高职高专教育电类系列教材

GAOZHI GAOZHUAN JIAOYU DIANLEI XILIE JIAOCAI

电子技术基础

（数字篇）

安徽省高职高专教育“十一五”规划教材

DIANZI JISHU JICHU SHUOZIPIAN

主编　杨林国　　副主编　丁向荣　孙晓雷

主　审　王　彪

安徽科学技术出版社

高职高专教育电类系列教材

安徽省高职高专规划教材

安徽省高职高专教育“十一五”规划教材

电子技术基础

（数字篇）

◉主　编　杨林国

副主编　丁向荣　孙晓雷

参　编　吴金权　李碧红

　　　　郝志廷　刘　兵

主　审　王　彪

安徽科学技术出版社

图书在版编目(CIP)数据

电子技术基础·数字篇/杨林国主编. —合肥:安徽科学技术出版社,2007.9
高职高专教育电类系列教材
ISBN 978-7-5337-3808-2

Ⅰ. 电… Ⅱ. 杨… Ⅲ. ①电子技术-高等学校:技术学校-教材②数字电路-电子技术-高等学校:技术学校-教材 Ⅳ. TN

中国版本图书馆 CIP 数据核字(2007)第 125900 号

内 容 提 要

本教材的主要内容有数字电路基础、集成逻辑门电路、组合逻辑电路、触发器、时序逻辑电路、半导体存储器和可编程逻辑器件、脉冲信号的产生与整形、数模和模数转换器。

本教材可供高职电气自动化类、电子信息类及相关专业使用,同时也可供从事电子技术工作的工程技术人员及参加职业技能鉴定的人员阅读参考。

电子技术基础·数字篇 杨林国 主编

出 版 人:朱智润
责任编辑:何宗华 期源萍
出版发行:安徽科学技术出版社(合肥市政务文化新区圣泉路 1118 号出版传媒广场,邮编:230071)
电 话:(0551)3533330
网 址:www.ahstp.com.cn
E - mail:yougoubu@sina.com
经 销:新华书店
排 版:安徽事达科技贸易有限公司
印 刷:合肥星光印务有限责任公司
开 本:787×1092 1/16
印 张:14.75
字 数:360 千
版 次:2007 年 9 月第 1 版 2007 年 9 月第 1 次印刷
印 数:3 000
定 价:22.00 元

前　言

高职教育必须“以就业为导向”，这就要求职业教育在教学内容上，注重培养学生的综合素质，既保证掌握基本理论，更注重加强实际能力的培养，同时，本教材在结构及内容安排上还充分考虑到与职业技能鉴定相结合。

本教材的主要内容有数字电路基础、集成逻辑门电路、组合逻辑电路、触发器、时序逻辑电路、半导体存储器和可编程逻辑器件、脉冲信号的产生与整形、数模和模数转换器。在内容上，以基本分析方法为重点，以集成电路的应用为目的，注重集成电路的实用性。对一些较深内容，仅作为拓展阅读部分，以便对学有余力的同学，进一步扩大其知识面。本教材在结构上，每章都配有习题，许多章节还安排有技能实训，书后还安排综合实训，以提高学生的实际操作能力和综合能力。

本书由杨林国主编，丁向荣任副主编。第一、四章由吴金权编写；第二、八章由李碧红编写；第三章由郝志廷编写；第五、七章由丁向荣编写；杨林国编写第六章及综合实训，并统稿全书。

本书由王彪先生主审，他对书稿提出了许多宝贵的意见和修改建议，在此深表谢意。

由于编写的时间仓促和作者的水平所限，书中疏漏和错误在所难免，恳请读者批评指正。

编　者

目　录

第一章　数字电路基础

在时间上和幅度上都不连续的信号称为数字信号，用于传递和处理数字信号的电子电路称为数字电路，数字电路广泛应用于各个领域。数字电路也称逻辑电路，主要研究电路输入信号和输出信号之间的逻辑关系。逻辑代数是分析和设计逻辑电路不可缺少的数学工具。

