普通高等教育"十一五"国家级规划教材

21世纪

高等院校电子信息类规划教材

Shuzi Dianlu

Luoji Sheji

数字电路逻辑设计

◎ 马义忠 常蓬彬 马浚 张晓帆 编著

人民邮电出版社
POSTS & TELECOM PRESS

普通高等教育“十一五”国家级规划教材

21世纪高等院校电子信息类规划教材

数字电路逻辑设计

马义忠　常蓬彬　马　浚　张晓帆　编著

人民邮电出版社

北　京

图书在版编目（CIP）数据

数字电路逻辑设计/马义忠等编著．—北京：人民邮电出版社，2007.9

21 世纪高等院校电子信息类规划教材．普通高等教育“十一五”国家级规划教材

ISBN 978-7-115-16105-5

Ⅰ.数... Ⅱ.马... Ⅲ.数字电路—逻辑设计—高等学校—教材 Ⅳ.TN79

中国版本图书馆 CIP 数据核字（2007）第 053946 号

内容提要

本书以知识性、实用性和先进性为宗旨，结合综合型人才培养目标和教学特点，从教材体系结构及内容的安排上，紧紧围绕抽象—理论—设计三个形态，优化课程结构，精练教学内容，拓宽专业基础，特别注重综合构思素质的培养。全面而系统地阐述了数字电路逻辑设计的基本理论、分析方法和设计原理，强化基础，突出应用。增加了电子设计自动化的内容，使数字逻辑电路的设计从传统的单纯硬件设计方法变为计算机软硬件协同设计。

全书共 10 章，内容涉及数字逻辑基础、逻辑代数与逻辑函数、集成逻辑门电路及工作原理、组合逻辑电路、集成双稳态触发器、同步时序逻辑电路、异步时序逻辑电路、可编程逻辑器件、D/A 及 A/D 转换和 EDA 技术等。

本书结构与内容新颖，力求反映数字电路逻辑设计的最新发展技术；叙述上通俗易懂，层次上由浅入深，分析与设计方法上灵活多样，收集了应用设计的实例，使读者容易掌握并应用。

本书作为普通高校计算机科学与技术、通信、电子信息以及自动化等专业的教材，也可作为成人教育的教材和相关专业技术人员的参考书。

普通高等教育“十一五”国家级规划教材

21 世纪高等院校电子信息类规划教材

数字电路逻辑设计

◆ 编　　著　马义忠　常蓬彬　马　浚　张晓帆

　责任编辑　滑　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　电子函件　315@ptpress.com.cn

　网址　http://www.ptpress.com.cn

　北京通州大中印刷厂印刷

　新华书店总店北京发行所经销

◆ 开本：787×1092　1/16

　印张：17.25

　字数：413 千字　　2007 年 9 月第 1 版

　印数：1-3 000 册　　2007 年 9 月北京第 1 次印刷

ISBN 978-7-115-16105-5/TN

定价：25.00 元

读者服务热线：（010）67170985　印装质量热线：（010）67129223

前　言

处于新世纪“十一五”中的中国高等教育，面临世界范围的综合素质竞争，实质上是科学技术的竞争和民族素质的竞争。面对中国高等教育的专业结构、课程体系、教学内容、知识结构和教学方法，要从时代与技术发展的总体考虑，必须进行系统结构上的调整与改革，跟踪世界先进水平。按照当前数字器件发展和大学教学改革的需求，从逻辑抽象、综合设计和创新构思三个方面进行数字电路逻辑设计教材编写，突出以下几个方面。

1. 根据数字逻辑器件的发展历程，系统地阐述数字电路逻辑设计的基本理论、分析方法和设计原理。对实际问题进行逻辑抽象，应用逻辑代数描述数字逻辑电路，用这一过程来建立和理解数字逻辑电路所依据的数学原理。抽象是模型化的过程，为数据采集方法和假设的形式说明，抽象的结果是概念、符号、模型。设计用来开发求解给定问题的系统；为设计与实现方法、测试和分析手段、需求与规格进行说明。从基本原理与应用出发，使数字逻辑电路的分析与设计方法更加灵活多样。

