21世纪高职高专系列教材

SHUZI LUOJI DIANLU

数字逻辑电路

陈 松 主编

东 南 大 学 出 版 社

21 世纪高职高专系列教材

数字逻辑电路

陈 松 主编

东南大学出版社
·南 京·

内 容 提 要

本书系统地介绍了数字逻辑电路的基本理论、数字逻辑电路的分析和设计方法、数字集成电路的功能和应用，以及可编程逻辑器件的构成原理和设计方法，并介绍了仿真软件在数字逻辑电路分析和设计中的应用。

每章末附有习题，书后附有实验及部分集成电路的资料。

本书可作为高等职业学校和中等职业学校电子技术、电子与信息技术、通信技术、自动控制和机电类等专业的专业基础课教材，也可供从事电子、信息技术的有关人员参考。

图书在版编目(CIP)数据

数字逻辑电路/陈松主编. 一南京:东南大学出版社,2002.7

ISBN 7-81050-977-2

Ⅰ.数... Ⅱ.陈... Ⅲ.数字电路:逻辑电路—高等学校:技术学校—教材 Ⅳ.TN79

中国版本图书馆CIP数据核字(2002)第046911号

东南大学出版社出版发行
(南京四牌楼2号 邮编210096)
出版人:宋增民
江苏省新华书店经销 江苏武进第三印刷厂印刷
开本:787mm×1 092mm 1/16 印张:13.25 字数:338千字
2002年7月第1版 2002年7月第1次印刷
印数:1-4000册 定价:21.00元

(凡因印装质量问题,可直接向发行科调换,电话:025-3792327)

前　言

本书是根据全国电子与信息技术类高等职业学校教材建设会议精神和电子与信息技术类专业关于数字逻辑电路教材的编写要求编写的，是高等职业学校电子技术、电子与信息技术、通信技术、自动控制和机电类等专业的专业基础课教材。

本书包括数字逻辑电路的理论部分和实验部分内容。书后附有一些常用的集成电路资料，让学生了解手册的内容和帮助理解各种器件的工作特点。

在内容安排上，注意贯彻从实际出发，注重器件外部特性的介绍和应用。在教材中将最新的计算机软件介绍给学生，帮助学生对所学知识的理解，并了解最新的设计概念。

全书包括数字电路基础、逻辑门电路、组合逻辑电路、组合逻辑电路的 PLD 实现、触发器、时序电路、脉冲信号的产生和变换、时序电路的 PLD 实现、模数转换与数模转换电路及实验，共 10 章。书中带星号(*)的内容作为选讲内容。

本书第 3 章和第 10 章由南京信息职业技术学院(原南京无线电工业学校)姜红编写，其余内容由南京信息职业技术学院陈松编写。全书由陈松统稿。

由于编者水平有限，书中难免存在不当之处，望广大读者批评指正。

编者电子邮件地址：njchensong@163.com

编　者

2002 年 6 月于南京

目　录

1 数字电路基础

【内容提要】

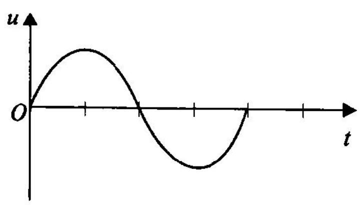
本章主要介绍:模拟信号和数字信号及两种信号之间的区别;逻辑代数的基本概念、基本公式及法则;逻辑函数的几种表示方法,逻辑函数的化简方法。

1.1 模拟信号和数字信号

1.1.1 模拟信号

在《模拟电子线路》教材中讨论的信号主要为模拟信号,这种信号的共同特点是:在时间上连续,并且幅度也是连续变化的,这些信号中有的可以用数学表达式来表达,例如交流电源的50 Hz 正弦信号、正弦信号发生器的输出信号、调幅信号等;但并不是所有的模拟信号都可以用数学表达式加以表示,例如人的讲话声音信号、电视中的视频信号、医院中检查身体的心电信号和脑电信号、人们四周环境温度的变化等自然界的信号,这种信号是随机的,是无法用数学表达式表示的。

