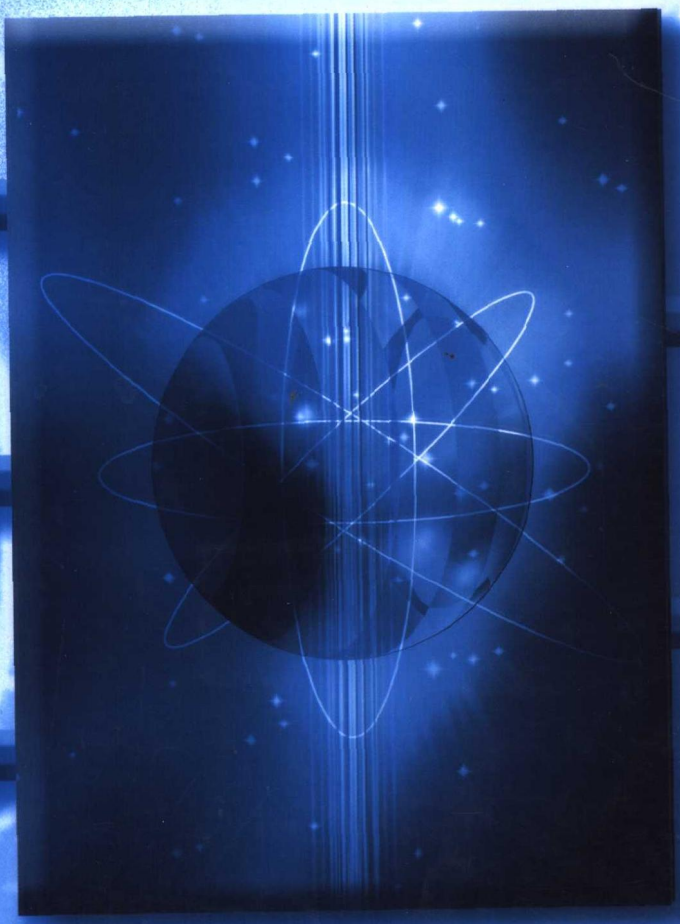
经典教材辅导用书

数字电子技术基础学习与解题指南

陈大钦 主编
陈大钦 彭容修 罗 杰 编



- 普通高等学校本、专科学生复习和备考
- 硕士研究生入学考试备考
- 成人高等教育、高等职业技术教育学生自学和备考

华中科技大学出版社

经典教材辅导用书

数字电子技术基础
学习与解题指南

陈大钦　主编
陈大钦　彭容修　罗　杰　编

华中科技大学出版社

图书在版编目(CIP)数据

数字电子技术基础学习与解题指南/陈大钦　主编
武汉:华中科技大学出版社，　2004 年 2 月
ISBN 7-5609-3082-4

Ⅰ. 数…
Ⅱ. ①陈…　②彭…　③罗…
Ⅲ. 数字电路-电子技术-高等学校-自学参考资料
Ⅳ. TN79

数字电子技术基础学习与解题指南　　陈大钦　主编

责任编辑:黄以铭　　封面设计:潘　群
责任校对:封春英　　责任监印:熊庆玉

出版发行:华中科技大学出版社
武昌喻家山　邮编:430074　电话:(027)87542624

录　　排:华中科技大学出版社照排室
印　　刷:湖北省京山县印刷厂

开本:787×960　1/16　　印张:20.75　　字数:364 000
版次:2004 年 2 月第 1 版　　印次:2004 年 2 月第 1 次印刷　　定价:26.00 元
ISBN 7-5609-3082-4/TN・78

内容提要

本书是以康华光主编、邹寿彬副主编《电子技术基础》(数字部分)(第四版)教材和阎石主编《数字电子技术基础》(第四版)教材为主要参考书而编写的教学和自学参考书。编者根据多年教学实践的经验,对教学内容进行了归纳、总结,为便于读者掌握课程的基本要求、重点和难点,书中精选了大量的例题与自我测试题,并附有解答。书末还附有典型的数字电子技术基础试卷和解答,以及硕士研究生入学考试模拟与数字电子技术基础试卷和解答。