2. 从教材体系结构及内容的安排上，紧紧围绕抽象—理论—设计三个形态，优化课程结构，精练教学内容，拓宽专业基础，特别注重综合构思素质的培养。对原来数字逻辑方面的内容作较大的调整。结合工程设计与现场可编程，强化学生利用基础知识解决工程问题的训练。

3. 引入 VHDL，使数字逻辑与数字系统的设计从传统的单纯硬件设计方法改变为现代计算机软件、硬件协同设计。有利于缩短课堂教学与工程技术问题的距离，加速教学改革。

4. 调整课后部分练习，加强综合设计、实际应用与创新方面的训练。

为了适应电子技术的迅猛发展，尤其是现代数字电子技术新型器件层出不穷，分析设计方法推陈出新，本书在介绍基础知识的同时，还精选一些代表当今数字电子技术发展水平的新技术和新方法作为教学内容，力求做到基本概念清晰，内容全面，定位准确，技术先进，有较强的可读性。

全书共分 10 章。第 1 章介绍数字逻辑基础，第 2 章是逻辑代数与逻辑函数基础知识，第 3 章阐述集成逻辑门电路及工作原理，第 4 章讨论组合逻辑电路，第 5 章介绍集成双稳态触发器，第 6 章讲述同步时序逻辑电路，第 7 章是异步时序逻辑电路的介绍，第 8 章是可编程逻辑器件，第 9 章分析 D/A 及 A/D 转换，第 10 章介绍 EDA 与 VHDL 技术。

本书在编写过程中，得到了兰州大学教务处领导的大力支持，并得到兰州大学信息科学与工程学院田亚菲教授的指导，在此一并表示衷心感谢。

由于数字逻辑技术发展迅速，加之作者水平有限，书中存在错误和疏漏之处在所难免，敬请读者批评指正。

编　者
2007 年 1 月于兰州大学

目　　录

第 1 章　数字逻辑基础

数字逻辑电路主要研究数字信息的产生、存储、变换及运算等，其分析及设计方法是信息科学的专业基础知识，二进制和逻辑代数是数字逻辑电路中进行算术运算及逻辑运算的主要数学工具。本章主要介绍进位计数制的表示方法，以二进制为重点，讨论各种进制数的相互转换，带符号数的代码表示法，码制和字符的编码方法等。

1.1　进位计数制

电信号可分为模拟信号与数字信号两大类，模拟信号是指在时间上和幅值上都是连续变化的电压或电流信号，处理这种模拟信号的电路统称为模拟电路。

所谓数字信号是指在时间上和幅值上都是离散的（不连续的）电压或电流信号，如图 1-1 所示，一方面，它们的变化在时间上是不连续的，总是发生在一系列离散的瞬间 $t_i(i = 0,1,2,\cdots)$；另一方面，它们的幅值大小及增减变化都采取数字形式，如图 1-1 中，t_2 时幅值由 2 跳变 1，t_4 时幅值由 3 跳变为 5 等。实际上，目前在数字电路中最常采用的是只有 0 和 1 两种数值组成的数字信号，如图 1-2 所示，这种两值数字信号又可分为电位型和脉冲型两种，图 1-2（a）是电位型数字信号，用高电位表示数字 1，低电位表示数字 0；每个 1，0 信号所占的时间间隔 Δt 都相等，并称之为一拍，一般一个 1 或一个 0 称为 1 比特（bit）或 1 位，几个连续的高（低）电位，就是几拍（位）长的 1 或 0 信号，图 1-2（a）所示为 11010 五位数字信号，图 1-2（b）是脉冲型数字信号，它是以每拍内有无脉冲来表示的信号，在一拍时间内有脉冲

