SHUZI DIANZI JISHU

数字电子技术

王桂馨　张惠敏　编
王学力　审

中国铁道出版社

数 字 电 子 技 术

王桂馨　张惠敏　编
王学力　　　审

中　国　铁　道　出　版　社
2 0 0 2年·北京

(京)新登字 063 号

内 容 简 介

本教材采用模块化结构,包括数字电路基础知识,组合逻辑电路,时序逻辑电路,脉冲产生、变换及数模接口电路,大规模集成电路,综合演练等内容。各模块分为理论学习和课堂演练部分,并附有阅读材料和自我检测题,使理论知识与实际应用紧密结合,突出重点。

本书可作为高等职业技术学校和高等专科学校工科类专业教材,也可供中等专业学校师生参考。

图书在版编目(CIP)数据

数字电子技术/王桂馨,张惠敏编 .—北京:中国铁道出版社,2002.2
ISBN 7-113-04534-0

Ⅰ.数… Ⅱ.①王…②张… Ⅲ.数字电路—电子技术—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 004640 号

书　　名:数字电子技术
作　　者:王桂馨 张惠敏 编
出版发行:中国铁道出版社(100054,北京市宣武区右安门西街 8 号)
责任编辑:任 军 崔忠文 联系电话 (021)73147(路电) (010)51873147(市电)
封面设计:李艳阳
印　　刷:北京兴顺印刷厂
开　　本:787×1 092 1/16 印张:11.5 字数:279 千
版　　本:2002 年 2 月第 1 版 2002 年 2 月第 1 次印刷
印　　数:1~3000 册
书　　号:ISBN 7-113-04534-0/TN·144
定　　价:22.00 元

前　言

本书是为高等职业技术学校和高等专科学校工科类专业“电子技术”课程编写的教材。

电子技术是一门理论与实践紧密结合的课程。它主要介绍与电子技术有关的基本概念、基本分析方法、基本逻辑器件和基本应用，包括模拟电子技术与数字电子技术两部分。本书讨论数字电子技术部分的有关知识。

根据高职、高专的培养目标和要求，在编写本书时突出了以下特点：

1.降低理论知识的深度。对理论知识，要求着重掌握应用特性，而对深奥繁杂的理论来源与推导尽量采用“黑箱”的处理方法。

2.注重技能的培养与训练。具体做法是：

(1)采用模块化结构。主要内容分为理论学习和课堂演练两部分，使理论知识与实际应用紧密联系，前后呼应，学用结合。课堂演练只提出了参考内容，没有统一的格式规范，教师可根据各校电子设备的实际情况灵活掌握。

(2)按照技能培训的规律，由浅入深、循序渐进地安排教学内容。

(3)书中有查阅电子元器件手册及合理选用电子元器件的训练内容。

3.各模块后均附有自我检测题，便于学生深入理解和掌握书中的内容。

4.书中某些模块后附有阅读材料，阅读材料不作为应知应会的内容。阅读材料有以下三类：

(1)“黑箱”化处理后某些“黑箱”内的理论知识，写入阅读材料以作参考。

(2)当课堂演练条件不具备时，作为正文的补充。

(3)拓宽知识内容，以满足不同层次学生的需要。

5.适当引入电子新器件、新技术的内容，以适应培养面向21世纪人才的需要。

本书自1998年试用以来，在教学实践中不断修改和完善，深受广大师生欢迎。此次出版又进行了适当的修订，但模块的基本结构不变。全书由王桂馨、张惠敏编，王学力审。模块1、模块4由王桂馨编写，模块2、模块3、模块5、综合演练由张惠敏编写。

本书与王瑞琴、刘素芳编，孙建设审的《模拟电子技术》配套使用。

本书在编写及出版过程中得到了郑州铁路职业技术学院各级领导及应用电子教研室的热情帮助和大力支持，在此表示衷心的感谢！

由于编者水平有限，书中的错误及不妥之处敬请读者批评指正。

编　者

2002 年 1 月

目　　录

模块 1　数字电路基础知识

数字量的计数体制,也叫数制。在生产实践中人们习惯用的计数体制是十进制,而在数字电路中用得最广泛的数制却是二进制和十六进制。

1.1　数制与码制

1.1.1　数　　制

1.十进制

十进制是日常生活和生产中最常用的计数体制。它的每一位用 0~9 十个数码表示,基数为 10。超过数码 9 的数则需要用多位数表示,其中相邻数间的关系是逢十进一或借一当十,简称十进制。