图 1.1(a) 为周期性正弦信号的波形,图 1.1(b) 为人们讲话的音频信号波形。



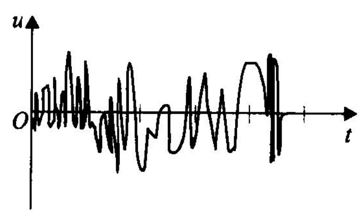


图 1.1 模拟信号的波形

在模拟电路中主要是对模拟信号进行处理,例如进行电压放大、电流放大、功率放大等,其讨论的内容是如何借助于晶体管电路对信号按照要求实现无失真的处理。

1.1.2 数字信号

模拟信号的幅度是连续变化的,在进行存储、传输时容易造成失真,并且无法借助于计算机进行分析处理;数字信号正好可以很好地克服这些问题,这是由于数字信号的状态是有限的。最常见的数字信号为二值数字信号,即只有 0 和 1 两个状态。

1) 数字逻辑和逻辑电平

数字信号在时间和幅度上都是离散的,常用数字0和1表示。这里的0和1不是十进制数中的数字,而是用 0 和 1 这两个数字来表示逻辑的两种状态,因而将“逻辑” 称为数字逻辑。

数字逻辑是从客观现实的许多事件描述抽象出来的。这两个逻辑状态可以对应于:好与

坏、真与假、高与低、开与关等。在电路中可以利用器件的开关特性来实现，由此产生了两种离散的电压状态，常将这两种电压状态称为逻辑电平。

逻辑电平不是一个固定的电压，而是一个相对的状态表示，例如在电源电压为5 V的电路中，将5 V定义为逻辑1，将0 V定义为逻辑0；在一个电源电压为3.3 V的电路中，将3.3 V定义为逻辑1，将0 V定义为逻辑0。5 V和3.3 V都可描述为高电平(H)，而将0 V描述为低电平(L)。表1.1列出了电压大小与逻辑状态及逻辑电平的对应关系。

表1.1　电压大小与逻辑状态及逻辑电平的对应关系

电压/V	逻辑	逻辑电平
5(3.3)	1	H
0	0	L

2）数字信号的波形

图1.2所示为数字信号的波形。图1.2(a)为幅度与时间的关系波形，从图中可见信号的幅度仅0 V和5 V两种电平；图1.2(b)为逻辑状态与时间的关系波形，通常的数字信号波形都表示为逻辑状态与时间的波形。

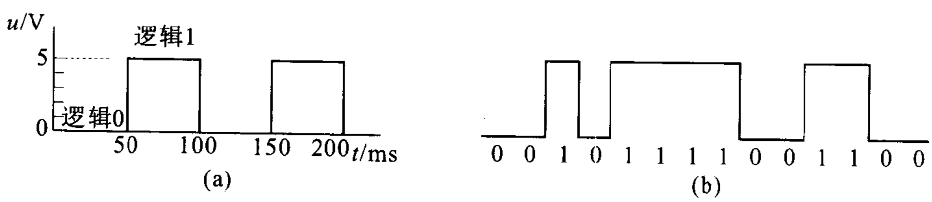


图1.2　数字信号的波形

数字信号的波形根据时间上是否重复可分为周期性数字信号和非周期性数字信号。

在实际的数字系统中，数字信号不可能做到立即上升或下降，而是需要经过一段时间。通常，将从低向高变化的过渡时间定义为上升时间t_r，从高向低变化的过渡时间定义为下降时间t_f。图1.2(b)所示波形为理想的数字信号波形，其t_r和t_f都为0。图1.3所示波形为实际的数字信号波形，其上升时间和下降时间都有一个非零的值与之对应。很显然，上升时间和下降时间越小越好。

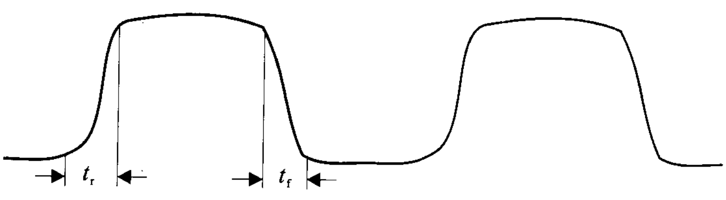


图1.3　实际的数字信号波形

在数字信号参数中，上升时间是指波形从幅值的10%上升到90%所经历的时间，下降时间是从幅度的90%下降到10%所经历的时间。一般数字电路的上升时间及下降时间在几纳秒至几十纳秒之间。

3）模拟信号的数字表示

前面提到，由于模拟信号的幅度是连续变化的，在存储、传输和分析时带来不便，而数字信号可以很好地克服这些不足，因此现在很多电子设备先将模拟信号转变为数字信号进行存储、分析和传输，例如 CD 音乐、数字电视、IP 电话等都是利用了这个特点。