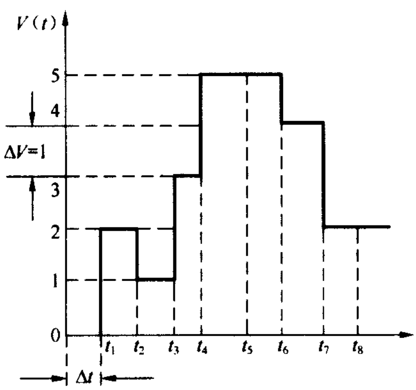

图 1-1　数字信号例 1

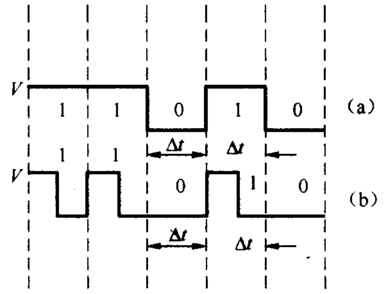

图 1-2　数字信号例 2

信号表示数字1，无脉冲信号表示数字0，图1-2（b）所示也是11010五位数字信号。

处理数字信号的电路称为数字电路，它能产生、存储、变换、传送数字信号以及具有对数字信号进行算术运算、逻辑运算、计数、显示等功能。由于数字电路的输入变量与输出变量都是数字信号，而且相互之间又都有一定的逻辑关系，这些逻辑关系常用一系列逻辑符号来表示，所以数字电路又称为数字逻辑电路或逻辑电路。

1.1.1 数制与十进制数的表示

用数字量表示物理量的大小时，仅用一位数码往往不够用。因此，用一组统一的符号和规则表示数的方法，常用进位计数的方法组成多位数码使用。一组多位数码中每一位的构成方法以及从低位到高位的进位规则称为数制。

原则上说，一个数可以用任何一种进位计数制来表示和运算，但不同数制其运算方法及难易程度互不相同。选择什么样的进位计数制来表示数，对数字系统的性能影响很大。

在日常生活中，人们通常采用十进制数来计数，每位数可用10个数码之一来表示，即0，1,2,3,4,5,6,7,8,9。十进制的基数为10，基数表示进位制所具有的数字符号个数，十进制具有的数字符号个数为10。

十进制数的计算规律是由低位向高位进位时“逢十进一”，也就是说，每位累计不能超过9，计满10就应向高位进1。

当人们看到一个十进制数，如632.45时，就会立刻想到：这个数的最左位（即第1位）为百位（6代表600），第2位为十位（3代表30），第3位为个位（2代表2），小数点右面第1位为十分位（4代表4/10），第2位为百分位（5代表5/100）。这里百、十、个、十分之一和百分之一都是10的次幂。在一个进位制表示的数中，不同数位上的固定常数称之为“权”。十进制数632.45按权展开的形式如下

$$632.45 = 6\times10^{2} + 3\times10^{1} + 2\times10^{0} + 4\times10^{-1} + 5\times10^{-2}$$

等式左边的表示方法称为位置计数表示法，等式右边则是其按权展开表示法。

一般说来，对于任意一个十进制数N，可用位置计数法表示为

$$(N)_{10} = (k_{n-1}k_{n-2}\cdots k_1k_0 . k_{-1}k_{-2}\cdots k_{-m})_{10}$$

也可用按权展开法表示为

$$\begin{aligned}(N)_{10} &= k_{n-1}\times10^{n-1} + k_{n-2}\times10^{n-2} + \cdots + k_1\times10^{1} + k_0\times10^{0}\\ &\quad + k_{-1}\times10^{-1} + k_{-2}\times10^{-2} + \cdots + k_{-m}\times10^{-m}\\ &= \sum_{i=-m}^{n-1} k_i\times10^{i}\end{aligned}$$

式中：k_i表示各个数字符号，为0～9这10个数码中的任意一个；n为整数部分的位数；m为小数部分的位数。

通常，对十进制数的表示，可以在数字的右下角标注10或D。

1.1.2 二进制数的表示

数字系统中使用的进位制并不仅限于十进制，当进位基数为2时，称为二进制。在二进

制中，只有 0 和 1 两个数码。二进制的计数规则是由低位向高位“逢二进一”，即每位计满 2 就向高位进 1，例如，$(1101)_2$ 就是一个二进制数。不同数位的数码表示的值不同，各位的权值是以 2 为底的连续整数幂，从右向左递增。