每个数字处在不同数位所代表的数值是不同的,例如十进制数 736 可表示为:

$$(736)_{10} = 7 \times 10^2 + 3 \times 10^1 + 6 \times 10^0$$

其中 10^2、10^1、10^0 分别为百位、十位、个位的权,也就是相应位的 1 所代表的实际数值。

一个有 n 位整数和 m 位小数的正十进制数应是各个位值的和。它的权展开式为:

$$\begin{aligned}(N)_{10} &= K_{n-1}10^{n-1} + \cdots + K_1 10^1 + K_0 10^0 + K_{-1}10^{-1} + \cdots + K_{-m}10^{-m} \\ &= \sum_{i=-m}^{n-1} K_i 10^i \qquad (1.1.1)\end{aligned}$$

式中 K_i 为基数 10 的 i 次幂的系数,它可为 0 ~ 9 中任一个数字。

n、m 可为 $-\infty \sim +\infty$ 之间的任一正整数。

关于负的十进制数表示方法则在计算机原理中介绍,在这儿就不多赘述。

例 1　将十进制数 444.55 写成权展开式。

解:　$(444.55)_{10} = 4 \times 10^2 + 4 \times 10^1 + 4 \times 10^0 + 5 \times 10^{-1} + 5 \times 10^{-2}$

2.二进制

在数字电路中应用最广泛的是二进制。

在二进制数中,每一位仅有 1 和 0 两个数码,所以计数的基数为 2。相邻位数间的关系是逢二进一或借一当二。一个有 n 位整数和 m 位小数的二进制权展开式为:

$$\begin{aligned}(N)_2 &= K_{n-1}2^{n-1} + \cdots + K_1 2^1 + K_0 2^0 + K_{-1}2^{-1} + \cdots + K_{-m}2^{-m} \\ &= \sum_{i=-m}^{n-1} K_i 2^i \qquad (1.1.2)\end{aligned}$$

式中 K_i 为基数 2 的 i 次幂的系数,它可以为 1,也可以为 0。

n、m 为 $-\infty \sim +\infty$ 之间的任一正整数。

例 2　将二进制数写成权展开式。

解:　$(11101.101)_2 = 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$

上述表示方法可以推广到以任何数字 r 为基数的 r 进制，其权展开式为：

$$(N)_r = K_{n-1}r^{n-1} + \cdots + K_1r^1 + K_0r^0 + K_{-1}r^{-1} + \cdots + K_{-m}r^{-m}$$
$$= \sum_{i=-m}^{n-1} K_i r^i \qquad (1.1.3)$$

3.十六进制数

(1)十六进制的计数方法

由于实现二进制数所用器件容易获得，加上二进制算术运算简单，所以在数字系统中广泛使用二进制。但二进制与十进制比较，人们使用不大方便，与等值十进制数相比，它所需要的位数多，人们不便于书写和记忆，因此在计算机应用中经常用十六进制数来表示。

十六进制数采用0~9、A、B、C、D、E、F共16个数码，基数为16，故相邻位数关系是逢十六进一或借一当十六。一个有 n 位整数和 m 位小数的正十六进制权展开式为：

$$(N)_{16} = \sum_{i=-m}^{n-1} K_i \quad 16^i$$

式中 K_i 为基数16的 i 次幂的系数，它可为0~F这16个数字中的任一个数字。

n,m 为 $-\infty \to +\infty$ 之间的任一正整数。

例3 将十六进制数5.A5写成权展开式。

解： $(5.A5)_{16} = 5\times16^0 + 10\times16^{-1} + 5\times16^{-2}$

(2)数制对照表

现将十进制、二进制、十六进制对照比较于表1.1.1中。

表1.1.1 数制对照表

对照的内容	十 进 制	二 进 制	十 六 进 制
数字符号	0,1,2,3,4,5,6,7,8,9	0,1	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
进位规律	逢十进一(借一当十)	逢二进一(借一当二)	逢十六进一(借一当十六)
基数 R	10	2	16
第 i 位权	10^i	2^i	16^i
任意整数、小数表达式	$(N)_{10} = \sum_{i=-m}^{n-1} K_i10^i$	$(N)_2 = \sum_{i=-m}^{n-1} K_i2^i$	$(N)_{16} = \sum_{i=-m}^{n-1} K_i16^i$

