国家电工电子教学基地、国家电工电子实验教学示范中心系列教材

北京市精品课程教材

电子技术
（数字部分）

Dianzi Jishu Shuzi Bufen

⊙主 编 任 希

北京邮电大学出版社
www.buptpress.com

国家电工电子教学基地、国家电工电子实验教学
北京市精品课程教材

电子技术

（数字部分）

主编 任希

北京邮电大学出版社

内容提要

本书是根据近年来数字电子技术的发展和编者丰富的教学实践，针对数字电子技术课程教学基本要求和学习的特点，编写而成的。

本书的主要内容包括：数字逻辑基础，主要介绍数制、编码、基本逻辑关系、逻辑代数等；逻辑门电路，主要介绍典型逻辑门电路及其主要参数；组合逻辑电路，主要介绍组合逻辑电路的分析和设计、典型集成组合逻辑电路的特点及应用；触发器，主要介绍典型触发器的逻辑功能和特点；时序逻辑电路，主要介绍触发器组成的时序逻辑电路的分析和简单的设计、集成计数器和寄存器的逻辑功能和典型应用；半导体存储器和可编程逻辑器件，主要介绍存储器和可编程逻辑器件的逻辑功能；脉冲波形的变换和产生，主要介绍数字电路使用的脉冲波形变换和产生的一些方法；数模与模数转换器，主要介绍数模与模数转换器的基本工作原理和典型集成数模与模数转换器；数字电路综合设计，主要介绍数字电路设计的基本要求和方法。

本书适合作为高等院校电气信息类各专业和部分非电类专业的教科书，也可作为工程技术人员的参考书。

图书在版编目(CIP)数据

电子技术.数字部分/任希主编.--北京:北京邮电大学出版社,2013.1

ISBN 978-7-5635-3326-8

Ⅰ.①电… Ⅱ.①任… Ⅲ.①数字电路—电子技术 Ⅳ.①TN

中国版本图书馆 CIP 数据核字(2012)第 283490 号

书　　名	电子技术(数字部分)
主　　编	任　希
策 划 人	唐咸荣　韩　霞
责任编辑	韩　霞
出版发行	北京邮电大学出版社
社　　址	北京市海淀区西土城路 10 号(100876)
电话传真	010-82333010　62282185(发行部)　010-82333009　62283578(传真)
网　　址	www.buptpress3.com
电子信箱	ctrd@buptpress.com
经　　销	各地新华书店
印　　刷	北京泽宇印刷有限公司
开　　本	787 mm×960 mm　1/16
印　　张	17
字　　数	370 千字
版　　次	2013 年 1 月第 1 版　2013 年 1 月第 1 次印刷

ISBN 978-7-5635-3326-8　　　**定价：**36.00 元

前　言

数字电子技术是电子技术中的一部分，是通信、电子、计算机和自动控制等专业的基础课之一。目前数字电子技术正处在日新月异的变化之中，特别是超大规模集成电路的使用，可编程逻辑器件的使用和各种数字电子技术仿真软件的出现，使数字电路从设计到实现变得更加容易和灵活。因而数字电子技术似乎出现了传统和现代的矛盾。传统就是数字电子技术几十年传承下来的基本理论、基本器件和基本的分析、设计方法。现代就是数字电子技术中出现的新器件、新技术和数字电子技术仿真软件的使用。

然而，万变不离其宗，我们可以发现利用数字电子技术中出现的新器件、新技术和仿真软件去分析和设计数字电路依然是根据传统的基本理论、基本器件和基本的分析、设计方法。本书主要介绍数字电子技术的基本理论、基本器件和基本的分析、设计方法，同时也适当地介绍新器件和新技术。根据多年的教学经验和实践经验，尽量使本书做到语言简练，内容具有系统性和适用性，便于阅读。为了使本书更具有实用性，书中还增加了一些应用方面的内容。

