机电一体化工程专业电工学之三

数字集成电路应用基础

吴建强　主编

航空工业出版社

机电一体化工程专业电工学之三

数字集成电路应用基础

吴建强　主编

航空工业出版社

1994

（京）新登字161号

内 容 提 要

这是一套把电工技术、电子工程、机械工程、传感技术、计算机和自动控制等高新技术有机地构成机电一体化的电工学教材，该书是这套教材中的第三册，考虑到目前数字电路的一个主要发展方向——集成化，本教材除了包括数字电路的基础知识外，还以中小规模集成电路为主来组织内容，并适当介绍大规模集成电路。全书注重集成电路的外部特性和应用，拓宽了数字电路的内容。为了帮助读者全面理解书中的主要内容，各章节均附有练习与思考题。

本书适合机械工程、仪器仪表和化工类各专业的本科、专科学生阅读，也可供从事电工、电子技术的工程技术人员自学参考。

机电一体化工程专业电工学之三
数字集成电路应用基础
吴建强　主编

航空工业出版社出版发行
（北京市安定门外小关东里14号）
—邮政编码：100029—
全国各地新华书店经售
煤炭工业出版社印刷厂印刷

1994年3月第1版　1994年3月第1次印刷
开本：787×1092 1/16　印张：13.625
印数 1—2200　字数：356千字
ISBN 7-80046-693-0

G·128
定价：6.90元

前　　言

在电子技术飞速发展的今天，集成电路、微型计算机和传感器等技术迅速发展，影响所及十分深远。工程中的数控技术、工业机器人、FMS等生产手段不断冲击旧机器组成的生产线、车间乃至工厂，使仪表、机械工程领域发生了根本性变化。为了适应新的形势，在电工学教学中，我们深感需要一套把电工技术、电子工程、机械工程、传感器技术、计算机和自动控制等高新技术有机地构成机电一体化的电工学教材。本书就是在这一思想的指导下，根据原航空航天工业部部属院校电工学课程协作组1989年12月制订的编写大纲编写的。本书是这套教材（共四册）中的第三册。

在编写教材时，考虑到目前数字电路的一个主要发展方向——集成化，本教材除了包括数字电路的基础知识外，还以中小规模集成电路为主来组织内容，并适当介绍大规模集成电路，以期与国内外电子技术的发展相适应，并注重集成电路的外部特性和应用，拓宽了数字电路的内容，以适应机电一体化的需要。

这本教材适用于机械工程、仪器和化工类各专业，授课时数为50学时（不包括实验）。

为了帮助读者理解书中的主要内容，各章中的各节均附有练习与思考题，各章均有小结与习题，以便读者练习。

本书由哈尔滨工业大学电工学教研室吴建强同志主编，其中第五章由吴建强执笔；第二章、第六章由沈阳航空工业学院张起时、刘文杰同志执笔；第三章、第七章由哈尔滨工业大学毕淑娥、杨士彦同志执笔；第一章、第四章由北京航空航天大学王邦柱同志执笔。最后由吴建强同志修改写成。全书由哈尔滨船舶工程学院电子工程系张保郁教授进行了认真的审阅，提出了详细的修改意见，在此向他以及其他关心过本书的同志表示衷心地感谢。

由于编者水平有限，书中难免有不妥或错误之处，望读者批评指正。

编　者

1993年2月20日

目　　录

第一章　数字电路的基本知识

本章主要介绍数制、码制和逻辑代数及其运算。

数字电路工作于离散时序。电路的输入信号和输出信号都是由“0”和“1”表征的二值量数字信号。二值量就是两种取值的变量，由于它可以表示出事物的两种截然不同的状态，故可用它表示事物的逻辑关系，而这种逻辑关系在数字电路的工作过程中总是在一定的条件下体现一种因果判断关系，这种因果可用一定的逻辑函数式（逻辑方程）来表示。逻辑函数式的转换和运算需要采用二进制计数法，这种二进制数同十进制数一样，可以作各种数字运算。由此可见，在研究处理数字信号和分析设计数字电路的过程中，经常要使用二进制数和逻辑代数的一些基本运算方法。所以在学习数字电路之前，我们应对有关数的进制和运算以及逻辑代数等基本知识有所了解，以便为后面的学习打好基础。