该书内容丰富,思路清晰,适于普通高等学校本、专科学生复习和备考,以及硕士研究生入学考试备考,也适于高等职业技术教育和成人高等教育院校的学生自学、复习和备考,并可供从事电子技术教学的人员参考。

前言

数字电子技术基础是高等院校电气信息类(包括原自动化类、电气类、电子类)专业重要的专业基础课。本书是以康华光主编、邹寿彬副主编《电子技术基础》(数字部分)(第四版)教材和阎石主编《数字电子技术基础》(第四版)教材为主要参考书。并参照国家教委颁布的《高等工业学校电子技术基础课程教学基本要求》而编写的教学参考书。本书与陈大钦、彭容修编写的《模拟电子技术基础学习与解题指南》均为《电子技术基础》教材的配套用教学参考书。

全书内容与教学要求紧密配合,每章内容均包含重点和难点、例题精选和学习自评三部分。对教学中的重点和难点内容进行了综述与总结,使读者对核心内容有更为清晰的认识与理解,并进一步搞清楚各教学知识点的关联。精心编排的例题可供读者在巩固基本概念、基本分析方法的同时,掌握解决问题的思路和方法。为帮助读者检查每章的学习效果,在学习自评中安排了内容丰富的自测题并附有解答。最后书中还附有数字电子技术基础(本科)试卷及解答、硕士研究生入学考试模拟与数字电子技术基础试卷及解答,可供读者复习备考。

参加本书编写工作的有陈大钦(第1、4、7、8章、附录A及附录B)、彭容修(第2、3章)、罗杰(第5、6章)。陈大钦同志为主编,负责全书的组织和定稿。

限于编者的水平,不足之处难免,诚恳希望各院校的老师和读者提出批评和改进意见。

编　者

2003年8月于华中科技大学

目录

1 数字逻辑基础

知识要点

- 数制与码制
- 逻辑代数中的3种基本运算
- 逻辑代数的定律和定理
- 逻辑函数及其表示方法
- 逻辑函数变换与代数法化简
- 逻辑函数的卡诺图化简法

1.1 重点与难点

1.1.1 数制与码制

一、数制

人们在日常生活中，习惯于用十进制数，而在数字系统中多采用二进制数，有时也采用八进制数或十六进制数。

1. 十进制数

在十进制数中，每一位有0～9等10个数码中的一个，计数的基数是10，即逢10进1，本位复0。例如，十进制数412.71可写为

$$412.71 = 4 \times 10^2 + 1 \times 10^1 + 2 \times 10^0 + 7 \times 10^{-1} + 1 \times 10^{-2} \qquad (1.1.1)$$

若以D表示任意一个十进制数，以N取代式(1.1.1)中的10，即可得到任意进制(N进制)数展开式的普遍形式

$$D = K_{n-1}N^{n-1} + K_{n-2}N^{n-2} + \cdots + K_iN^i + \cdots + K_1N^1 + K_0N^0 + K_{-1}N^{-1} + K_{-2}N^{-2} + \cdots + K_{-m}N^{-m}$$

$$= \sum K_i N^i \qquad (1.1.2)$$

式中，K_i 是第 i 位的系数，是各种进制的数字符号中的某一个，如果是十进制，则是 0～9 这 10 个数码中的任何一个；

N^i 称为第 i 位的权；i 为位数，若整数部分的位数是 n，而小数部分的位数是 m，则 i 包含从 $n-1$ 到 0 的所有正整数和从 -1 到 $-m$ 的所有负整数。

2. 二进制数及其和十进制数的转换

在二进制数中，式(1.1.2)中的 $N=2$，每一位权仅有 0 和 1 两个可能的数码，即逢 2 进 1，本位复 0。

根据式(1.1.2)，任何一个二进制数展开式可表示为

$$D = \sum K_i 2^i \qquad (1.1.3)$$

例如，二进制数 1011.11 表示为

$$(1011.11)_2 = 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 1\times 2^0 + 1\times 2^{-1} + 1\times 2^{-2}$$