本书的主要内容包括：数字逻辑基础，主要介绍数制、编码、基本逻辑关系、逻辑代数等；逻辑门电路，主要介绍典型逻辑门电路及其主要参数；组合逻辑电路，主要介绍组合逻辑电路的分析和设计、典型集成组合逻辑电路的特点及应用；触发器，主要介绍典型触发器的逻辑功能和特点；时序逻辑电路，主要介绍触发器组成的时序逻辑电路的分析和简单的设计、集成计数器和寄存器的逻辑功能和典型应用；半导体存储器和可编程逻辑器件，主要介绍存储器和可编程逻辑器件的逻辑功能；脉冲波形的变换和产生，主要介绍数字电路使用的脉冲波形变换和产生的一些方法；数模与模数转换器，主要介绍数模与模数转换器的基本工作原理和典型集成数模与模数转换器；数字电路综合设计，主要介绍数字电路设计的基本要求和方法。增加数字电路综合设计这部分内容的目的是，让读者可以全面了解数字电路从理论到实际电路的产生过程。

本书由任希担任主编，参加编写的人员还有张海涛、刘宗良。

由于编者水平有限，书中难免有不妥和错误之处，恳请读者批评指正。

编　者

2012 年 9 月

目　　录

第 1 章 数字逻辑基础

1.1 数　　制

数制就是人们计数的方式，在日常生活中人们习惯用十进制，在数字电路中，常采用二进制，有时也采用十六进制或八进制。

1.1.1 十进制数

十进制数是由 0 ～ 9 十个不同的数码组成的，所以计数的基数是 10，超过 9 的数必须用多位数表示，其计数规律是“逢十进一”。这样，每一数码处于不同的位置时，它所代表的数值是不同的，从左到右表示由高到低。例如，十进制数 369.12 可以表示为

$$369.12 = 3 \times 10^2 + 6 \times 10^1 + 9 \times 10^0 + 1 \times 10^{-1} + 2 \times 10^{-2}$$

上式等号的右边为该数的按权展开，10^2、10^1、10^0、10^{-1} 和 10^{-2} 分别为百位、十位、个位、十分位和百分位的权，位数越高权值越大。

任意十进制数可表示为

$$(N)_{\mathrm{D}} = (K_{n-1}K_{n-2}\cdots K_1K_0K_{-1}\cdots K_{-m})_{\mathrm{D}} = \sum_{i=-m}^{n-1} K_i 10^i \tag{1.1.1}$$

式中，下标 D 表示十进制数；K_i 代表第 i 位的数码(0 ～ 9)；10^i 表示第 i 位的权值；m 和 n 为正整数，分别表示十进制数的整数和小数部分的位数。

若以 R 取代式(1.1.1) 中的 10，即可得到任意进制数的表达式：

$$(N)_R = \sum_{i=-m}^{n-1} K_i R^i \tag{1.1.2}$$

式中，下标 R 表示任意进制数；K_i 代表第 i 位的数码；R^i 表示第 i 位的权值；m 和 n 为正整数，分别表示任意进制数的整数和小数部分的位数。任意进制数的计数规律是“逢 R 进一”。

1.1.2　二进制数

二进制数是由0和1两个数码组成的，所以计数的基数数是2，超过1的数必须用多位数表示，其计数规律是“逢二进一”，如1011.11。二进制数的一般表达式为

$$(N)_B = \sum_{i=-m}^{n-1} K_i 2^i \tag{1.1.3}$$

式中，下标B表示二进制数；K_i 代表第 i 位的数码(0～1)；2^i 表示第 i 位的权值；m 和 n 为正整数，分别表示二进制数的整数和小数部分的位数。则二进制数1011.11可表示为

$$(1011.11)_B = 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 1\times 2^0 + 1\times 2^{-1} + 1\times 2^{-2}$$

在数字电路中通常使用二进制数，这是因为二进制数只有0和1两个数字符(或两个状态)，因此很容易用电路器件的开关状态实现。比如二极管的导通和截止状态，三极管的饱和与截止状态，继电器的通和断，电平的高和低。这种表示方法简单方便，处理、存储和传输数据也十分可靠。