本章讨论的内容为数字电路的计数体制、逻辑代数、逻辑函数及其化简等。

第一节　数字电路中的数制与码制

数字电路中经常遇到计数问题。数制是计数的方法，即多位数码中每一位的构成方法以及从低位到高位的进位规则，最常见的数制是十进制。而在数字系统中进行数字信号的运算和处理时，广泛采用的则是二进制、八进制和十六进制。本节将介绍几种常见数制的表示方法以及不同数制相互间的转换方法和几种常见的二－十进制码（简称 BCD 码）。

一、数　制

（一）十进制数

十进制数是人们最常用的以10为基数的计数体制。它采用0～9十个基本数码，任何一个十进制数都可以用上述十个数码按一定规律排列起来表示，其计数规律是“逢十进一”。

例如，2 305 可写为：$2305 = 2\times10^3+3\times10^2+0\times10^1+5\times10^0$。

由上式可见，十进制数的特点是：

（1）基数是10。基数即计数制中所用到数码的个数。十进制数中的每一位必定是0～9十个数码中的一个。

（2）计数规律是“逢十进一”。

（3）同一数码处于不同的位置时，它代表的数值是不同的，即不同的数位有不同的位权。

上述表示方法也可扩展到小数，但小数点右边的各位数码要乘以基数的负幂次。例如，43.12 表示为：$43.12=4\times10^1+3\times10^0+1\times10^{-1}+2\times10^{-2}$。

一般地，任何一个十进制数 N 都可以表示为：

$$(N)_{10}=(a_{n-1}a_{n-2}\cdots a_1a_0.\,a_{-1}a_{-2}\cdots a_{-m})_{10}$$

这种表示方法称为并列表示法。

对于一个十进制数来说，小数点左边的数码，位权依次是 $10^0,10^1,10^2,\cdots$；小数点右边的数码，位权分别为 $10^{-1},10^{-2},10^{-3},\cdots$。广义来讲，任意一个十进制数 N 所表示的数值，等于其各位加权系数之和，可表示为：

$$(N)_{10}=\sum_{i=-m}^{n-1}K_i\times10^i$$

其中，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，K_i 表示各个数字符号。这种表示方法称为多项式表示方法或按权展开式。式中，下角标 10 表示 N 是十进制数，也可以用字母 D 来代替数字“10”。

（二）二进制数

在数字电路中，常用二进制来表示数并进行运算。采用二进制具有以下优点：

（1）二进制的基数为2，只有0和1两个数字符号，容易用元器件两个不同的物理状态来表示。

（2）二进制运算规则简单，其进位规则是"逢二进一"，便于进行算术运算。

（3）采用二进制来表示数可以节省设备，其运算逻辑电路的设计也比较方便。

任何一个二进制数 N 可以表示为：

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

利用上式可以将任何一个二进制数转换为十进制数。式中，下角标2表示 N 是二进制数，也可以用字母B来代替数字"2"。

【例1-1】 将二进制数110.011转换为十进制数。

解 $(110.011)_2=(1\times2^2+1\times2^1+0\times2^0+0\times2^{-1}+1\times2^{-2}+1\times2^{-3})_{10}=(6.375)_{10}$

（三）八进制数

二进制数的缺点是当位数很多时不便于书写和记忆，容易出错。因此，在数字电路的资料中通常采用八进制和十六进制。

八进制的基数为8，采用0~7这8个数字符号表示，进位规则为"逢八进一"。任何一个八进制数 N 可以表示为：

$$(N)_8=\sum_{i=-m}^{n-1}K_i\times 8^i$$

式中，下角标8表示 N 是八进制数，也可以用字母O来代替数字"8"。一般为了防止手写"O"和"0"混淆，也有用"Q"来代替数字"8"。利用上式可以将任何一个八进制数转换为十进制数。

【例1-2】 将八进制数746.5转换为十进制数。

解 $(746.5)_8=(7\times8^2+4\times8^1+6\times8^0+5\times8^{-1})_{10}=(486.625)_{10}$

（四）十六进制数

十六进制数的基数为16，采用0~9、A~F这16个数字符号表示，其中字母A、B、C、D、E、F分别代表十进制数10、11、12、13、14、15，进位规则为"逢十六进一"。任何一个十六进制数 N 可以表示为：