4.不同数制之间的转换

(1)二进制转成十进制

任何二进制数均可按权相加法转化成十进制数，也就是将各位二进制数的权值乘上系数，再相加就可得相应十进制数。

例4 求二进制数11 010.101相应的十进制数。

解：$(11010.101)_2 = 1\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0 + 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3}$

$= 16 + 8 + 0 + 2 + 0 + 0.5 + 0 + 0.125 = (26.625)_{10}$

为了方便地使用按权相加法，应熟记二进制数各位的权，表1.1.2列出了二进制数常用的位权。

(2)十进制数转换成二进制数

数值的转换分整数和小数两部分进行。

表 1.1.2　常用二进制数的权

i	−4	−3	−2	−1	0	1	2	3	4	5	6	7	8	9	10	11	12
2^i	0.062 5	0.125	0.25	0.5	1	2	4	8	16	32	64	128	256	512	1 024	2 048	4 096

①整数换算——除 2 取余法

为了便于理解，以实例说明。

例 5　将十进制数 29 换算为二进制数。

解：换算步骤表示为

```
2|29
2|14……1
2|7 ……0        二进制数
2|3 ……1        读写次序
2|1 ……1
  0 ……1
```

换算结果：$(29)_{10}=(11101)_2$。

由此可见，将十进制数不断用 2 去除，直到商为 0，从下向上写出余数，即为所求的二进制数。

②小数的换算——乘 2 取整法

下面以实例说明。

例 6　将十进制纯小数 0.562 转换成误差 ε 不大于 2^{-6} 的二进制小数。

解：可由乘 2 取整法求取相应二进制小数。

	取整	
$0.562\times2=1.124$	1	二进制数
$0.124\times2=0.248$	0	读写次序
$0.248\times2=0.496$	0	
$0.496\times2=0.992$	0	
$0.992\times2=1.984$	1	

由于最后余的小数 $0.984>0.5$，则根据“四舍五入”的原则可得 $(0.562)_{10}=(0.100\ 011)_2$，且误差 $\varepsilon<2^{-6}$。

任何十进制数均可按整数部分和纯小数部分分开，分别用除 2 取余法和乘 2 取整法化成二进制数形式，然后将二进制形式的整数和纯小数合成与十进制数所对应的二进制数。

1.1.2　码　　制

在数字系统中，常将有特定意义的信息（如数字、文字、符号）用一种规定的二进制代码表示，这种规定叫做码制。各种码制的编码方式有所不同。

1.常用二－十进制码（简称 BCD 码）

在计算机中，十进制数除了转换成二进制数以外，还有一种表示方法，那就是十进制数的代码表示法。它具有二进制数的形式，却有十进制数的特点。它是用四位二进制数来表示十

进制数中的 0～9 十个数码,简称 BCD 码。由于四位二进制数码可以表示 16 种不同的组合状态,所以表示一位十进制数(只有 0～9 十个数码)只需选择其中十个状态的组合,其余 6 种组合是无效的。按选取方式的不同,可以得到不同的二－十进制编码。表 1.1.3 列出了常见的几种 BCD 编码。

表 1.1.3 常用的几种 BCD 编码

十进制数	有权码				无权码		
	8421	5421	2421A	2421B	余 3 码	余 3 循环码	右移码
0	0000	0000	0000	0000	0011	0010	00000
1	0001	0001	0001	0001	0100	0110	10000
2	0010	0010	0010	0010	0101	0111	11000
3	0011	0011	0011	0011	0110	0101	11100
4	0100	0100	0100	0100	0111	0100	11110
5	0101	1000	0101	1011	1000	1100	11111
6	0110	1001	0110	1100	1001	1101	01111
7	0111	1010	0111	1101	1010	1111	00111
8	1000	1011	1110	1110	1011	1110	00011
9	1001	1100	1111	1111	1100	1010	00001