二进制数和十进制数的转换，是将上式各项的积相加，即得该二进制数相对应的十进制数，即

$$\begin{aligned}(1011.11)_2 &= 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 1\times 2^0 + 1\times 2^{-1} + 1\times 2^{-2}\\ &= (11.75)_{10}\end{aligned}$$

上式中分别使用下脚注 2 和 10 表示括号里的数是二进制数和十进制数①。

3. 十六进制数及其和十进制数的转换

在十六进制数中，它的每一位有 16 个不同的数码，分别用 0～9、A(10)、B(11)、C(12)、D(13)、E(14)、F(15)表示。因此，任何一个十六进制数均可展开为

$$D = \sum K_i \times 16^i \qquad (1.1.4)$$

例如，十六进制数 3BC.8C 可表示为

$$(3BC.8C)_{16} = 3\times 16^2 + 11\times 16^1 + 12\times 16^0 + 8\times 16^{-1} + 12\times 16^{-2}$$

十六进制数和十进制数的转换，是将上式各项的积相加，即得该十六进制数相对应的十进制数，即

---

① 有时也用 B(Binary)、D(Decimal)和 H(Hexadecimal)分别表示 2(进制)、10(进制)和 16(进制)脚注。

$$(3BC.8C)_{16} = 3\times16^2 + 11\times16^1 + 12\times16^0 + 8\times16^{-1} + 12\times16^{-2}$$
$$= 768 + 176 + 12 + 0.5 + 0.046875$$
$$= (956.546875)_{10}$$

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位的二进制数可以分别用2位、4位和8位的十六进制数表示，应用十分方便。

二、几种常用数制的相互转换

下面介绍几种常用计数制的互相转换法。

1. 多项式替代法

例如，要将十进制数12.5转换为二进制数，则首先将十进制数12.5改为多项式表示，即

$$(12.5)_{10} = (1\times10^1 + 2\times10^0 + 5\times10^{-1})_{10}$$

然后，将等式右边多项式中所有的十进制数转换为等值的二进制数，即

$$(12.5)_{10} = [1\times(1010)^1 + 10\times(1010)^0 + 101\times(1010)^{-1}]_2$$

再计算上列等式右边多项式之值，则得

$$(12.5)_{10} = [1010 + 10 + 0.1]_2 = [1100.1]_2$$

2. 基数除/乘法

基数除/乘法包括基数除法和基数乘法两种。整数的转换用基数除法，小数的转换用基数乘法。

(1) 基数除法

例如，用基数除法将十进制数102转换为二进制数，就是将102不断除2得余数。

2 | 102 ……余数=0=K_0 低位(LSB)①
2 | 51 ……余数=1=K_1
2 | 25 ……余数=1=K_2
2 | 12 ……余数=0=K_3
2 | 6 ……余数=0=K_4
2 | 3 ……余数=1=K_5
2 | 1 ……余数=1=K_6
0 高位(MSB)

① LSB系Least Significant Bit的缩写。MSB系Most Significant Bit的缩写。

由于十进制的数码 0 和 1 也就是二进制的数码 0 和 1，故余数不需要再替换，所以，转换后的结果为

$$(102)_{10}=(1100110)_2$$

(2) 基数乘法

用基数乘法将十进制小数转换为二进制小数很方便。例如，将十进制小数 0.71875 不断乘 2 取整数，便为转换后的二进制小数，即

```
        0.7 1 8 7 5
×                 2                          高位
-------------------
        1.4 3 7 5 0     ……整数部分=1=K-1        ↓
        0.4 3 7 5                              ↓
×                 2                            ↓
-------------------                            ↓
        0.8 7 5 0       ……整数部分=0=K-2        ↓
        0.8 7 5                                ↓
×                 2                            ↓
-------------------                            ↓
      1.7 5 0           ……整数部分=1=K-3        ↓
      0.7 5                                    ↓
×               2                              ↓
-------------------                            ↓
      1.5 0             ……整数部分=1=K-4        ↓
      0.5                                      ↓
×               2                              ↓
-------------------                            ↓
      1.0               ……整数部分=1=K-5      低位
```