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

式中，下角标16表示 N 是十六进制数，也可以用字母H来代替数字"16"。利用上式可以将任何一个十六进制数转换为十进制数。

【例1-3】 将十六进制数3AE.8转换为十进制数。

解 $(3AE.8)_{16}=(3\times16^2+10\times16^1+14\times16^0+8\times16^{-1})_{10}=(942.5)_{10}$

二、数制转换

（一）二进制数、八进制数、十六进制数转换为十进制数

只要将二进制数、八进制数、十六进制数按权展开，求出其各位加权系数之和，即得相应的十进制数。

（二）十进制数转换为二进制数、八进制数、十六进制数

将十进制整数转换为二进制数、八进制数、十六进制数一般采用基数除法，也称除基取余

法。转换步骤如下：

第一步：把给定的十进制数$(N)_{10}$除以相应进制的基数，取出余数，即为最低位数的数码K_0。

第二步：将前一步得到的商再除以相应进制的基数，再取得余数，即得次低位数的数码K_1。

以下各步类推，直到商为0为止，最后得到的余数即为最高位数的数码K_{n-1}。

【例1-4】 将$(44)_D$分别转换成二进制数、八进制数和十六进制数。

解

2	44	余数	低位
2	22	$0=K_0$	
2	11	$0=K_1$	
2	5	$1=K_2$	
2	2	$1=K_3$	
2	1	$0=K_4$	
	0	$1=K_5$	高位

所以，$(44)_D=(101100)_B$

8	44	余数	低位
8	5	$4=K_0$	
	0	$5=K_1$	高位

所以，$(44)_D=(54)_O$

16	44	余数	低位
16	2	$12=K_0$	
	0	$2=K_1$	高位

所以，$(44)_D=(2C)_H$

将十进制小数转换为二进制数、八进制数、十六进制数一般采用基数乘法，也称为乘基取整法。具体方法为：不断用相应的基数乘要转换的十进制小数，将每次所得的整数（0或1），依次记为$K_{-1}, K_{-2}, K_{-3}, \cdots$直到小数为0或达到转换所要求的精度为止，然后将所得的整数从高到低读出即可。

【例1-5】 将十进制小数0.6875分别转换为二进制数、八进制数和十六进制数。

解

$0.6875 \times 2 = 1.3750$	整数 $1=K_{-1}$	高位
$0.3750 \times 2 = 0.7500$	$0=K_{-2}$	
$0.7500 \times 2 = 1.5000$	$1=K_{-3}$	
$0.5000 \times 2 = 1.0000$	$1=K_{-4}$	低位

所以，$(0.6875)_D=(0.1011)_B$

0.6875		
× 8	整数	高位
5.5000	$5=K_{-1}$	↓
0.5000		
× 8		
4.0000	$4=K_{-2}$	低位

所以，$(0.6875)_D=(0.54)_Q$

0.6875	
× 16	整数
11.0000	$11=K_{-1}$

由于$(11)_D=(B)_H$，所以$(0.6875)_D=(0.B)_H$

(三)二进制数与八进制数之间的转换

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，见表1－1。

表1－1　八进制与二进制的对应关系

八进制	0	1	2	3	4	5	6	7
二进制	000	001	010	011	100	101	110	111

利用这种对应关系，可以很方便地在八进制与二进制之间进行数的转换。因此，二进制转换为八进制的方法是：以小数点为界，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。

【例1－6】 将二进制数11101110.0101转换为八进制数。

解

011	101	110	.	010	100
↓	↓	↓		↓	↓
3	5	6	.	2	4

所以，$(11101110.0101)_B=(356.24)_Q$

将八进制数转换为二进制数，只要将每1位八进制数用3位二进制数表示即可。

【例1－7】 将八进制数274.36转换为二进制数。

解

2	7	4	.	3	6
↓	↓	↓		↓	↓
010	111	100	.	011	110

所以，$(274.36)_Q=(10111100.01111)_B$

(四)二进制数与十六进制数之间的相互转换

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种不同组合，所以十六进制与二进制之间有简单的对应关系，见表1－2。