从图 1.2 所示的数字信号的波形上看，数字信号仅两种状态，这是无法用来表示模拟信号的无限状态的。在模拟信号转变为数字信号时采用了编码方法，用数字 0 和 1 来表示模拟信号幅度的大小。这里的 0 和 1 是指数字 0 和 1，图 1.4 显示模拟信号转换为数字信号的原理。

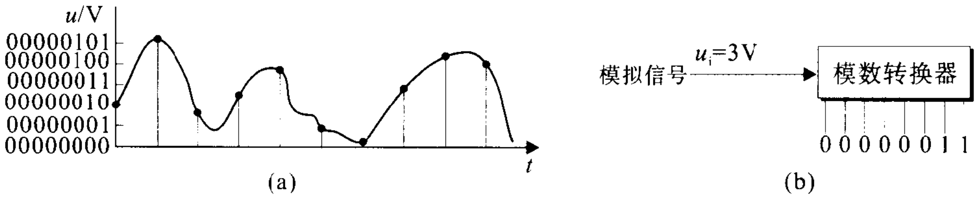


图 1.4　模拟信号的数字表示

图 1.4(a) 为模拟信号转换为数字信号的原理曲线，可以看出，它是将一些点的电压值用一串 0 和 1 来表示，这种不同的 0,1 组合代表不同的幅度值；图 1.4(b) 表示当输入电压为 3 V 时，输出的二进制数码为 00000011，相当于十进制数的 3。从图 1.4(a) 所示的转换过程可以看出转换可能存在误差，但如果二进制数的位数足够多，误差可以忽略不计，就可以用二进制数码复制原来的模拟信号，在需要的时候通过相反的方法将其还原为模拟信号。

1.2　数制和码制

1.2.1　十进制数

日常生活中，当人们记录事件的多少时通常用十进制数来表示。所以被称为十进制数，是因为采用了 10 个有序数码 0,1,2,3,4,5,6,7,8,9 代表 1 位十进制数字的 10 种不同状态，并遵循“逢十进一，借一当十”的进借位规则。通常将用来进行计数的数码个数称为基数，即十进制数的基数为 10。

十进制数中数码所在的位置不同，它所表示的值就不同。例如：

$$6\,834 = 6 \times 1\,000 + 8 \times 100 + 3 \times 10 + 4 \times 1$$

式中，每个数码分别有一个系数 1 000,100,10,1，这些系数称为权或位权。

十进制数 N 可表示为：

$$(N)_{10} = a_{n-1} \times 10^{n-1} + a_{n-2} \times 10^{n-2} + \cdots + a_1 \times 10 + a_0 \times 10^0 + a_{-1} \times 10^{-1} + a_{-2} \times 10^{-2} + \cdots + a_{-m} \times 10^{-m} = \sum_{i=-m}^{n-1} a_i \times 10^i$$

式中：a_i 为0 ~ 9 中任一数码，是基数10 的第 i 次幂的系数；10 为进制的基数；10^i 为第 i 位的权；n 和 m 为正整数，n 为整数部分的位数，m 为小数部分的位数。

例如：

$$(32.45)_{10} = 3 \times 10^1 + 2 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2}$$

任一进制 R 的数可表示为：

$$(N)_R = \sum_{i=-m}^{n-1} a_i \times R^i$$

式中：R 为基数；R^i 为第 i 位的权；a_i 为第 i 位对应的数码；n 和 m 为正整数，n 为整数部分的位数，m 为小数部分的位数。

在数字系统中，除了常用的十进制外，还有二进制、八进制、十六进制等。

1.2.2 二进制数

与十进制相似，二进制也遵循两个规则：一是由两个不同的数码0，1来表示；二是遵循“逢二进一，借一当二”的进借位规则。任一个二进制数 N 可以表示为：

$$(N)_2 = \sum_{i=-m}^{n-1} a_i \times 2^i$$

式中：a_i 为第 i 位的数码，为0，1 两数码中的一个。

于是

$$(1011.01)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$$

二进制数由于只有0，1两个数码，所以其运算规则比较简单。下面是二进制数加法和乘法的运算规则：

加　法	乘　法
$0+0=0$	$0\times0=0$
$1+0=0+1=1$	$1\times0=0\times1=0$
$1+1=10$	$1\times1=1$

二进制数和十进制数在书写时为了不至于引起混淆，常在十进制数后面加一个下标10，二进制数后面加一个下标2，以便区分，有时也用B(Binary)和D(Decimal)来代替2和10两个下标。例如，十进制数32.7可以写为 32.7_D 或32.7D；二进制数1001.11可以写为 1001.11_B 或1001.11B。