所以　　$(0.71875)_{10}=(0.10111)_2$

3. 混合转换法

前面介绍的两种方法虽可适用于任意进制之间的转换，但转换过程的计算却是在不同的进位计数制中进行的。由于人们非常熟悉十进制，因而都尽量设法使两种不同进制数之间的转换在十进制中进行计算，这就是为什么一般总是将二进制数转换为十进制数时采用多项式替代法，而将十进制数转换为二进制数时则采用基数除/乘法的原因。

如果相互转换的两种进制数都不是人们所熟悉的十进制数，此时可以采用混合转换法，即将多项式替代法和基数除/乘法混合应用。

例如，如要将四进制数 2023.231 转换为五进制数，首先可用多项式替代法将四进制数 2023.231 转换为十进制数，即

$$\begin{aligned}(2023.231)_4=&(2\times4^3+0\times4^2+2\times4^1+3\times4^0\\&+2\times4^{-1}+3\times4^{-2}+1\times4^{-3})_{10}\end{aligned}$$

$$= (128 + 8 + 3 + 0.5 + 0.1875 + 0.015625)_{10}$$
$$= (139.703125)_{10}$$

再用基数除/乘法将所得十进制数转换为五进制数：

5 | 139	……余数=4	低位	↑
5 | 27	……余数=2		
5 | 5	……余数=0		
5 | 1	……余数=1	高位	
0			

0.703125 × 5 = 3.515625	……整数部分=3	高位
0.515625 × 5 = 2.578125	……整数部分=2	↓
0.578125 × 5 = 2.890625	……整数部分=2	↓
0.890625 × 5 = 4.453125	……整数部分=4	低位

转换结果为 $(2023.231)_4 = (1024.3224)_5$

4. 直接转换法

当一个 α 进制数要转换为 β 进制数时，如 α 进制与 β 进制的进位基数之间满足 2^k(k 为整数）关系，则采用直接转换法比采用上述任何方法都来得简单。例如，二进制的进位基数为 2，而八进制的进位基数为 8，若将二进制数 11011010011.10111 转换为八进制数，此时 $2^3=8$，即 $k = 3$，可用直接转换法将二进制数转换为八进制数。为此，首先将待转换的二进制数按 k(即 3) 位为一组分组。分组规则是，整数从低位到高位每隔 k 位分一组，最高位不够 k 位的在其前以 0 补足(如下式中虚线框内所示)；小数则从高位到低位每隔 k 位分一组，最低位不够 k 位的在其后也以 0 补足，即

$$\underbrace{\boxed{0}11}_{3}\ \underbrace{011}_{3}\ \underbrace{010}_{2}\ \underbrace{011}_{3}\ \cdot\ \underbrace{101}_{5}\ \underbrace{11\boxed{0}}_{6}$$

这样就可3位并1位而直接得到转换后的八进制数，即

$$(11011010011.10111)_2 = (3323.56)_8$$

三、码制

不同的数码不仅可以表示数量的不同大小，而且还能用来表示不同的事物。在后一种情况下，这些数码已没有表示数量大小的含意，只是表示不同事物的代号而已。这些数码称为代码。

在实际工作中，在编制代码时需要遵循一定的规则，这些规则就叫码制。

例如，在用4位二进制数码表示1位十进制数的0～9这十个状态时，就有多种的码制。通常将这些代码称为二-十进制代码，简称BCD代码。

表1.1.1列出了几种常用的BCD码，其中最简单也是用得最多的是8421BCD码。由于这种二进制码从左到右每一位的1分别表示8、4、2、1，所以把这种代码叫8421码。每一位的1代表的十进制数称为这一位的权。8421码属于恒权代码。