1.1.3　八进制数与十六进制数

由于二进制数比十进制数位数多，不便于书写和记忆，因此常用八进制数或十六进制数来表示二进制数。

1. 八进制数

八进制数是用0～7八个不同数码组成的，其计数规律是“逢八进一”，如5703.6。八进制数的一般表达式为

$$(N)_O = \sum_{i=-m}^{n-1} K_i 8^i \tag{1.1.4}$$

式中，下标O表示八进制数；K_i 代表第 i 位的数码(0～7)；8^i 表示第 i 位的权值；m 和 n 为正整数，分别表示八进制数的整数和小数部分的位数。则八进制数5703.6可表示为

$$(5703.6)_O = 5\times 8^3 + 7\times 8^2 + 0\times 8^1 + 3\times 8^0 + 6\times 8^{-1}$$

2. 十六进制数

十六进制数是用0～9和A(10)、B(11)、C(12)、D(13)、E(14)、F(15)十六个不同数码组成的，其计数规律是“逢十六进一”，如FB8.A。十六进制数的一般表达式为

$$(N)_H = \sum_{i=-m}^{n-1} K_i 16^i \tag{1.1.5}$$

式中，下标H表示十六进制数；K_i 代表第 i 位的数码(0～9和A、B、C、D、E、F)；16^i 表示第 i 位的权值；m 和 n 为正整数，分别表示十六进制数的整数和小数部分的位数。十六进制数FB8.A

可表示为

$$(FB8.A)_H = F \times 16^2 + B \times 16^1 + 8 \times 16^0 + A \times 16^{-1}$$

由于目前在微型计算机中普遍采用8位、16位和32位二进制并行运算，而8位、16位和32位的二进制数可以用2位、4位和8位的十六进制数表示，因而用十六进制符号书写程序十分简便。

表1.1.1是十进制数0～15与等值二进制、八进制、十六进制数的对照表。

表1.1.1　不同进制数的对照表

十进制(Decimal)	二进制(Binary)	八进制(Octal)	十六进制(Hexadecimal)
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	7
08	1000	10	8
09	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

1.1.4　不同数制间的转换

一、二进制数、八进制数和十六进制数转换成十进制数

1. 二进制数转换成十进制数

利用二进制数的一般表达式 $(N)_B = \sum_{i=-m}^{n-1} K_i 2^i$ 即可将二进制数转换成十进制数。例如

$$(1011.11)_B = 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 + 1\times2^{-1} + 1\times2^{-2} = (11.75)_D$$

2. 八进制数转换成十进制数

利用八进制数的一般表达式$(N)_O = \sum\limits_{i=-m}^{n-1} K_i 8^i$ 即可将八进制数转换成十进制数。例如

$$(5703.6)_O = 5\times8^3 + 7\times8^2 + 0\times8^1 + 3\times8^0 + 6\times8^{-1} = (3011.75)_D$$

3. 十六进制数转换成十进制数

利用十六进制数的一般表达式$(N)_H = \sum\limits_{i=-m}^{n-1} K_i 16^i$ 即可将十六进制数转换成十进制数。例如

$$(\mathrm{FB8.A})_H = \mathrm{F}\times16^2 + \mathrm{B}\times16^1 + 8\times16^0 + \mathrm{A}\times16^{-1} = (4024.625)_D$$

二、十进制数转换成二进制数

1. 十进制整数转换成二进制数

十进制整数转换成二进制数可以采用“除 2 取余”法。因为任意十进制整数 N 可以写成

$$(N)_D = K_n\times2^n + K_{n-1}\times2^{n-1} + K_{n-2}\times2^{n-2} + \cdots + K_1\times2^1 + K_0\times2^0$$