1.2.3 进制之间的相互转换

二进制数具有易于表示、方便运算和易于实现的优点，但当位数较多时，二进制数就不便于读写；十进制数虽然是生活中最常用、最习惯的一种进制数，但由于每一位数所使用的数码较多，计算机在实现运算、处理时所需设备较复杂而且不易实现。因此，常常需要把十进制数转换成二进制数进行运算、处理，再将运算及处理的结果逆变换为十进制数进行读取及记录。因此，在使用中经常要进行不同进制数之间的转换。

1）二进制数转换为十进制数

二进制数转换为十进制数时，只要将二进制数按公式 $(N)_2 = \sum_{i=-m}^{n-1} a_i \times 2^i$ 展开，然后把所有各项按十进制数相加，就可以得到等值的十进制数。

【例1.1】将二进制数 1011.011_B 转换为十进制数。

$$\begin{aligned}1011.011_B &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} \\ &= 8 + 2 + 1 + 0.25 + 0.125 = 11.375_D\end{aligned}$$

2）十进制数转换为二进制数

十进制数转换为二进制数时，可将其分为整数和小数两个部分分别进行转换，最后将结果合并为目标数。

（1）整数部分的转换

整数部分的转换采用除基取余法。所谓除基取余法，是指用目标数的基数去除十进制数的整数部分，第一次除以所得余数为目标数的最低位，把得到的商再除以该基数，所得余数为目标数的次低位，依次类推，继续上述过程，直至商为 0 止，此时所得余数为目标数的最高位。

【例 1.2】将 53_D 转换为二进制数。

2	53		
2	26	……1	低位
2	13	……0	
2	6	……1	
2	3	……0	
2	1	……1	
	0	……1	高位

$$53_D = 110101_B$$

（2）小数部分的转换

小数部分的转换是采用乘基取整法。所谓乘基取整法，是指用该小数乘以目标数的基数，第一次乘得结果的整数部分为目标小数部分的最高位，其小数部分再乘以目标数的基数，所得结果的整数部分为目标数的次高位，依次类推，继续上述过程，直至小数部分为 0 或达到要求的精度为止。

【例 1.3】将十进制数 0.625_D 转换为二进制数（转换结果取 4 位小数）。

$$\begin{array}{r l l}
 & 0.625 & \\
\times & 2 & \\
\hline
 & \boxed{1}.250 & \cdots\cdots 1 \quad \text{最高位} \\
\times & 2 & \\
\hline
 & \boxed{0}.500 & \cdots\cdots 0 \quad \text{次高位} \\
\times & 2 & \\
\hline
 & \boxed{1}.000 & \cdots\cdots 1 \quad \text{最低位}
\end{array}$$

$$0.625_D = 0.101_B$$

【例 1.4】将十进制数 0.345_D 转换为二进制数（转换结果取 4 位小数）。

$$\begin{array}{r l l}
 & 0.345 & \\
\times & 2 & \\
\hline
 & \boxed{0}.690 & \cdots\cdots 0 \quad \text{高位} \\
\times & 2 & \\
\hline
 & \boxed{1}.380 & \cdots\cdots 1 \quad \text{次高位} \\
\times & 2 & \\
\hline
 & \boxed{0}.760 & \cdots\cdots 0 \\
\times & 2 & \\
\hline
 & \boxed{1}.520 & \cdots\cdots 1 \quad \text{低位}
\end{array}$$

从上述过程可以看出，再往下计算，其结果不可能使小数部分为0，这时就取决于要求达到的精度。本题要求的精度为4位小数，故计算结果为：

$$0.345_D \approx 0.0101_B$$

这里需要指出的是：任何非十进制数（包括整数部分和小数部分）都可以完整地转换为十进制数，而十进制数有时不能完全转换为非十进制数，这时只能根据精度要求，求到一定位数就近似表示。

3）利用工具软件进行进制转换

前面介绍了手工进行进制转换的方法。在实际使用时，可以借助于计算机软件来实现进制的相互转换，最简单的方法是通过 Win 98/NT/2000 操作系统软件中附带的计算器软件进行进制的转换。

在 Win 98/NT/2000 桌面上单击“开始”按钮，先后打开“程序”、“附件”、“计算器”，在计算器界面中“查看”菜单下的选择“科学型”就可以出现图1.5所示的科学型计算器的界面，选择源进制后输入数据，再单击转换的目标进制，这时计算器界面中将显示出转换的结果。