对于任意一个二进制数 N，用位置计数法表示为

$$(N)_2 = (k_{n-1}k_{n-2}\cdots k_1k_0 \cdot k_{-1}k_{-2}\cdots k_{-m})_2$$

用按权展开法表示为

$$\begin{aligned}(N)_2 &= k_{n-1}\times 2^{n-1} + k_{n-2}\times 2^{n-2} + \cdots + k_1 \times 2^1 + k_0\times 2^0 + k_{-1}\times 2^{-1} \\ &\quad + k_{-2}\times 2^{-2} + \cdots + k_{-m}\times 2^{-m} \\ &= \sum_{i=-m}^{n-1} k_i \times 2^i\end{aligned}$$

式中：k_i 表示各个数字符号，为 0 或 1；n 为整数部分的位数；m 为小数部分的位数。

通常，对二进制数的表示，可以在数字右下角标注 2 或 B。

在数字系统中，常用二进制来表示数字和进行运算。因为它具有如下特点。

(1) 二进制数只有 0 或 1 两个数码，任何具有两个不同稳定状态的元件都可用来表示 1 位二制数，例如，晶体管的导通或截止，脉冲信号的“有”或“无”等。

(2) 二进制运算规则简单。其运算规则如表 1-1 所示。

表 1-1　　二进制数的运算规则

加法规则	减法规则	乘法规则	除法规则
0+0=0；1+0=1 0+1=1；1+1=0	0-0=0；0-1=1 1-0=1；1-1=0	0×0=0；0×1=0 1×0=0；1×1=1	0÷1=0；1÷1=1

下面举例说明。

例 1-1　进行 1101+1011 运算。

解：

$$\begin{array}{r} 1101 \\ +)\ 1011 \\ \hline 11000 \end{array}$$

两个二进制数的加法运算和十进制数的加法运算相似，但采用“逢二进一”的法则，每位数累计到 2 时，本位就记为 0，同时向相邻高位进 1。

例 1-2　进行 11101-10011 运算。

解：

$$\begin{array}{r} 11101 \\ -)\ 10011 \\ \hline 1010 \end{array}$$

二进制减法运算从低位起按位进行，在遇到 0 减 1 时，就要采用“借一当二”的法则向相邻高位借 1，也就是从那一高位减去 1。

例 1-3　进行 1101×1001 运算。

解：

$$\begin{array}{r} 1101 \\ \times)\ 1001 \\ \hline 1101 \\ 0000 \\ 0000 \\ 1101 \\ \hline 1110101 \end{array}$$

二进制的乘法运算和十进制数的乘法运算相似，所不同的是对部分积进行累加时要按“逢二进一”的法则。

例 1-4 进行 10010001 ÷ 1011 运算。

解:

```
                1 1 0 1 ………商
1 0 1 1 √1 0 0 1 0 0 0 1
          1 0 1 1
          -------
            1 1 1 0
            1 0 1 1
            -------
              1 1 0 1
              1 0 1 1
              -------
                  1 0 ………余数
```

二进制数的除法运算同十进制数的除法运算类似，但采用二进制数的运算规则。

(3) 二进制数只有两个状态，数字的传输和处理不容易出错，可靠性高。

(4) 二进制数的数码 0 和 1，可与逻辑代数中逻辑变量的值“假”和“真”对应起来。也就是说，可用一个逻辑变量来表示一个二进制数码。这样，在逻辑运算中就可以使用逻辑代数这一数学工具。

1.1.3 任意进制数的表示

二进制数运算规则简单，便于电路实现，它是数字系统中广泛应用的一种数制。但用二进制表示一个数时，所用的位数比用十进制数表示的位数多，人们读写很不方便，容易出错，不便记忆。因此，人们常采用八进制数和十六进制数，这两种数制不但容易书写和阅读，便于记忆，而且具有二进制数的特点，十分容易将它们转换成二进制数。