BCD 码可以作为人与计算机联系时的一种中间表示。在某些情况下,计算机也可以对这种形式表示的数直接进行运算。

(1)8421 码　这是一种最常用的 BCD 码。由于这种编码的四位数码从左到右各位对应权值分别为 2^3、2^2、2^1、2^0,即 8、4、2、1,所以叫做 8421 码,是有权码的一种。四位二进制代码共有 16 种不同组合形式。在这 16 种代码形式中,选取 0000～1001 前十种组合表示一位十进制数,而对于后六种 1010～1111 则舍去不用。可见,每个十进制数码所对应的四位二进制代码就是该十进制数等值的二进制数。例如,一个十进制数 349.65 可用 8421 码写成:

十进制数	3	4	9	6	5
8421 码	0011	0100	1001	0110	0101

(2)其他有权码　只要满足最低权位值为 1,四位权值之和大于等于 9,且能区分开 0～9 十个数码的均可构成有权的 BCD 码。表中其他各种有权码只是编码顺序不同而已,可根据需要选用。

(3)余 3 码　余 3 码各位无固定权,属无权码。它是由 8421 码加 3 后得到的。或者说,它是取 16 个四位二进制代码中间的 10 个,前后各去掉三个得到的。这样,按二进制换算出等值十进制数间差 3,故称为余 3 码。

余 3 码的主要特点是表示 0 和 9、1 和 8、2 和 7、3 和 6、4 和 5 的二进制代码间互为反码(对应位数码相反),常称自补代码。它对简化二－十进制运算电路非常有利。

2.格雷码

格雷码又称循环码、反射码,这也是一种无权码。表 1.1.4 列出了四位的格雷码。它利用了所有 16 种组合,因此也是 BCD 码。它的编码特点是任意两个相邻代码间只有一位数码不

表 1.1.4 格雷码

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

同，这对代码的转换和传输非常有利。当从一个代码过渡到相邻的另一代码时，不会瞬间出现许多不该出现的代码。反之，若相邻代码中有两个或两个以上数码不同，则从一个代码过渡到另一代码时，会瞬间出现许多别的代码，在有些场合，可能造成逻辑错误。格雷码则可以避免这种瞬间模糊状态。所以，它是一种错误最小化代码，因此获得广泛应用。

［课堂演练］

1.数制的转换

(1)将十进制数13.562转换成误差不超过2^{-6}的二进制数。

解:先将整数部分用除2取余法化成二进制数:

$$(13)_{10}=(1101)_2$$

再用乘2取整法将纯小数部分化成二进制数:

$$(0.562)_{10}=(0.100011)_2$$

然后将所得结果合成相应的二进制数:

$$(13.562)_{10}=(1101.100011)_2$$

2.十进制数的代码表示

(1)将十进制数173转换成三位8421BCD码。

解:$(173)_{10}=(0001\quad 0111\quad 0011)_{8421BCD}$

(2)将三位8421BCD码1001　0111　1000转换成十进制数。

(3)将十进制数173转换成三位5421BCD码。

(4)将十进制数173转换成三位余3BCD码。

(5)将三位8421BCD码1001　0011　0101转换成5421BCD码和三位余3BCD码。

［阅读材料］

1.二进制数的优缺点

二进制数与十进制数相比，其优点是:

①二进制数只有二个数字符号0和1，因此很容易用电路元件的状态来表示。例如，二极管的通和断，三极管的饱和及截止，继电器的接通和断开，灯泡的明灭，电平的高低等，都可以将其中一个状态定为0，另一个状态定为1，来表示这个二进制数，这种表示简单可靠，所用元件少，存储和传送二进制数也十分方便。

②二进制基本运算规则与十进制运算规则相似，但要简单得多。

例如:二个一位十进制数相乘，其规律要用“九九乘法表”才能表示，而二个一位二进制数相乘只有$0\times0=0$，$0\times1=0$，$1\times0=0$，$1\times1=1$这四种组合，用电路来实现二进制运算会很方便。

与十进制相比较，二进制的缺点是:

①人们对二进制数不熟悉，使用不习惯。

②同样表示一个数，二进制数的位数要比十进制位数多，例如一个二位十进制数58，用二进制数表示111010，需六位。

因此用计算机计算时，通常先要将人们熟悉的十进制原始数据先进行“十翻二”运算，变成计算机能接受的二进制数;运算的结果为二进制数，再进行“二翻十”转换，变成人们易接受的十进制数。