式中,$K_n, K_{n-1}, K_{n-2}, \cdots, K_1, K_0$ 是等值的二进制数各位数码。所以将等式两边分别除以 2,可得

$$(N)_D \div 2 = K_n\times2^{n-1} + K_{n-1}\times2^{n-2} + K_{n-2}\times2^{n-3} + \cdots + K_1\times2^0 + K_0$$

余数 K_0 即为二进制数的数码。以此类推,反复将每次得到的商再除以 2,直到商为 0,就可以根据余数得到二进制数的每一位数码。

例 1.1.1　将十进制数 173 转换成二进制数。

解:根据“除 2 取余”法

2	173	………………………… 余数=1=K_0
2	86	………………………… 余数=0=K_1
2	43	………………………… 余数=1=K_2
2	21	………………………… 余数=1=K_3
2	10	………………………… 余数=0=K_4
2	5	………………………… 余数=1=K_5
2	2	………………………… 余数=0=K_6
2	1	………………………… 余数=1=K_7
	0	

所以,$(173)_D = (K_7K_6K_5K_4K_3K_2K_1K_0)_B = (10101101)_B$。

2. 十进制小数转换成二进制数

十进制小数转换成二进制数可以采用“乘 2 取整”法。因为任意十进制小数 N 可以写成

$$(N)_D = K_{-1}\times2^{-1} + K_{-2}\times2^{-2} + \cdots + K_{-(n-1)}\times2^{-(n-1)} + K_{-n}\times2^{-n}$$

式中,$K_{-1}, K_{-2}, K_{n-2}, \cdots, K_{-(n-1)}, K_{-n}$ 是等值的二进制数各位数码。所以将等式两边分别乘以

2,可得

$$(N)_D \times 2 = K_{-1} + K_{-2} \times 2^{-1} + \cdots + K_{-(n-1)} \times 2^{-(n-2)} + K_{-n} \times 2^{-(n-1)}$$

整数 K_{-1} 即为二进制数的数码。以此类推,反复将每次得到的小数再乘以 2,直到小数部分为 0,或者小数部分虽然不为 0,但二进制小数的位数或精度已经达到要求,就可以得到二进制小数的每一位数码。

例 1.1.2　将十进制小数 0.812 5 转换成二进制数。

解:根据"乘 2 取整"法

$$\begin{array}{rl}
0.8125 & \\
\times\qquad 2 & \\
\hline
1.6250 & \cdots\cdots\cdots\cdots\cdots\cdots \text{整数部分}=1=K_{-1} \\
0.6250 & \\
\times\qquad 2 & \\
\hline
1.2500 & \cdots\cdots\cdots\cdots\cdots\cdots \text{整数部分}=1=K_{-2} \\
0.2500 & \\
\times\qquad 2 & \\
\hline
0.5000 & \cdots\cdots\cdots\cdots\cdots\cdots \text{整数部分}=0=K_{-3} \\
0.5000 & \\
\times\qquad 2 & \\
\hline
1.0000 & \cdots\cdots\cdots\cdots\cdots\cdots \text{整数部分}=1=K_{-4}
\end{array}$$

所以,$(0.8125)_D = (0.K_{-1}K_{-2}K_{-3}K_{-4})_B = (0.1101)_B$。

若十进制小数用"乘 2 取整"法不能使最后的小数部分为 0,或者所得到的二进制数的位数过长,可以根据二进制小数的位数要求或精度,确定二进制小数的最后位数。

由上述分析可知,一个十进制数转换成二进制数,可以对其整数部分和小数部分分别转换,然后再将转换结果合并,就可得到二进制数。

3. 二进制数与十六进制数相互转换

由于 4 位二进制数恰好有 16 个状态,而把这 4 位二进制数看作一个整体时,它的进位输出又正好是逢十六进一,即 4 位二进制数可以看成 1 位十六进制数。所以只要从低位到高位将整数部分每 4 位二进制数分为一组并代之以等值的十六进制数,同时从高位到低位将小数部分每 4 位数分为一组并代之以等值的十六进制数。若不足 4 位时,可在整数的最高位前和小数的最低位后补 0 构成 4 位。即可得到十六进制数。