表 1－2　十六进制与二进制的对应关系

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

利用这种对应关系,可以很方便地在十六进制与二进制之间进行数的转换。

因此,二进制转换为十六进制的方法是:以小数点为界,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 4 位分成一组,头尾不足 4 位的补 0,然后将每组的 4 位二进制数转换为 1 位十六进制数。

【例 1－8】 将二进制数 1101101101.0100101 转换为十六进制数。

解

$$\begin{array}{cccccc} \underline{\mathbf{0011}} & \underline{\mathbf{0110}} & \underline{\mathbf{1101}} & . & \underline{\mathbf{0100}} & \underline{\mathbf{1010}} \\ \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\ \mathbf{3} & \mathbf{6} & \mathbf{D} & . & \mathbf{4} & \mathbf{A} \end{array}$$

所以,$(1101101101.0100101)_B = (36D.4A)_H$

【例 1－9】 将十六进制数 8FB. CE 转换为二进制数。

解

$$\begin{array}{cccccc} \mathbf{8} & \mathbf{F} & \mathbf{B} & . & \mathbf{C} & \mathbf{E} \\ \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\ \underline{\mathbf{1000}} & \underline{\mathbf{1111}} & \underline{\mathbf{1011}} & . & \underline{\mathbf{1100}} & \underline{\mathbf{1110}} \end{array}$$

所以,$(8FB.CE)_H = (100011111011.1100111)_B$

三、码　制

数字系统中常用 0 和 1 组成的二进制数码来表示数值的大小,或采用一定位数的二进制数码来表示各种文字、符号信息,这个特定的二进制码称为代码。建立这种代码与文字、符号或特定对象之间的一一对应的关系称为编码。

(一) 二-十进制编码(BCD 码)

二-十进制编码是指用 4 位二进制数来表示 1 位十进制数的编码方式,简称 BCD 码。BCD 码有很多种形式,常用的有 8421 码、余 3 码、2421 码、5421 码等,如表 1－3 所示。

表 1－3　常用 BCD 码

十进制数	8421 码	余 3 码	2421 码	5421 码
0	0000	0011	0000	0000
1	0001	0100	0001	0001
2	0010	0101	0010	0010
3	0011	0110	0011	0011
4	0100	0111	0100	0100

续表

十进制数	8421 码	余 3 码	2421 码	5421 码
5	0101	1000	1011	1000
6	0110	1001	1100	1001
7	0111	1010	1101	1010
8	1000	1011	1110	1011
9	1001	1100	1111	1100
权	8421		2421	5421

1. 8421 码

在 8421 码中，10 个十进制数字符号与自然二进制数一一对应，即用二进制数的 0000 ~ 1001 来分别表示十进制数的 0 ~ 9。8421 码是一种有权码，各位的权从左到右分别为 8、4、2、1，所以根据代码的组成便可知道代码所代表的值。8421 码与十进制数之间的转换只要直接按位转换即可。例如：

$$(901.73)_D = (1001\ 0000\ 0001\ .\ 0111\ 0011)_{BCD}$$

8421 码只利用了 4 位二进制数的 16 种组合中的 0000 ~ 1001，其余 6 种组合 1010 ~ 1111 是无效的。从 16 种组合中选取 10 种不同的组合方式，可以得到其他二-十进制码，如 2421 码、5421 码、余 3 码等。

2. 余 3 码

余 3 码由 8421 码加 3(0011)得来的，这是一种无权码。一个十进制数用余 3 码表示时，只要按位表示成余 3 码即可。例如：

$$(83.06)_D = (1011\ 0110\ .\ 0011\ 1001)_{\text{余3码}}$$

3. 2421 码

2421 码也是一种有权码，各位的权从左到右分别为 2、4、2、1。需要指出的是，2421 码的编码方案不止一种，表 1 -3 中给出的只是其中的一种方案。

4. 5421 码

5421 码也是一种有权码，各位的权从左到右分别为 5、4、2、1。

(二)可靠性编码

代码在形成和传输过程中，难免因干扰而发生错误。为了尽可能减少错误的发生，或者在错误发生后能及时发现和纠正，在工程应用中普遍采用可靠性编码技术。格雷码、奇偶校验码是常用的简单可靠性编码。