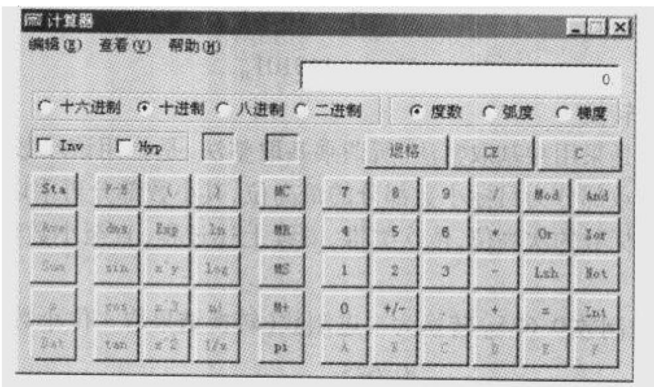


图1.5 Win 98 中计算器界面

1.2.4 BCD编码

在数字系统中，各种数据要转换为二进制代码才能进行处理，而人们习惯于使用十进制数，所以在数字系统的输入／输出中仍采用十进制数，这样就产生了用4位二进制数表示1位十进制数的编码方法。这种用于表示十进制数的二进制代码称为二－十进制代码（Binary Coded Decimal），简称为 BCD 码。它具有二进制数的形式，可以满足数字系统的要求，又具有十进制数的特点（只有10种数码状态有效）。在某些情况下，计算机也使用这种形式的数直接进行运算。

BCD 码有多种形式，下面介绍几种常见的 BCD 码形式。

1）8421 BCD 码

8421 BCD 码是 BCD 编码中使用最多的一种编码形式，是一种有权码，其各位（从最高位开始至最低位）的权分别是8，4，2，1。如果把每一个代码看成一个4位的二进制数，这个代码的数值恰好等于它所代表的十进制数的大小。

【例1.5】写出 563.97_D 对应的 8421 BCD 码。由于：

$$\begin{array}{ccccccc} (& 5 & 6 & 3 & . & 9 & 7\ )_D \\ (& 0101 & 0110 & 0011 & . & 1001 & 0111\ )_{8421\ BCD} \end{array}$$

因此，

$$563.97_D = (0101\ 0110\ 0011\ .\ 1001\ 0111)_{8421\ BCD}$$

【例 1.6】写出 $01101001.01011000_{8421\ BCD}$ 对应的十进制数。由于

$$\begin{array}{cccccc} (\ \underline{0110} & \underline{1001} & \underline{.} & \underline{0101} & \underline{1000} &)_{8421\ BCD} \\ (\ \ 6 & 9 & . & 5 & 8 &)_D \end{array}$$

因此，

$$(01101001.01011000)_{8421\ BCD} = 69.58_D$$

注意:8421 BCD 码的有效码为 0000 ~ 1001,不可能出现 1010 ~ 1111 之间的编码。

2）2421 码

2421 码也是一种有权码,该码也是用 4 位二进制代码来表示 1 位十进制数,自左至右各位的权值分别为 2,4,2,1。

2421 码的位权展开式可写为：

$$a_4a_3a_2a_1 = 2 \times a_4 + 4 \times a_3 + 2 \times a_2 + 1 \times a_1$$

从表 1.2 中不难看出,在 2421 码中,十进制数 0 和 9,1 和 8,2 和 7,3 和 6,4 和 5 的对应位码的一个为 0 时,另一个就为 1,即互为反码。

表 1.2　常见 BCD 编码

十进制数	8421 BCD 码	2421 码	余 3 码
0	0000	0000	0011
1	0001	0001	0100
2	0010	0010	0101
3	0011	0011	0110
4	0100	0100	0111
5	0101	1011	1000
6	0110	1100	1001
7	0111	1101	1010
8	1000	1110	1011
9	1001	1111	1100
10	0001,0000	0001,0000	0100,0011

3）余 3 码

余 3 码也是用 4 位二进制数代表 1 位十进制数,由于它是在 8421 BCD 码上加 0011 得到的,所以这种编码称为余 3 码。

从表 1.2 中不难看出,其对应的十进制数 0 和 9,1 和 8,2 和 7,3 和 6,4 和 5 的码也是互为反码。它是一种无权码。

1.2.5　格雷码

格雷码是一种无权码,其特点是任意两个相邻的码组之间只有 1 位数不同。表 1.3 中给出了 4 位格雷码的编码情况。从表中可以看出,最小数 0 与最大数 15 之间也只有 1 位数不同,可将