鲍可进　赵念强　赵不贿 等　编著

数字逻辑电路设计

清华大学出版社

鲍可进　赵念强　赵不贿 等　编著

数字逻辑电路设计

清华大学出版社
北京

内 容 简 介

本书从数字电路的基础知识出发，介绍数制和编码、逻辑代数、门电路、组合逻辑、时序逻辑、硬件描述语言(ABEL，VHDL)、可编程器件(PLD，CPLD，HDPLD，FPGA)、在系统编程技术(ISP)及 EDA 技术的设计思想等内容。最后一章介绍了复杂逻辑电路的设计实例。每章末有小结并附有一定数量的习题与思考题。

本书可作为高等院校计算机、通信、电子信息、自动化等专业的“数字逻辑”课程的教材，也可作为相关技术人员的参考书。

图书在版编目(CIP)数据

数字逻辑电路设计/鲍可进等编著．—北京：清华大学出版社，2004
ISBN 7-302-07878-5

Ⅰ．数… Ⅱ．鲍… Ⅲ．数字电路－逻辑设计－高等学校－教材 Ⅳ．TN790．2

中国版本图书馆 CIP 数据核字(2003)第 124786 号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编**：100084
社 总 机：010-62770175　　**客户服务**：010-62776969
组稿编辑：陈国新
文稿编辑：马幸兆
印 装 者：北京市清华园胶印厂
发 行 者：新华书店总店北京发行所
开　　本：185×260　**印张**：23.75　**字数**：543 千字
版　　次：2004 年 2 月第 1 版　2004 年 2 月第 1 次印刷
书　　号：ISBN 7-302-07878-5/TN·166
印　　数：1～4000
定　　价：36.00 元

前　言　FOREWORD

数字逻辑电路设计是电子信息类专业必修的技术基础课。本书主要讲述数字系统的基础知识及用数字电路进行数字系统的分析与设计的基本理论和方法。

20 世纪 90 年代末以来，信息技术飞速发展。电子器件从传统的小规模集成芯片到中、大规模集成芯片，从复杂可编程器件到高密度可编程器件。设计方法从经典的手工设计到电子设计自动化(EDA)。EDA 使得几乎硬件电子电路的所有设计过程都可以通过计算机来完成，这种方法大大缩短了专用集成电路的设计周期，使得生产厂商的产品能够迅速上市，提高了产品的竞争力。

电子设计自动化技术是 20 世纪 90 年代以后发展起来的，它打破了传统的由固定集成芯片组成数字系统的模式，给数字系统设计带来了革命性的变化，电子信息类专业的学生掌握这个新技术十分必要。因此我们在编写本书时除保留数字逻辑最基本的内容外，还增加了对硬件描述语言 ABEL 及 VHDL 的介绍；在介绍逻辑电路传统设计方法的同时，插入了对硬件描述语言的描述，以便为学生掌握 EDA 技术打下良好的基础。本书还用相当的篇幅介绍了近年来发展迅速的高密度可编程器件，以美国 Lattice 公司在系统可编程芯片(ISP 芯片)为模型讲述了在系统可编程技术，同时介绍了 ALTERA，XINLINX 公司的 FPGA 芯片的基本结构及原理。

为适应教学改革的需求，我们总结了多年教学与科研的经验，对书中的内容做了合理安排。本书共分 8 章，按循序渐进的原则，前面 5 章主要讲述数字电路的基础知识及逻辑电路设计的基本方法，介绍了硬件描述语言的描述方法。硬件描述语言是学习数字逻辑电路课程必需的知识，也是学习可编程器件及 EDA 技术的基础。第 6 章、第 7 章、第 8 章主要讨论了可编程逻辑器件(PLD)、高密度可编程器件(HDPLD)、在系统可编程技术(ISP)和现场可编程门阵列(FPGA)等，重点讲述这些器件的基本结构及利用它们设计电路及系统的基本原理和方法。同时给出一些通俗易懂的设计实例。特别在第 8 章介绍了较复杂的数字系统设计的实例，列出的 VHDL 语言的源程序，可帮助读者更好地掌握数字系统设计的方法，供设计数字系统时参考。受篇幅限制，有关 HDPLD 器件开发软件的使用放到与该书配套的实验指导书中讲解。为方便读者学习，每章附有小结与思考题。整个内容建议安排 50～70 学时，并配以一定学时的实验课及课程设计，以加深对基本理论的理解和对新技术的掌握。