2. 十进制数转换成二进制数

①任何十进制数正数的整数部分均可用除 2 取余法转换成二进制数。

除 2 取余法是由下列原理推出的。

若某一个十进制数 N 可转化为三位二进制数：

$$(N)_{10}=(K_2K_1K_0)_2 \tag{1.1.4}$$

则
$$(K_2K_1K_0)_2=(K_2\times 2^2+K_1\times 2^1+K_0\times 2^0)_{10}=[2(K_2\cdot 2^1+K_1)+K_0]_{10} \tag{1.1.5}$$

该式说明：

$$(N)_{10}\div 2=(K_2\cdot 2^1+K_1)\quad \cdots\cdots 余数为\ K_0 \tag{1.1.6}$$

用上面除得的商再除以 2，则得：

$$(K_2\cdot 2+K_1)\div 2=K_2\quad \cdots\cdots 余数为\ K_1 \tag{1.1.7}$$

不断用前次的商除以 2，一直算到最后的商为零，即 $K_2\div 2=0$ ……余数为 K_2。

②任何十进制数的纯小数部分可用乘 2 取整法求相应的二进制数。

乘 2 取整法是由下列原理推出的。

由于十进制纯小数化成 n 位二进制纯小数可写成：

$$(N)_{10}=K_{-1}\cdot 2^{-1}+K_{-2}\cdot 2^{-2}+\cdots+K_{-(n-1)}\cdot 2^{-(n-1)}+K_{-n}2^{-n} \tag{1.1.8}$$

将上式两边乘以 2 得：

$$2\times(N)_{10}=K_{-1}\times 2^0+K_{-2}\times 2^{-1}+\cdots+K_{-(n-1)}\cdot 2^{-(n-2)}+K_{-n}2^{-(n-1)} \tag{1.1.9}$$

由此可见，十进制小数乘以 2，取其整数即为 2^0 的系数，由上式可求得 K_{-1}。

不难推知，将十进制纯小数每次减去上次所得积中 2^0 的系数（整数），再乘以 2，…直到满足误差要求，进行"四舍五入"为止，各次取整所得系数依次为 K_{-1}、K_{-2}、…、$K_{-(n-1)}$、K_{-n}。

3. 十进制数、二进制数与十六进制数的相互转换

(1)二进制数转换成十六进制数

由于十六进制数的基数 $R=16$，而 $16=2^4$，因此四位二进制数就相当于一位十六进制数。若要将多位二进制数化为十六进制数，则其整数部分可由小数点起向左推，四位成一组，直至分完最高位，若此时不足四位，可在高位补零；其小数部分可由小数点起向右推，四位成一组，直至分完最低位，若此时不足四位，可在低位补零。完成上述分组后，将每组的四位二进制数转换成十六进制数，然后将总的结果按序排列，就完成了二进制数到十六进制数的转换。

例 7 将二进制数（1011011.101101）转换为十六进制数的形式。

解：先将 1011011.101101 化成分组形式：$(0101\quad 1011.\quad 1011\quad 0100)_2$

然后将各组四位二进制数化成十六进制数，得：$(5B.B4)_{16}$。

(2)十六进制数转换成二进制数

由于十六进制的一位数对应于四位二进制数，因此任意十六进制数均可由各位数先变成四位二进制数，然后按十六进制数的排序将这些四位二进制数由高到低排序，就可得相应二进制形式。