表1.1.1 几种常见的BCD代码

编码 / 种类 / 十进制数	8421码	5421码	2421码(A)	2421码(B)	5211码	余3码	余3循环码	格雷码
0	0000	0000	0000	0000	0000	0011	0010	0000
1	0001	0001	0001	0001	0001	0100	0110	0001
2	0010	0010	0010	0010	0100	0101	0111	0011
3	0011	0011	0011	0011	0101	0110	0101	0010
4	0100	0100	0100	0100	0111	0111	0100	0110
5	0101	1000	0101	1011	1000	1000	1100	0111
6	0110	1001	0110	1100	1001	1001	1101	0101
7	0111	1010	0111	1101	1100	1010	1111	0100
8	1000	1011	1110	1110	1101	1011	1110	1100
9	1001	1100	1111	1111	1111	1100	1010	1101
权	8421	5421	2421	2421	5211	无	无	无

如果要将8421码转换为十进制数，只需从最低位开始，按4位分一组，然后写出每4位8421码相对应的十进制数即可。例如，

$$(100101110101)_{8421BCD} = (975)_{10}$$

5211码从左到右各位的权值分别是5、2、1、1；2421码从左到右各位的权

分别是2、4、2、1。它们都是恒权码。

余3码中4位二进制数要比它所表示的十进制数码多3，因此，余3码是由8421码加3构成的。余3循环码和格雷码的特点是，相邻两个代码间仅有一位不同。例如，在余3循环码中，十进制数7的代码是1111，而8的代码是1110，只有最低位不同。在格雷码中，7的代码是0100，而8的代码是1100，只有最高位不同，若要从7变到8，它们只有一位发生变化，不会发生竞争冒险现象，这大大提高了可靠性，故它们都属于可靠性码。

余3码、余3循环码和格雷码，由于它们的每一位二进制码都没有固定的权值，故称为无权码。

此外，从表1.1.1中还可看出，0和9、1和8、2和7、3和6、4和5的余3码互为反码，这对于求取对10的补码是很方便的。

四、算术运算和逻辑运算

当两个二进制数码表示两个数量大小时，它们之间可以进行数值运算，这种运算称为算术运算。二进制算术运算和十进制算术运算的规则基本相同，唯一的区别是，二进制数是逢2进1，而十进制数是逢10进1。

例如，两个二进制数1001和0110的算术运算有

加法运算

$$\begin{array}{r} 1001 \\ +\ 0110 \\ \hline 1111 \end{array}$$

减法运算

$$\begin{array}{r} 1001 \\ -\ 0110 \\ \hline 0011 \end{array}$$

乘法运算

$$\begin{array}{r} 1001 \\ \times\ 0110 \\ \hline 0000 \\ 1001 \\ 1001 \\ 0000 \\ \hline 0110110 \end{array}$$

除法运算

$$\begin{array}{r|l} & 1.1 \\ \hline 0110 & 1001 \\ & 0110 \\ \hline & 0110 \\ & 0110 \\ \hline & 0 \end{array}$$

为了简化运算电路，在数字电路中两数相减的运算是用它们的补码来完成的。二进制数的补码定义是：

最高位为符号位，正数为0，负数为1；

正数的补码和它的原码同；

负数的补码可通过原码的数值逐位求反，然后在最低位加 1 得到。

例如，$(1001)_2-(0110)_2=0011$，若采用补码运算时，首先求出$(+1001)_2$和$(-0110)_2$的补码，它们分别为

$$(+1001)_2 = \underset{\text{符号位}}{0}\ 1\ 0\ 0\ 1$$

$$(-0110)_2 = \underset{\text{符号位}}{1}\ 1\ 0\ 1\ 0$$

然后将两个补码相加并舍去进位，即

$$\begin{array}{rl} & 0\ 1\ 0\ 0\ 1 \\ & +1\ 1\ 0\ 1\ 0 \\ \hline \text{舍去进位}\longleftarrow & 1\ \underset{\text{符号位}}{0}\ 0\ 1\ 1 \end{array}$$