本书由鲍可进担任主编。书中第 1，5，7 章由鲍可进编写；第 3 章由鲍可进、赵念强编写；第 6，8 章由赵念强编写；第 2，4 章由赵不贿编写；袁晓云编写了第 3，5，7 章中部分内容；邹婷婷编写了第 8 章中的部分内容。由于水平有限，加之时间较仓促，书中难免有一些缺点和错误，殷切希望广大读者批评指正。

编　者

2003 年 10 月

目 录 CONTENTS

第1章 CHAPTER

数字系统与编码

在数字系统中，信息的离散元素是以称为信号的物理量来表示的，电压和电流就是最常用的电信号。通常，数字系统的信号只有两个量“有”或“无”，称为二进制信号。具有导通和截止两种工作状态的电子器件能十分可靠地反映两个离散量，且在工程上较容易实现，加上人类的逻辑思维方式也倾向于二值，所以，数字系统常采用二进制信号，即“0”和“1”两个数值。

本章主要围绕数字系统中的二进制信号，讨论数字系统中数的表示方法、数字系统中的编码等基础知识。这些编码不仅在数字系统中使用，在计算机系统中也得到广泛的应用。

1.1 数字系统中的进位制

1.1.1 数制

数制是人们对数量计数的一种统计规律，也就是按进位方式实现计数的一种规则。在日常生活中常用的数制是十进制、十二进制、六十进制等。

对于任何一个数，可以用不同的进位制来表示。先从熟悉的十进制开始，分析各种进位制的特点和表示方法。

十进制有10个数字符号，即0,1,2,3,4,5,6,7,8,9。将若干个这样的符号并列在一起可以表示一个进制数，每位不超过“9”，由低位向高位进位是“逢十进一”，这是十进制的特点。这里要引入两个术语：一个叫“基数”，它表示某种进位制所具有的数字符号的个数，例如十进制的基数为“10”。另一个叫“位权”或“权”，它表示某种进位制的数中不同位置上数字的单位数值，例如十进制数234.56，最左位为百位(2代表200)，权为10^2；第二位为十位(3代表30)，权为10^1；第三位为个位(4代表4)，权为10^0；小数点左边第一位为十分位(5代表5/10)，权为10^{-1}；第二位为百分位(6代表6/100)，权为10^{-2}。

基数和权是进位制的两个要素，根据基数和权的概念，可以将任何一个数表示成多项式的形式。

例如：

$$234.56=2\times10^2+3\times10^1+4\times10^0+5\times10^{-1}+6\times10^{-2}$$

对于一个一般的十进制数N，它可表示成

$$(N)_{10}=(d_{n-1}d_{n-2}\cdots d_1d_0\cdot d_{-1}d_{-2}\cdots d_{-m})_{10} \tag{1.1}$$

或

$$(N)_{10} = d_{n-1}(10)^{n-1} + d_{n-2}(10)^{n-2} + \cdots + d_1(10)^1 + d_0(10)^0 + d_{-1}(10)^{-1} + d_{-2}(10)^{-2} + \cdots + d_{-m}(10)^{-m} = \sum_{i=-m}^{n-1} d_i(10)^i \tag{1.2}$$

式中，n 表示整数部分的位数；m 表示小数部分的位数；10 表示基数，$(10)^i$ 为第 i 位的权；d_i 表示各个数字符号，在十进制中有

$$d_i \in \{0,1,2,3,4,5,6,7,8,9\}$$

通常，把式(1.1)称为并列表示法，把式(1.2)称为多项式表示法或按权展开式。

一般地，对于任意进制数可表示为

$$(N)_R = (r_{n-1}r_{n-2}\cdots r_1\ r_0 \cdot r_{-1}r_{-2}\cdots r_{-m})_R$$

$$= r_{n-1}R^{n-1} + r_{n-2}R^{n-2} + \cdots + r_1R^1 + r_0R^0 + r_{-2}R^{-2} + \cdots + r_{-m}R^{-m} = \sum_{i=-m}^{n-1} r_iR$$