1. 格雷码

格雷码有多种编码形式，但所有的格雷码都有一个共同的特点：即从一个代码变为相邻的另一个代码时只有 1 位发生变化。表 1 -4 给出了一种典型格雷码与十进制码及二进制码的对应关系。

表 1-4　典型格雷码与十进制及二进制码的对应关系

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

由表 1-4 可以看出，不仅两个相邻的格雷码之间只相差 1 位数码，而且这个 4 位二进制码的首、尾格雷码之间也只相差 1 位数码，所以格雷码又称循环码。格雷码虽不直观，但可靠性高，在输入、输出等场合应用广泛。

格雷码也可以作二-十进制编码，表 1-5 给出了十进制的两种格雷码。其中修改格雷码也具有循环性，即十进制数的首、尾两个数(0 与 9)的格雷码也只有 1 位不同。

表 1-5　十进制数的两种格雷码

十进制数	典型格雷码	修改格雷码	十进制数	典型格雷码	修改格雷码
0	0000	0010	5	0111	1100
1	0001	0110	6	0101	1101
2	0011	0111	7	0100	1111
3	0010	0101	8	1100	1110
4	0110	0100	9	1101	1010

2. 奇偶校验码

二进制信息在传送、存储过程中，可能会发生错误，即有的 1 错成 0，或者有的 0 错成 1。奇偶校验码是一种能检查出这类错误的可靠性编码。

奇偶校验码由信息位和校验位两部分组成。信息位就是传送的信息本身，可以是位数不限的二进制代码。例如，并行传送 8421BCD 码，信息位就是 4 位；校验位是根据规定算法求得并附加在信息后的冗余位。

奇偶检验码分为奇校验和偶校验两种。校验位产生的规律是：对于奇校验，若信息位中有奇数个“1”，则校验位为“0”，若信息位中有偶数个“1”，则校验位为“1”；对于偶校验，若信息位中有奇数个“1”，则校验位为“1”，若信息位中有偶数个“1”，则校验位为“0”。也即通过调节校验位的“0”或“1”，使得整个代码中“1”的个数恒为奇数或者恒为偶数。表 1-6 为 8421BCD 码的奇校验码和偶校验码。

表 1-6 8421BCD 码的奇校验码和偶校验码

十进制数	奇校验码		偶校验码	
	信息位	校验位	信息位	校验位
0	0000	1	0000	0
1	0001	0	0001	1
2	0010	0	0010	1
3	0011	1	0011	0
4	0100	0	0100	1
5	0101	1	0101	0
6	0110	1	0110	0
7	0111	0	0111	1
8	1000	0	1000	1
9	1001	1	1001	0

接收方对收到的加有校验位的代码进行校验，若信息位和校验位中“1”的个数的奇偶性符合约定的规则，则认为信息没有发生错误，否则可以确定信息已经出错。

奇偶校验码算法简单，实现容易，在计算机中有着广泛的应用。但奇偶校验码只能发现代码 1 位(或奇数位)出错，不能发现 2 位(或偶数位)出错。由于 2 位或 2 位以上出错的概率相当小，所以奇偶校验码用来检测代码在传送过程中的错误是相当有效的。奇偶校验码只能发现 1 位出错，但不能定位错误，因而也就不能纠正错误。

第二节　逻 辑 函 数

逻辑代数又称布尔代数，是分析和研究逻辑电路的数学工具，是学习数字电路的基础。逻辑代数所研究的内容是逻辑函数与逻辑变量之间的关系。应用逻辑代数可以将复杂函数式进行恒等变换，以便得到合理的逻辑电路。

一、逻辑变量与逻辑函数

(一)逻辑变量

自然界中，许多现象总是存在着对立的两种状态，为了描述这种相互对立的逻辑关系，往往采用仅有两个取值的变量来表示，这种二值变量就称为逻辑变量。例如，电位的高或低、开关的开或关、晶体管的饱和或截止等现象，都可以用逻辑变量来表示。