得到与前面同样的结果。这样就把减法运算转化成了加法运算。进一步分析还可发现，二进制的加、减、乘、除运算都可用加法运算来完成，因而大大简化了运算电路结构。

当二进制代码不表示数量大小，而表示不同逻辑状态时，它们之间按照指定的某种因果关系进行的运算称为逻辑运算。逻辑运算与算术运算有着根本的区别。下面重点介绍各种逻辑运算。

1.1.2 逻辑代数的基本定律和定理

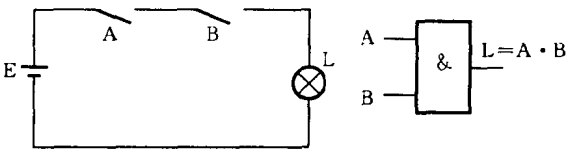
十进制中十个数字符号中的 0 和 1 是数值，而逻辑变量中的 0 和 1 不是数值，而是分别代表对立或矛盾着的两个方面。例如，信号的有和无，一件事情的是和非、真和假等，只代表两种不同的逻辑状态。

一、逻辑代数的三种基本运算

在普通代数学中，变量的运算包括加、减、乘、除、乘方及开方等多种运算，而逻辑代数中变量的基本运算只有与、或、非三种。

1. 与运算——逻辑乘

图 1.1.1(a)表示一个简单的与逻辑电路，电源 E 通过开关 A 和 B 向灯泡 L 供电，只有 A 和 B 同时接通，灯泡 L 才亮；A 和 B 中只要有一个或两个不接通时，则灯泡 L 不亮。因此，逻辑与电路可总结为这样的逻辑关系：只有当一件事(灯亮)的几个条件全部具备(开关 A 和 B 都接通)之后，这件事(灯



(a) 电路　　(b) 与逻辑门电路符号

图 1.1.1　与逻辑运算

亮)才发生,这种关系称为与逻辑。如果用二值逻辑 0 和 1 来表示,并设开关断开和灯不亮均用 0 表示,开关闭合和灯亮都用 1 表示,则可列出其逻辑关系的图表如表 1.1.2 所示,这种图表叫逻辑真值表(简称真值表)。与运算的逻辑表达式为

$$L=A\cdot B \qquad (1.1.4)$$

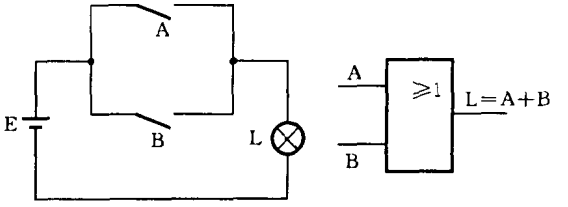
式中,小圆点"·"表示 A、B 的与运算,也表示逻辑乘。为书写简便起见,乘号"·"常被省略,图 1.1.1(b)是与逻辑门电路符号。

表 1.1.2　与逻辑运算真值表

A	B	L
0	0	0
0	1	0
1	0	0
1	1	1

2. 或运算——逻辑加

图 1.1.2(a)是用来说明或运算基本概念的电路图。显然,在此电路中,只要开关 A、B 中有一个闭合,灯 L 亮这件事就会发生,故逻辑或的定义为:如果一个事件(灯亮)的发生决定于多个条件(开关 A 或 B 闭合),只要其中一个条件具备,此事就会发生。



(a) 电路　　(b) 或逻辑门电路符号

图 1.1.2　或逻辑运算

仿照前述,可得到用 0、1 表示的或逻辑真值表如表 1.1.3 所示。或运算的逻辑表达式为

$$L=A+B \qquad (1.1.5)$$

式中,符号"+"表示 A、B 或运算,也表示逻辑加。

表 1.1.3　或逻辑运算真值表

A	B	L
0	0	0
0	1	1
1	0	1
1	1	1