式中，n 表示整数的位数，m 表示小数的位数；R 为基数，在十进制中 R 应写成“10”；r_i 是 R 进制中各个数字符号，即有

$$r_i \in \{0,1,2,\cdots,R-1\}$$

在数字系统中，常用二进制来表示数并进行运算。这时 R 写成“2”，$r_i \in \{0,1\}$。

二进制算术运算十分简单，规则如下：

加法规则 0+0=0，0+1=1+0=1，1+1=10；

乘法规则 0×0=0，0×1=1×0=0，1×1=1。

下面举几个二进制数四则运算的例子，从中领会其运算规则。

例 1.1 两个二进制数相加，采用“逢二进一”的法则。

解：

$$\begin{array}{r} 1101 \\ +)\ 1001 \\ \hline 10110 \end{array}$$

例 1.2 两个二进制数相减，采用“借一当二”的法则。

解：

$$\begin{array}{r} 1101 \\ -)\ 0110 \\ \hline 0111 \end{array}$$

例 1.3 两个二进制数相乘，其方法与十进制乘法运算相似，但采用二进制运算规则。

解：

$$
\begin{array}{r}
1011 \\
\times)\ 1101 \\
\hline
1011 \\
0000 \\
1011 \\
1011 \\
\hline
10001111
\end{array}
$$

例 1.4　两个二进制数相除，其方法与十进制除法运算相似，但采用二进制运算规则。

解：

$$
\begin{array}{r|l}
 & 1010\cdots\text{商} \\
\cline{2-2}
1101 & 10001001 \\
 & 1101 \\
\cline{2-2}
 & 10000 \\
 & 1101 \\
\cline{2-2}
 & 111\cdots\text{余数}
\end{array}
$$

虽然数字系统广泛采用二进制，但当二进制数的位数很多时，书写和阅读很不方便，容易出错。为此，人们通常采用二进制的缩写形式——八进制和十六进制。

八进制的基数 $R=8$，每位可取 8 个不同的数字符号（即 0,1,2,3,4,5,6,7），其进位规则是“逢八进一”。

由于 1 位八进制的 8 个数字符号正好对应于 3 位二进制数的 8 种不同组合，所以，八进制与二进制之间有简单的对应关系：

八进制　0　1　2　3　4　5　6　7

二进制　000 001 010 011 100 101 110 111

这样，八进制与二进制之间数的转换就极为方便。

例如，将二进制数 11010.1101 转换为八进制数，按八-二进制对应关系，有

$$
\begin{array}{cccccc}
011 & 010 & . & 110 & 100 \\
3 & 2 & . & 6 & 4
\end{array}
$$

所以，$(11010.1101)_2=(32.64)_8$。

由上述可知，二进制数转换成八进制数的方法是：以小数为界，将二进制数的整数部分从低位开始，小数部分从高位开始，每 3 位分成一组，头尾不足 3 位的补 0；然后将每组的 3 位二进制数转换为 1 位八进制数。同理，由八进制数转换成二进制数同样很方便。

例如，将八进制数 357.6 转换为二进制数，按八-二进制对应关系，有

$$\begin{array}{ccccccc} 3 & & 5 & & 7 & . & 6 \\ \downarrow & & \downarrow & & \downarrow & . & \downarrow \\ 011 & & 101 & & 111 & . & 110 \end{array}$$

所以，$(357.6)_8=(11101111.11)_2$。

十六进制的基数 $R=16$，每位可取 16 个不同的数字符号（即 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F），其进位规则是“逢十六进一”。

同理，由于 1 位十六进制的 16 个数字符号正好对应于 4 位二进制数的 16 种不同的组合，所以，十六进制与二进制之间有简单的对应关系：

十六进制	0	1	2	3	4	5	6	7
	8	9	A	B	C	D	E	F
二进制	0000	0001	0010	0011	0100	0101	0110	0111
	1000	1001	1010	1011	1100	1101	1110	1111

这样，十六进制与二进制之间数的转换也很方便。

例如，将二进制数 1010110110.110111 转换为十六进制数，按十六-二进制对应关系，有

$$\begin{array}{cccccc} 0010 & 1011 & 0110 & . & 1101 & 1100 \\ 2 & B & 6 & . & D & C \end{array}$$

所以，$(1010110110.110111)_2=(2B6.DC)_{16}$。

例如，将十六进制数 5D.6E 转换为二进制数，按十六-二进制对应关系，有

$$\begin{array}{cccccc} 5 & D & . & 6 & E \\ \downarrow & \downarrow & & \downarrow & \downarrow \\ 0101 & 1101 & . & 0110 & 1110 \end{array}$$