逻辑变量和普通代数中的变量一样，可以用字母 $A,B,C\cdots X,Y,Z$ 等来表示。但逻辑变量表示的是事物的两种对立的状态，只允许取两个不同的值，分别是逻辑 0 和逻辑 1。

(二)逻辑函数

在数字逻辑电路中，如果输入变量 $A,B,C\cdots$ 的取值确定后，输出变量 Y 的值也被唯一确定了，则称 Y 是 $A,B,C\cdots$ 的逻辑函数。逻辑函数的一般表达式可以写为：

$$Y=f(A,B,C\cdots)$$

也就是说,如果某逻辑变量 Y 是由其他逻辑变量 $A,B,C\cdots$ 经过有限个基本逻辑运算确定的,则 Y 就是 $A,B,C\cdots$ 的逻辑函数。

在逻辑代数中,逻辑函数和逻辑变量一样,都只有逻辑 0 和逻辑 1 两种取值,但要注意它们没有大小之分,不同于普通代数中的 0 和 1。

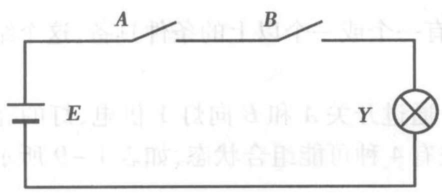
二、基本逻辑关系

逻辑关系是指一定的因果关系,即条件和结果的关系。最基本的逻辑关系只有与逻辑、或逻辑和非逻辑三种,所以逻辑代数中变量的运算也只有与运算、或运算和非运算三种基本逻辑运算。其他任何复杂的逻辑运算都可以用这三种基本逻辑运算来实现。

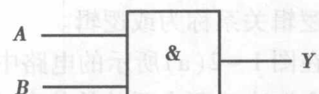
(一)与逻辑和与运算

只有当决定一件事情结果的所有条件全部具备时,这个结果才会发生,这样的逻辑关系称为与逻辑。

例如,在图 1-1(a)所示的电路中,电源 E 通过开关 A 和 B 向灯 Y 供电,灯的亮或灭这两种状态,完全取决于两个开关的状态。两个开关共有 4 种可能组合状态,如表 1-7 所示。



(a)电路图



(b)国标符号

图 1-1　与运算的例子和逻辑符号

表 1-7　图 1-1(a)所示电路的功能表

开关 A	开关 B	灯 Y
断开	断开	灭
断开	闭合	灭
闭合	断开	灭
闭合	闭合	亮

如果用二元常量 0 和 1 来表示图 1-1(a)所示电路的逻辑关系,把开关 A、B 作为输入变量并用 A、B 来表示,而把灯 Y 作为输出变量并用 Y 来表示,且用 0 表示开关断开和灯灭,用 1 来表示开关闭合和灯亮,则可以得到如表 1-8 所示的表格。这种用字母表示开关和电灯的过程称为设定变量,用二元常量 0 和 1 表示开关和电灯状态的过程称为状态赋值,经过状态赋值得到的反映开关状态和电灯亮灭之间所有可能逻辑关系的表格称为逻辑真值表,简称真值表。

由表 1-8 可知,Y 与 A、B 之间的关系是:只有当 A 和 B 都是 1 时,Y 才为 1;否则 Y 为 0。这一关系可用逻辑表达式表示为:

$$Y=A\cdot B$$

式中小圆点"·"表示 A、B 的与运算,与运算又叫逻辑乘,通常"·"可以省略。上式读作

"Y 等于 A 与 B",或者"Y 等于 A 乘 B"。由与运算的逻辑表达式 $Y=A\cdot B$ 和表 1-8 所示的真值表,可知与运算的规律是:

$$0\cdot 0=0 \quad 0\cdot 1=1 \quad 1\cdot 0=0 \quad 1\cdot 1=1$$

与运算除用真值表和逻辑表达式表示外,还可以用逻辑符号表示,如图 1-1(b)所示。

表 1-8　图 1-1(a)所示电路的真值表

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

(二)或逻辑和或运算

在决定一件事情结果的所有条件中,只要有一个或一个以上的条件具备,这个结果就会发生,这样的逻辑关系称为或逻辑。

例如,在图 1-2(a)所示的电路中,电源 E 通过开关 A 和 B 向灯 Y 供电,灯的亮或灭这两种状态,完全取决于两个开关的状态,两个开关有 4 种可能组合状态,如表 1-9 所示。设定变量并经赋值后,可得真值表如表 1-10 所示。