八进制数的基数是 8，采用的数码是0,1,2,3,4,5,6,7。计数规则是从低位向高位“逢八进一”，对于相邻两位来说，高位的权值是低位权值的 8 倍。例如，数$(47.6)_8$就表示一个八进制数。由于八进制的数码和十进制前 8 个数码相同，为了便于区分，通常在八进制数字的右下角标注 8 或 O。

十六进制数的基数为 16，采用的数码是0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F。其中 A,B,C,D,E,F 分别代表十进制数字10,11,12,13,14,15。十六进制的计算规则是从低位向高位“逢十六进一”，对于相邻两位来说，高位的权值是低位权值的 16 倍。例如，$(54AF.8B)_{16}$就是一个十六进制数。通常，在十六进制数字的右下角标注 16 或 H。

与二进制数一样，八进制数和十六进制数均可用位置计数法的形式和按权展开法的形式表示。

一般说来，对于任意的数 N，都能表示成以 R 为基数的 R 进制数。数 N 的位置记数法为

$$(N)_R = (k_{n-1}k_{n-2}\cdots k_1k_0 \cdot k_{-1}k_{-2}\cdots k_{-m})_R$$

而相应的按权展开表示法为

$$\begin{aligned}(N)_R &= k_{n-1}\times R^{n-1} + k_{n-2}\times R^{n-2} + \cdots + k_1\times R^1 + k_0\times R^0 \\ &\quad + k_{-1}\times R^{-1} + k_{-2}\times R^{-2} + \cdots + k_{-m}\times R^{-m} \\ &= \sum_{i=-m}^{n-1} k_i R^i\end{aligned}$$

式中：k_i 表示各个位的数字符号，为0 ~ $(R-1)$ 数码中任意一个；R 为进位制的基数；n 为整数部分的位数；m 为小数部分的位数。

R 进制的计数规则是从低位向高位“逢 R 进一”。

常用的不同数制的各种数码如表 1-2 所示，该表列出了当 R 为10,2,8,16时各种进位计数制中开头的 16 个自然数。

表 1-2　　**不同进位计数制的各种数码**

十进制数 ($R=10$)	二进制数 ($R=2$)	八进制数 ($R=8$)	十六进制数 ($R=16$)
0	0000	00	0
1	0001	01	1
2	0010	02	2
3	0011	03	3
4	0100	04	4
5	0101	05	5
6	0110	06	6
7	0111	07	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　数 制 转 换

1.2.1　二进制数与十进制数的转换

在计算机和其他数字系统中，普遍使用二进制数，采用二进制数的数字系统只能处理二进制数和用二进制代码形式表示的其他进制数。而人们习惯于使用十进制数，所以，在信息处理中首先必须把十进制数转换成计算机能处理的二进制数进行运算，然后，再将二进制数的计算结果转换成人们习惯的十进制数。

二进制数转换成十进制数是很方便的，只要将二进制数写成按权展开式，并将式中各乘积项的积算出来，然后各项相加，即可得到与该二进制数相对应的十进制数。例如

$$\begin{aligned}(11010.101)_2 &= 1\times2^4+1\times2^3+0\times2^2+1\times2^1+0\times2^0\\&\quad+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}\\&=16+8+2+0.5+0.125\\&=(26.625)_{10}\end{aligned}$$

十进制数转换成二进制数时，需将待转换的数分成整数部分和小数部分，并分别加以转换。一个十进制数可写成

$$(N)_{10}=(\text{整数部分})_{10}\cdot(\text{小数部分})_{10}$$

转换时，首先将（整数部分）$_{10}$转换成（整数部分）$_2$，然后，再将（小数部分）$_{10}$转换成（小数部分）$_2$。待整数部分和小数部分确定后，就可写成

$$(N)_2=(\text{整数部分})_2\cdot(\text{小数部分})_2$$

1. 整数转换

十进制数的整数部分采用“除 2 取余”法进行转换，即把十进制数除以 2，取出余数 1 或 0 作为相应二进制数的最低位，把得到的商再除以 2，取余数 1 或 0 作为二进制数的次低位，依次类推，继续上述过程，直到商为 0，所得余数为最高位。