例 8 $(7E65A)_{16}=(\quad ?\quad)_2$

解：可分别将 7、E、6、5、A 化成相应的二进制数 0111、1110、0110、0101、1010，再按位数的高低依次排列，就可得：

$$(7E65A)_{16}=(111\quad 1110\quad 0110\quad 0101\quad 1010)_2$$

(3)十六进制数转换成十进制数

可由按权相加法将十六进制数转换为十进制数。

例 9 $(7A.58)_{16}=(\quad ? \quad)_{10}$

解：$(7A.58)_{16}=7\times16^{1}+10\times16^{0}+5\times16^{-1}+8\times16^{-2}$

$$=112+10+0.312\,5+0.031\,25=(122.343\,75)_{10}$$

(4)十进制数转换成十六进制数

可将十进制数用前面介绍的方法先转换成二进制数，再将二进制数分成四位一组，转换成十六进制数。

例 10 $(18.625)_{10}=(\quad ? \quad)_{16}$

解：可先将整数 18 化成二进制数：$(18)_{10}=(10010)_{2}$，

再将纯小数 0.625 化成二进制数：$(0.625)_{10}=(0.101)_{2}$，

可得：$(18.625)_{10}=(10\,010.101)_{2}$，

再由二进制数转换成十六进制数：$(18.625)_{10}=(00010010.1010)_{2}=(12.A)_{16}$。

当然也可由十进制数的整数部分直接除 16 取余，纯小数部分不断乘 16 取整来转换。$(18.625)_{10}$的整数部分为 18，不断除 16 取余为：

$$18\div16=1\qquad 取余\ 2\cdots\cdots K_0$$

$$1\div16=0\qquad 取余\ 1\cdots\cdots K_1$$

$(18.625)_{10}$的小数部分为 0.625，乘 16 取整为：$0.625\times16=10=A\cdots K_{-1}$

$$(18.625)_{10}=(12.A)_{16}$$

4.码制

除了前面介绍的两种可靠性编码外，奇偶校验码也是一种能够检验出代码传递是否错误的可靠性编码。表 1.1.5 列出了这种代码。

表 1.1.5 奇偶校验码

十进制数	奇校验 8421BCD		偶校验 8421BCD	
	信　息　位	校验位	信　息　位	校验位
0	0 0 0 0	1	0 0 0 0	0
1	0 0 0 1	0	0 0 0 1	1
2	0 0 1 0	0	0 0 1 0	1
3	0 0 1 1	1	0 0 1 1	0
4	0 1 0 0	0	0 1 0 0	1
5	0 1 0 1	1	0 1 0 1	0
6	0 1 1 0	1	0 1 1 0	0
7	0 1 1 1	0	0 1 1 1	1
8	1 0 0 0	0	1 0 0 0	1
9	1 0 0 1	1	1 0 0 1	0

在奇偶校验码中，一个代码组包含两部分：一是信息位，这就是需要传送的信息本身可以是位数不限的二进制代码组，例如并行传送 8421BCD 码，信息就是如表中信息位所示的四位；二是奇偶校验位，它仅有一位。它的编码有两种方式：使得一个代码组中信息位和校验位中“1”的总个数为奇数的叫奇校验；“1”的总个数为偶数的叫偶校验。奇校验和偶校验在计算机中都获得广泛的应用。通常，对接收到的奇偶校验码要进行检查，看码中“1”的个数的奇偶是否正确。如果不对，就是错误代码(或称非法码)，也就是说明信息传递有错。

按表 1.1.5 传送奇校验码，如收到的代码组为 00010，00111 等，我们认为传送正确。若收到代码组为 01001、01111 等，则因为“1”的总数为偶数，就是非法码。应该指出的是奇偶校验码中如发生双错（有两位出错），奇偶校验码是查不出来的，但双错的机率要比单错少得多，所以这种校验还是很有价值的。奇偶校验码增加了一位校验码，当然增加了设备量和信息传送量，这是以数量换取可靠性的一种方法。

当信息处理设备需要处理十进制数字、字母和专门符号时，需要采用一种字符代码，它以二进制编码形式对所有符号作了规定，其中用得最广泛的是二种：国际通用的美国信息交换标准代码 ASCII 码和国际标准化组织的 ISO 代码。在本课程中并不涉及，就不多介绍。

1.2 基本逻辑门

生产和生活中的事物之间都存在某种因果关系，通常把这种因果关系称为逻辑关系。最基本的逻辑关系为“与”关系、“或”关系和“非”关系。而逻辑门是指利用电子元件实现因果关系的电子电路。因此，基本逻辑门有与门、或门和非门。

1.2.1 与逻辑及与门

1. 与逻辑

（1）与逻辑的定义与真值表

若决定某一事件的所有条件都成立，这个事件就发生，否则这个事件就不发生，这样的逻辑关系称为逻辑与或者逻辑乘。

例如图 1.2.1 中，灯 L 亮的条件是：开关 A 和 B 都闭合。如规定：开关“闭合”为逻辑“1”、开关“断开”为逻辑“0”；灯“亮”为逻辑“1”，灯灭为逻辑“0”，那么，就可把 A、B 看作输入变量，L 看作输出变量，并且用表 1.2.1 列出输入变量 A、B 的各种取值组合和输出变量 L 的一一对应关系，这种用表格形式列出的逻辑关系，叫做真值表。