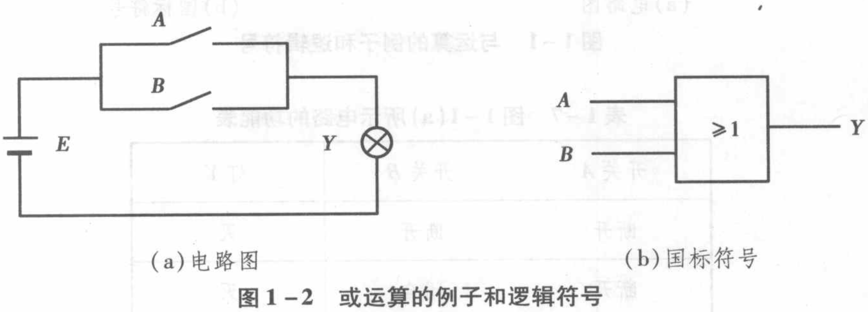


图 1-2　或运算的例子和逻辑符号

表 1-9　图 1-2(a)所示电路的功能表

开关 A	开关 B	灯 Y
断开	断开	灭
断开	闭合	亮
闭合	断开	亮
闭合	闭合	亮

表 1－10　图 1－2(a)所示电路的真值表

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

由表 1－10 可知，Y 与 A、B 之间的关系是：只要 A、B 当中有 1 个或全是 1 时，Y 就为 1；若 A 和 B 全为 0，则 Y 为 0。这一关系可用逻辑表达式表示为：

$$Y=A+B$$

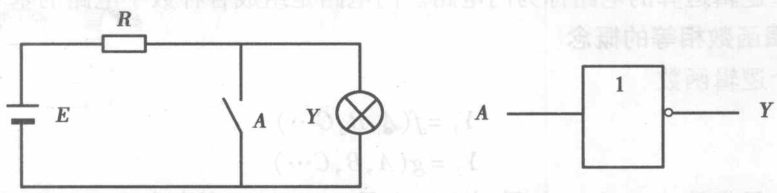
式中符号“＋”表示 A、B 的或运算，或运算又叫逻辑加。上式读作“Y 等于 A 或 B”，或者“Y 等于 A 加 B”。由或运算的逻辑表达式 $Y=A+B$ 和表 1－10 所示的真值表，可知或运算的规律是：

$$0+0=0 \qquad 0+1=1 \qquad 1+0=1 \qquad 1+1=1$$

或运算也可以用逻辑符号表示，如图 1－2(b)所示。

(三)非逻辑和非运算

在某一事件中，如果条件和结果的状态总是相反，这样的逻辑关系称为非逻辑。非就是相反，就是否定。例如，在图 1－3(a)所示的电路中，当一个开关 A 和一个指示灯 Y 并联在电路中，其逻辑功能表和真值表分别如表 1－11 和表 1－12 所示。



(a)电路图　　(b)国标符号

图 1－3　非运算的例子和逻辑符号

表 1－11　图 1－3(a)所示电路的功能表

开关 A	灯 Y
断开	亮
闭合	灭

表 1－12　图 1－3(a)所示电路的真值表

A	Y
0	1
1	0

由表 1－12 可知，Y 与 A 之间的关系是：当 $A=0$ 时，$Y=1$；而当 $A=1$ 时，则 $Y=0$。这一关系可用逻辑表达式表示为：

$$Y=\overline{A}$$

式中字母 A 上方的符号“—”表示 A 的非运算或者反运算。上式读作“Y 等于 A 非”，或者“Y 等于 A 反”。显然，非运算的规律是：

$$\overline{0}=1 \qquad \overline{1}=0$$