进制数之间的换算关系。

表 1-1　二进制和八进制、十进制、十六进制数之间的换算

二进制数	八进制数	十进制数	十六进制数
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F
10000	20	16	10

1.3　二进制数的特点和运算

1.3.1　二进制数的特点

计算机处理的数字、文字、图形、图像和声音信息都是采用二进制数编码的，那么为什么在计算机中要使用二进制数呢？这是由于二进制数具有的下列一些特点所决定的。

1．在计算机中容易实现

在二进制数中，只需要 0 和 1 两个数码就可以表示任意数。这样，每一位二进制数可以用具有两种不同状态的电路或元件表示。如电平的高低、晶体管的导通与截止、磁性元件的

正负磁化状态，可分别用来表示 0 和 1。由于制造具有两种状态的元件比制造具有多种状态的元件要容易的多，而且两种状态间的转换也比多种状态间的转换要容易控制和实现，这是计算机采用二进制数的一个主要原因。

2．运算规则简单

由于数的运算规则随基数的增大而变得复杂，对 R 进制的数，在作加法运算时需要记住 $R(R+1)/2$ 条规则，作乘法运算需要记住同样的规则。如十进制数的乘法口诀有 55 条，而二进制数由于在所有的进位记数制中基数最小，因而规则也最少，乘法规则仅有 3 条。运算规则少，将有利于计算机运算功能部件的制造和运算速度的提高。

3．工作可靠

表示二进制数的元件具有的两种状态，一般都是质的不同，十分稳定。如电平的正和负、晶体管的导通和截止、磁场的正和负。利用质的不同区别信息，要比利用量的不同区别信息更加可靠。它可以减少电源变化、元器件特性改变等因素产生的干扰，而且采用二进制还可以为计算机的自动纠错提供方便。当发现某一位有错时，将该位的状态变反即可实现纠正，这对其他进制的数来说是很难办到的。

4．适合逻辑运算

在计算机的非数值计算应用中，存在着大量的逻辑运算。二进制数的 1 和 0 正好与逻辑代数中的“真”与“假”、“是”与“非”这两种逻辑值相对应，为计算机在非数值计算领域的应用提供了强有力的支持

1.3.2 二进制数的算术运算

在各种进位计数制中，二进制的运算规则是最简单的。下面讨论二进制数的四则运算。

1．加法运算

二进制的加法运算规则为：

$$0+0=0 \qquad 0+1=1 \qquad 1+0=1 \qquad 1+1=10$$

例 1-19　已知 $A=10010110$，$B=01001110$， 求 $A+B$。

解：

$$\begin{array}{r} 10010110 \\ +\ 01001110 \\ \hline 11100100 \end{array}$$

所以，$A+B=11100100$

例 1-20　已知 $A=11011101$，$B=10110011$，求 $A+B$。

解：

$$\begin{array}{r} 11011101 \\ +10110011 \\ \hline 110010000 \end{array}$$

所以，$A+B=110010000$

可见，二进制的加法运算非常简单。根据运算规则可以求出每一位的和，当和大于等于 2 时要向高位进 1。如本位相加的两个数都是 1，而低位又有一个进位， 则该位相当于 3 个 1 相加，本位的和是 1，同时向高位进 1。

2．减法运算

二进制减法运算的规则为：

$$0-0=0 \qquad 1-0=1 \qquad 1-1=0 \qquad 0-1=1 \quad \text{(有借位)}$$

例 1-21　已知 A=11101010，B=10101011，求 $A-B$。

解：

$$\begin{array}{r} 11101010 \\ -\ 10101011 \\ \hline 00111111 \end{array}$$

所以，$A-B$=00111111

例 1-22　已知 A=101011，B=110011，求 $A-B$。

解：因为 $A<B$，所以应该用 B 去减 A，且差为负。

$$\begin{array}{r} 110011 \\ -101011 \\ \hline 001000 \end{array}$$

所以，$A-B$=－001000

在二进制减法运算中，当某一位不够减时，应向高位借 1，而借来的 1 是当 2 用的，类似于十进制减法的"借 1 当 10"。如被减数小于减数时，差是负的，则应该用减数去减被减数。在计算机中实现减法运算，实际上是用补码加法完成的，不必考虑操作数的大小。

3．乘法运算

二进制乘法运算的规则为：

$$0\times0=0 \qquad 1\times0=0 \qquad 0\times1=0 \qquad 1\times1=1$$

在二进制乘法中，乘数的每一位只有 0 和 1 两种情况，这使得乘法运算变得非常容易。当乘数的某位是 1 时，对应的部分积是被乘数。当乘数的某位是 0 时，对应的部分积是 0，计算时可以省略。从低位到高位逐次求出乘数每一位对应的部分积，累加即可求得两个二进制数的积。在求累加和时，当某一位的和大于等于 2 时向高位进 1，大于等于 4 时，向高位进 2，其余依次类推。

例 1-23　已知 A=10101，B=10011，　求 $A\times B$。

解：

$$\begin{array}{r} 10101 \\ \times\ 10011 \\ \hline 10101 \\ 10101 \\ 10101 \\ \hline 110001111 \end{array}$$

所以，$A\times B$=110001111

在计算机中实现乘法一般采用移位相加的方法进行。

4．除法运算

二进制的除法运算是乘法运算的逆运算，和十进制除法运算过程是类似的，但要简单的多，每一位的商只有 1 和 0 两种选择。在除法运算中，当除数是 n 位时，首先将除数和被除数的高 n 位比较，当除数小于等于被除数的最高 n 位时，商为 1，然后从被除数中减去除数得部分余数；否则，商为 0，被除数成为部分余数。重复上述过程，将除数和部分余数比较，直到被除数的所有位都处理完为止。

例 1-24　已知 A=110111，B=101，　求 $A\div B$。

此为试读，需要完整PDF请访问：www.ertongbook.com