所以，$(5D.6E)_{16}=(1011101.0110111)_2$。

由此可见，采用八进制和十六进制要比用二进制书写简短，易读易记，而且转换也方便。因此，计算机工作者普遍采用八进制或十六进制来书写和表达。

1.1.2　数制转换

在计算机和其他数字系统中普遍采用二进制，采用二进制的数字系统只能处理二进制数或用二进制编码形式表示的其他进位制数。由于人们习惯于使用十进制数，所以在用计算机进行信息处理时，首先必须把十进制数转换成二进制数以便计算机接受；然后进行运算；最后必须把二进制数的运算结果转换成人们习惯的十进制数。

1. 二进制数和十进制数的转换

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：

$$\begin{aligned}(11010.101)_2 &= 1\times 2^4+1\times 2^3+0\times 2^2+1\times 2^1\\ &\quad +0\times 2^0+1\times 2^{-1}+0\times 2^{-2}+1\times 2^{-3}\\ &= 16+8+2+0.5+0.125=(26.626)_{10}\end{aligned}$$

将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。将一个十进制数写成

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

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 $(N)_2=<\text{整数部分}>_2.<\text{小数部分}>_2$

(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，即把十进位制整数除以 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，再取余数 1 或 0 作为二进制数的次低位，依次类推，继续上述过程，直至商为 0，所得余数为最高位。

例如，要将十进制整数 58 转换为二进制整数，就要把它写成如下形式：

$$\begin{aligned}(58)_{10}&=(a_{n-1}a_{n-2}\cdots a_1d_0)_2=a_{n-1}\times 2^{n-1}+a_{n-1}\times 2^{n-2}+\cdots+a_{n-1}\times 2^1+a_{n-1}\times 2^0\\ &=2(a_{n-1}\times 2^{n-2}+a_{n-2}\times 2^{n-3}+\cdots+a_1)+a_0\end{aligned}$$

只要求出等式中的各个系数 $a_{n-1},a_{n-2},\cdots,a_1,a_0$，便得到二进制数。将上式两边除以 2，得 $(29)_{10}=a_{n-1}\times 2^{n-2}+a_{n-2}\times 2^{n-3}+\cdots+a_1+a_0/2$。

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，则取 a_0 为 0，因而得

$$(29)_{10}=2(a_{n-1}\times 2^{n-3}+a_{n-2}\times 2^{n-4}+\cdots+a_2)+a_1$$

将等式两边再除以 2，得

$$(14+1/2)_{10}=a_{n-1}\times 2^{n-3}+a_{n-2}\times 2^{n-2}+\cdots+a_2+a_1/2$$

比较等式两边，等式左边余数为 1，则取 a_1 为 1。依次类推，可得系数 $a_2,a_3,\cdots,a_{n-2},a_{n-1}$。

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数。

2	58	
2	29	余数 0 (a_0)最低位
2	14	余数 1 (a_1)
2	7	余数 0 (a_2)
2	3	余数 1 (a_3)
2	1	余数 1 (a_4)
	0	余数 1 (a_5)最低位

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：

$$(0.625)_{10}=(0.a_{-1}a_{-2}\cdots a_{-m})_2=a_{-1}\times 2^{-1}+a_{-2}\times 2^{-2}+\cdots+a_{-m}\times 2^{-m}$$

$$=\frac{a_{-1}}{2}+\frac{1}{2}(a_{-2}\times 2^{-1}+\cdots+a_{-m}\times 2^{-m-1})$$

只要求出各系数 a_{-1}，a_{-2}，$\cdots$，a_{-m}，便得到二进制小数。将上式两边乘 2，得

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

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，a_{-1} 等于左边的整数，则 a_{-1} 为 1。

等式右边括号内的数仍为小数，因而

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

再将等式两边乘 2，得

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

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

根据上面讨论的方法，可很方便地将十进制小数转换成二进制数，即

```
      0. 6  2  5
×)              2
-----------------
    [1]. 2  5  0   整数 1(a-1)最高小数位
×)              2
-----------------
      0. 5  0  0   整数 0(a-2)
×)              2
-----------------
    [1]. 0  0  0   整数 1(a-3)最低小数位
```