例如，将十进制整数 58 转换为二进制整数，先把它写成如下形式

$$\begin{aligned}(58)_{10}&=(k_{n-1}k_{n-2}\cdots k_1k_0)_2\\&=k_{n-1}\times2^{n-1}+k_{n-2}\times2^{n-2}+\cdots+k_1\times2^1+k_0\times2^0\\&=2(k_{n-1}\times2^{n-2}+k_{n-2}\times2^{n-3}+\cdots+k_1)+k_0\end{aligned}$$

只要求出等式中的各个系数 k_{n-1}，k_{n-2}，…，k_1，k_0 便得到二进制整数。将上式两边除以 2，得

$$(29)_{10}=k_{n-1}\times2^{n-2}+k_{n-2}\times2^{n-3}+\cdots+k_1+\frac{k_0}{2}$$

两数相等，整数部分和小数部分必须对应相等，等式左边余数为 0，则 k_0 为 0。
因而得

$$(29)_{10}=2(k_{n-1}\times2^{n-3}+k_{n-2}\times2^{n-4}+\cdots+k_2)+k_1$$

将等式两边再除以 2，得

$$\left(14+\frac{1}{2}\right)_{10}=k_{n-1}\times2^{n-3}+k_{n-2}\times2^{n-4}+\cdots+k_2+\frac{k_1}{2}$$

比较等式两边，等式左边余数为 1，则取 k_1 为 1。

依次类推，可得系数 k_2，k_3，…，k_{n-2}，k_{n-1}。

根据上面讨论的方法，可用下列形式很方便地将十进制整数转换成二进制整数。

$$\begin{array}{r|l}
2 & 58 \\
2 & 29 \quad \longrightarrow k_0=0\ (\text{余数 0 最低位}) \\
2 & 14 \quad \longrightarrow k_1=1\ (\text{余 1}) \\
2 & 7 \quad \longrightarrow k_2=0\ (\text{余 0}) \\
2 & 3 \quad \longrightarrow k_3=1\ (\text{余 1}) \\
2 & 1 \quad \longrightarrow k_4=1\ (\text{余 1}) \\
 & 0 \quad \longrightarrow k_5=1\ (\text{余数 1 最高位})
\end{array}$$

因此，$(58)_{10}=(111010)_2$。

2. 纯小数转换

十进制数的小数部分采用“乘 2 取整”法进行转换，即先将十进制小数乘以 2，取其整数 1 或 0 作为二进制小数的最高位；然后将乘积的小数部分再乘以 2，并再取整数作为次高位。重复上述过程，直到小数部分为 0 或达到所要求的精度。

例如，将十进制小数 0.625 转换为二进制小数，需把它写成如下形式

$$
\begin{aligned}
(0.625)_{10} &= (0.k_{-1}k_{-2}\cdots k_{-m})_2 \\
&= k_{-1}\times 2^{-1}+k_{-2}\times 2^{-2}+\cdots+k_{-m}\times 2^{-m} \\
&= \frac{k_{-1}}{2}+\frac{1}{2}(k_{-2}+\cdots+k_{-m}\times 2^{-m+1})
\end{aligned}
$$

只要求出各系数 k_{-1}，k_{-2}，…，k_{-m}，便得到二进制小数。

将上式两边乘 2，得

$$(1.25)_{10}=k_{-1}+(k_{-2}\times 2^{-1}+\cdots+k_{-m}\times 2^{-m+1})$$

根据两个数相等，其整数部分和小数部分必须分别相等的道理，k_{-1} 等于左边的整数，则 k_{-1} 为 1。

等式右边括号内的数仍为小数，因而

$$(0.25)_{10}=\frac{k_{-2}}{2}+\frac{1}{2}(k_{-3}\times 2^{-1}+\cdots+k_{-m}\times 2^{-m+2})$$