例 1.1.3　将二进制数 111110.101011 转换成十六进制数。

解:$(111110.101011)_B = (0011\ 1110.1010\ 1100)_B = (3E.AC)_H$

若将十六进制数转换成二进制数,只需将十六进制数的每一位用等值的 4 位二进制数代替即可。

例 1.1.4　将十六进制数 FB8.A 转换成二进制数。

解:$(FB8.A)_H = (1111\ 1011\ 1000.1010)_B$

4. 二进制数与八进制数相互转换

将二进制数转换成八进制数，可将二进制数分为3位一组，再将每组的3位二进制数转换成等值的1位八进制数即可。

例 1.1.5　将二进制数 11110.10101 转换成八进制数。

解：$(11110.10101)_B = (011\ 110.101\ 010)_B = (36.52)_O$

若将八进制数转换成二进制数，只需将八进制数的每一位用等值的3位二进制数代替即可。

例 1.1.6　将八进制数 703.6 转换成二进制数。

解：$(703.6)_O = (111\ 000\ 011.110)_B$

思考题

(1) 二进制数的特点是什么？

(2) 在数字电路中是如何实现二进制数的？

(3) 十六进制数主要用于何种场合？

(4) 十进制数与二进制数之间如何进行转换？

(5) 十六进制数与二进制数之间如何进行转换？

1.2　几种常用的编码

数字系统中的信息可分为两类，一类是数值，另一类是文字符号(包括控制符)。数值信息的表示方法如前所述。为了表示文字符号信息，往往也采用一定位数的二进制数码表示，这些数码并不表示数量的大小，仅仅区别不同事物而已。这些特定的二进制数码称为代码。以一定的规则编制代码，用以表示十进制数值、字母、符号等的过程称为编码。将代码还原成所表示的十进制数、字母、符号等的过程称为解码或译码。

1.2.1　二进制编码

若所需编码的信息有 N 项，则需要的二进制数码的位数 n 应满足如下关系：

$$2^n \geqslant N$$

例如，4位二进制码可以表示16个不同的数码，表1.2.1是常用的按8421权位排列的4位二进制编码表示的16个十进制数。

表 1.2.1　二进制编码

十进制数	二进制码	十进制数	二进制码
0	0000	8	1000
1	0001	9	1001
2	0010	10	1010
3	0011	11	1011
4	0100	12	1100
5	0101	13	1101
6	0110	14	1110
7	0111	15	1111

1.2.2　二-十进制编码(BCD)

二-十进制码就是用 4 位二进制数来表示 1 位十进制数中的 0 ～ 9 这 10 个数码，简称 BCD 码。4 位二进制数有 16 种不同的组合方式，即 16 种代码，根据不同的规则从中选择 10 种来表示十进制的 10 个数码，其方案有很多种。表 1.2.2 所示为几种常用的 BCD 码。

表 1.2.2　几种常见的 BCD 码

十进制数	8421BCD 码	2421BCD 码	5421BCD 码	余三码
0	0000	0000	0000	0011
1	0001	0001	0001	0100
2	0010	0010	0010	0101
3	0011	0011	0011	0110
4	0100	0100	0100	0111
5	0101	1011	1000	1000
6	0110	1100	1001	1001
7	0111	1101	1010	1010
8	1000	1110	1011	1011
9	1001	1111	1100	1100
权位	8421	2421	5421	无权

(1)8421BCD 码每一位权值是固定的，为恒权码，是使用最广泛的一种 BCD 码，具有自然而简单的特点。它从高位次到低位的权值分别为 8、4、2、1，每一位都像二进制数一样，具有标

准的 8421 位权,所以这种代码称为有权码。不过在 8421BCD 码中,是在 4 位自然二进制数去掉了后 6 种 1010 ~ 1111 状态,仅使用了 0000 ~ 1001 十个状态。