因此，$(0.625)_{10}=(0.101)_2$。必须指出：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。

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：

$$
\begin{array}{rll}
 & 0.\ 1\ \ 8 & \\
\times) & \qquad\ 2 & \\
\hline
 & [0].\ 3\ \ 6 & \text{整数 } 0(a_{-1}) \\
\times) & \qquad\ 2 & \\
\hline
 & [0].\ 7\ \ 2 & \text{整数 } 0(a_{-2}) \\
\times) & \qquad\ 2 & \\
\hline
 & [1].\ 4\ \ 4 & \text{整数 } 1(a_{-3}) \\
\times) & \qquad\ 2 & \\
\hline
 & [0].\ 8\ \ 8 & \text{整数 } 1(a_{-4}) \\
\times) & \qquad\ 2 & \\
\hline
 & [1].\ 7\ \ 6 & \text{整数 } 1(a_{-5})
\end{array}
$$

十进制数 0.18 连续 4 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 再乘一次 2，小数点后第 5 位四舍五入后得$(0.18)_{10}\approx(0.0011)_2$。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。

例 1.6　将$(58.625)_{10}$转换成二进制数。

解：$(58.625)_{10}=(58)_{10}+(0.625)_{10}=(111010)_2+(0.101)_2=(111010.101)_2$

2. 任意两种进制之间的转换

前面介绍的方法并不局限于十进制与二进制之间的转换，可用于任意 α，β 进制之间的转换。因为人们对十进制运算十分熟悉，所以 α 进制→β 进制，一种比较方便的方法是利用十进制作桥梁，先把 α 进制转换为十进制数，这时可以采用按权展开，然后再将十进制数转换为 β 进制数，这时可分为整数（除 β 取余）和小数（乘 β 除整）两部分进行。其示意图如图 1-1 所示。

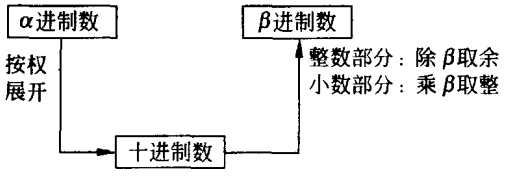

图 1-1　任意进制转换示意图

例 1.7　$(121.02)_4=(?)_3$。

解：先用按权展开的方法将数转换成十进制数：

$(121.02)_4=(1\times4^2+2\times4^1+2\times4^0+2\times4^{-2})_{10}=(16+8+1+0.125)_{10}=(25.125)_{10}$

再将十进制数的整数部分、小数部分分别进行转换，即

```
        3 | 2 5        余数
        3 | 1          1 低位
        3 | 2          2
          0            2 高位
      0.1 2 5
  ×)      3            整数
-----------------
    [0].3 7 5          0 高位
  ×)      3
-----------------
    [1].1 2 5          1
  ×)      3
-----------------
    [0].3 7 5          0 低位
      ⋮
```

所以，$(121.02)_4=(221.010\cdots)_3$。

1.2　数字系统中的编码

1.2.1　带符号数的代码表示

1. 真值与机器数

前面讨论的数都没有考虑符号，一般认为是正数，但在算术运算中总会出现负数。不带符号的数是数的绝对值，在绝对值前加上表示正负的符号就成了带符号数。一个带符号的数由两部分组成：一部分表示数的符号；另一部分表示数的数值。数的符号是一个具有正、负两种值的离散信息，它可以用一位二进制数来表示。习惯上以 0 表示正数；而以 1 表示负数。对于一个 n 位二进制数，如果数的第一位为符号位，则剩下的 $n-1$ 位表示数的数值部分。一般直接用正号“+”和负号“−”来表示符号的二进制数，叫做符号数的真值。数的真值形式是一种原始形式，不能直接用于计算机中。但是，当符号被数值化以后，就可在计算机中使用。计算机中使用的符号数叫做机器数，而原始形式的数称为真值。

例如，二进制正数+0.1011 在机器中的表示如图 1-2(a)所示；二进制负数−0.1011 在机器中的表示如图 1-2(b)所示。

由前面介绍的二进制数的加、减、乘、除 4 种运算可知，乘法运算实际上是做移位加法运算，而除法运算是做移位减法运算。这就是说，在机器中需要做加、减两种运算。但做