再将等式两边乘 2，得

$$(0.5)_{10}=k_{-2}+(k_{-3}\times 2^{-1}+\cdots+k_{-m}\times 2^{-m+2})$$

比较等式两边的整数，又取 k_{-2} 为 0。如此连续乘 2，直到小数部分等于 0，即可求得系数 k_{-1}，k_{-2}，…，k_{-m}。

根据上面讨论的方法，可用下列形式很方便地将十进制小数转换成二进制小数：

$$
\begin{array}{rl}
0.625 & \\
\times)\quad 2 & \\
\hline
\boxed{1}.250 & \text{整数 } 1(k_{-1})\text{ 最高小数位} \\
\times)\quad 2 & \\
\hline
\boxed{0}.500 & \text{整数 } 0(k_{-2})\text{ 次高位小数位} \\
\times)\quad 2 & \\
\hline
\boxed{1}.000 & \text{整数 } 1(k_{-3})\text{ 最低位小数位}
\end{array}
$$

最后得：$(0.625)_{10}=(0.101)_2$。

必须指出：运算时，式中的整数不参加连乘。

在十进制的小数部分转换中，有时连续乘 2 不一定能使小数部分等于 0，这说明该十进制小数不能用有限位二进制小数表示。这时，只要取足够多的位数，使其误差达到所要求的精度就可以了。下面举例说明。

例 1-5　将十进制数 0.18 转换成二进制数，精确到小数点后 4 位。

解：

$$
\begin{array}{r l}
0.18 & \\
\times)\quad 2 & \\
\hline
\boxed{0}.36 & \text{整数}0(k_{-1}) \\
\times)\quad 2 & \\
\hline
\boxed{0}.72 & \text{整数}0(k_{-2}) \\
\times)\quad 2 & \\
\hline
\boxed{1}.44 & \text{整数}1(k_{-3}) \\
\times)\quad 2 & \\
\hline
\boxed{0}.88 & \text{整数}0(k_{-4}) \\
\times)\quad 2 & \\
\hline
\boxed{1}.76 & \text{整数}1(k_{-5})
\end{array}
$$

十进制数0.18连续四次乘2后，其小数部分等于0.88，仍不为0。由于要求精确到小数点后4位，因此将0.88再乘2，小数点后第5位为1，得

$$(0.18)_{10} \approx (0.00101)_2$$

如果一个十进制数既有整数部分又有小数部分，转换时，整数部分采用“除2取余”法，小数部分采用“乘2取整”法，然后再把转换的结果合并起来即可。

1.2.2 二、八、十六进制数的转换

八进制数的基数 $R=8$，而3位二进制数恰好是8个状态，即 $8=2^3$；十六进制数的基数 $R=16$，而4位二进制数恰好是16个状态，即 $16=2^4$。二进制数、八进制数和十六进制数之间具有2的整指数倍关系，因而可直接进行转换。

将二进制数转换为八进制或十六进制的方法是：从小数点开始，分别向左、右按3位一组转换为八进制或按4位一组转换为十六进制，最后不满3位或4位的则需补0。将每组以对应的等值八进制数或十六进制数代替，举例如下。

$$
\begin{array}{lccccccccc}
\text{二进制数：} & \underbrace{010} & \underbrace{101} & \underbrace{111} & . & \underbrace{000} & \underbrace{101} & \underbrace{101} & \underbrace{100} \\
\text{八进制数：} & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow & \downarrow \\
 & 2 & 5 & 7 & . & 0 & 5 & 5 & 4
\end{array}
$$

$$
\begin{array}{lccccccc}
\text{二进制数：} & \underbrace{1010} & \underbrace{1111} & . & \underbrace{0001} & \underbrace{0110} & \underbrace{1100} \\
\text{十六进制数：} & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow \\
 & A & F & . & 1 & 6 & C
\end{array}
$$

最后得：$(257.0554)_8 = (10101111.0001011011)_2 = (AF.16C)_{16}$

若将八进制数或十六进制数转换成二进制数时，可按上述方法的逆过程进行。

1.3 带符号数的代码表示

1.3.1 真值与机器数

人们通常都用符号“+”表示正，用符号“-”表示负。“+”和“-”无非是表示