表 1.2.1 与逻辑真值表

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

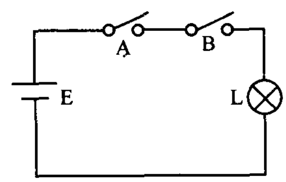


图 1.2.1 与逻辑电路

开关 A、B 是串联的，只要有一个或二个处于“断开”状态，灯 L 就不会亮，也就是“有 0 出 0”，只有 A、B 全处于“闭合”状态，灯 L 才会亮，也就是“全 1 出 1”。因此，在表 1.2.1 中，前三行因 A、B 中有“0”，L 为“0”；第四行 A、B 均为“1”，L 输出为“1”。这是一个与逻辑关系的真值表。

（2）与逻辑的逻辑函数式及运算规则

逻辑与除用真值表表示外，还可以用输入、输出变量之间的逻辑关系表达式来表示，也就是用逻辑函数式来表示：

$$L = A \cdot B \tag{1.2.1}$$

式中符号“·”叫做逻辑乘(又叫与运算),它不是普通代数中的乘号。L是输入变量A、B逻辑乘的结果,叫做逻辑积,它也不是普通代数中的乘积。

根据逻辑与的定义,逻辑乘的表达式可以推广到多输入变量的一般形式:

$$L = A \cdot B \cdot C \cdot D \cdots \tag{1.2.2}$$

为了书写方便,式中符号“·”可以不写,式(1.2.2)可简写为:

$$L = ABCD \cdots \tag{1.2.3}$$

它表示与运算中,所有输入与输出的关系为“有0出0,全1出1”。

与运算的规则如下:

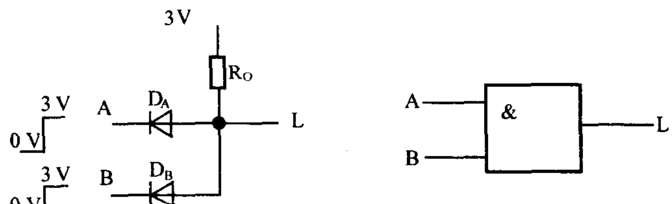
$$0 \cdot 0 = 0 \qquad 0 \cdot 1 = 0$$

$$1 \cdot 0 = 0 \qquad 1 \cdot 1 = 1$$

2.与门

实现与逻辑运算的电路叫与门。

二极管与门电路如图1.2.2(a)所示,输入端A、B代表条件,输出端L代表结果。



(a)与门电路 (b)逻辑符号

图1.2.2 与门

当 $V_A = V_B = 0$ V时,D_A、D_B 均处于正偏而导通,则输出为低电平,$V_{OL} = 0$ V(略去二极管的正向导通电压,也就是把二极管看作理想的二极管)。

当 $V_A = 3$ V,$V_B = 0$ V时,由于 D_B 两端电位差大,因此比 D_A 优先导通,D_B 导通后,使输出L钳位在 $V_{OL} = 0$ V,导通二极管 D_A 反偏,故 D_A 截止。

同理,当 $V_A = 0$ V,$V_B = 3$ V时,则 $V_{OL} = 0$ V。

当 $V_A = V_B = 3$ V时,D_A、D_B 均正偏而导通,L点被钳定在高电平,$V_{OH} = V_A = 3$ V。

表1.2.2列出上述A、B四种组合时各二极管的工作情况及L的输出电平。

表1.2.2 与门输入—输出电位关系

V_A(V)	D_A	V_B(V)	D_B	V_L(V)
0	导 通	0	导 通	0
0	导 通	+3	截 止	0
+3	截 止	0	导 通	0
+3	导 通	+3	导 通	+3

如果规定高电平3V为逻辑“1”,低电平0V为逻辑“0”,那么由表1.2.2就可推得表1.2.1所示逻辑关系,即输入变量A、B与输出变量L的关系完全符合逻辑与真值表。因此图1.2.2(a)电路是一个与门电路,其逻辑表达式也如式(1.2.1)所示:

$$L = A \cdot B$$

如果与门的输入变量不仅是A和B,还有C和D,那么只要在图1.2.2(a)电路的基础上再加上二个二极管就可以了。如图1.2.3(a)。

与门电路可用逻辑符号来代表,如图1.2.2(a)所示的二输入与门可用图1.2.2(b)逻辑符