(2)2421BCD 码也是恒权码。从高位依次到低位的权值分别为 2、4、2、1,每组代码按位权展开求和就是它所代表的十进制码。由表 1.2.2 可看出 2421BCD 码具有互补性,该码中的 0 和 9、1 和 8、2 和 7、3 和 6、4 和 5 这 5 对代码互为反码,即两码对应位的取值相反。

(3)5421BCD 码也是恒权码。从高位依次到低位分别为 5、4、2、1,每组代码按权位展开求和就是它所代表的十进制码。

(4) 余 3 码是在 8421BCD 码加 3 后得到的,是一种常用的 BCD 码。余 3 码也具有互补性,0 和 9、1 和 8、2 和 7、3 和 6 及 4 和 5 的码组之间互为反码。余 3 码求反容易,有利于简化 BCD 码的减法运算。余 3 码是一种无权码。

BCD 码用 4 位二进制码表示的只是十进制数的 1 位。如果是多位十进制数,应先将该十进制数的每一位用 BCD 码表示,然后组合起来。

例 1.2.1 将十进制数 83 分别用 8421BCD 码、2421BCD 码和余 3 码表示。

解:由表 1.2.1 可得

$$(83)_D = (1000\ 0011)_{8421BCD}$$
$$(83)_D = (1110\ 0011)_{2421BCD}$$
$$(83)_D = (1011\ 0110)_{\text{余3码}}$$

1.2.3 其他编码

1. 格雷码

格雷码又称循环码。从表 1.2.3 的 4 位格雷码编码表中可以看出格雷码的构成方法,这就是每一位的状态变化都按一定的顺序循环。如果从 0000 开始,最右边一位的状态按 0110 顺序循环变化,右边第二位的状态按 00111100 顺序循环变化,右边第三位按 0000111111110000 顺序循环变化。可见,自右向左,每一位状态循环中连续的 0、1 数目增加一倍。由于 4 位格雷码只有 16 个,所以最左边一位的状态只有半个循环,即 0000000011111111。按照上述原则,我们就很容易得到更多位数的格雷码。

与普通的二进制代码相比,格雷码的最大优点就在于当它按照表 1.2.3 的编码顺序依次变化时,相邻两个代码之间只有一位发生变化。这样在代码转换的过程中就不会产生过渡“噪声”。而在普通二进制代码的转换过程中,则有时会产生过渡噪声。例如,第四行的二进制代码 0011 转换为第五行的 0100 过程中,如果最右边一位的变化比其他两位的变化慢,就会在一个极短的瞬间出现 0101 状态,这个状态将成为转换过程中出现的噪声。而在第四行的格雷码 0010 向第五行的 0110 转换过程中则不会出现过渡噪声。这种过渡噪声在有些情况下甚至会影响电路的正常工作,这时就必须采取措施加以避免。

表 1.2.3　4 位格雷码与二进制码的比较

编码顺序	二进制代码	格雷码
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

2. 美国信息交换标准代码(ASCII)

美国信息交换标准代码(American Standard Code for Information Interchange,ASCII 码)是由美国国家标准化协会(ANSI) 制定的一种信息代码,广泛地用于计算机和通信领域中。ASCII 码已经由国际标准化组织(ISO) 认定为国际通用的标准代码。

ASCII 码是一组 7 位二进制代码($b_7b_6b_5b_4b_3b_2b_1$),共 128 个,其中包括表示 0 ～ 9 的十个代码,表示大、小写英文字母的 52 个代码,32 个表示各种符号的代码以及 34 个控制码。表 1.2.4 是 ASCII 码的编码表,每个控制码在计算机操作中的含义列于表 1.2.5 中。

表 1.2.4　美国信息交换标准代码(ASCII 码)

$b_4b_3b_2b_1$	$b_7b_6b_5$							
	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	、	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	“	2	B	R	b	r