此外，在数字电路中，有时常常要直接处理带符号的数、十进制数、字母以及专门符号等。这就要用代码来表示。而这些代码往往都是以二进制数的编码来表示的。所以，学习一些有关码制的知识，对后面内容的学习和理解，是有很大帮助的。

1.1　数制和码制

一、数制

任何用于计数和完成算术运算的符号和字符的集合都可以认为是一种数制。数制由表达数量所需要的不同数字的参数来表征。对十进制来说，需要十个数字：0，1，2，3，4，5，6，7，8和9。为了表明一个给定的量，任何一个数字都可以根据需要重复若干次，例如3333。

1．十进制

十进制数我们是非常熟悉的，现在我们来考察十进制数2315。这个数能分解成：

$$2000+300+10+5=2315$$

而

$$2000=2\times10^3=2\times1000=2000$$

$$300=3\times10^2=3\times100=300$$

$$10=1\times10^1=1\times10=10$$

$$5=5\times10^0=5\times1=5$$

$$\frac{+}{2315}$$

（注意：$10^0=1$，因为任何数的零次幂等于1）。

数10被叫做这个数制的基（或称为底数）。这个基等于该数制里不同字符的个数。因为字符本身从0开始，所以基总是比该数制里所用的最大字符多1个单位。数本身并不一定能表明它的基是什么。例如符号1010可能表示一个十进制数，也可能表示一个二进制数、八进

制数或者十六进制数。为了避免混淆，不是十进制写的数应该用基做下标。例如在八进制中应写成$[1010]_8$。

对于十进制里的0，应当记住：零乘以任何数结果还是等于零。考察数106。

$$106=(1\times10^2)+(0\times10^1)+(6\times10^0)=106$$

在上述的数2315中，根据两方面的因素，给每一位数字赋值。这两个因素是：（1）数字基本值，即它本身作为个数所表达的值。（2）权值，它由该数字在数中的位置决定。在十进制中，其权值是1，10，100，1000等。不管用什么数制，具有最高权值的位在左边，而具有最低权值的位在右边。不难发现，各位数的权都是基数的幂。显然，某位数的数值等于该位的数字基本值和权的乘积，人们常把这种乘积称为加权数字基本值。

一般地说，几位十进制正整数$[N]_{10}$的表达式可写成：

$$[N]_{10}=a_{n-1}\times10^{n-1}+a_{n-2}\times10^{n-2}+\cdots+a_1\times10^1+a_0\times10^0=\sum_{i=0}^{n-1}a_i\times10^i \qquad (1\text{-}1)$$

式中a_0、$a_1\cdots a_{n-1}$为各位数的数字基本值（a_i是第i位的数字基本值），它可以取0～9十个数字符中的任意一个；10^0、$10^1\cdots10^{n-1}$为各位数的权。

2．二进制

二进制数的基为2。它只需两个不同的数字，0和1，来表示任何数量。二进制广泛地用于计算和数字电路中。二进制数是逢二进位的，所以二进制数的权为2的幂。n位二进制正整数$[N]_2$表达式可写成：

$$[N]_2=a_{n-1}\times2^{n-1}+a_{n-2}\times2^{n-2}\cdots+a_1\times2^1+a_0\times2^0=\sum_{i=0}^{n-1}a_i\times2^i \qquad (1\text{-}2)$$

式中a_0、$a_1\cdots a_{n-1}$为数字基本值，可取0和1的任何一个；2^0，2^1，…，2^{n-1}为各位数的权。

二进制数只有两个数字符号，不仅进行算术运算简单，而且在电路上实现起来也比较容易，所以在数字电路中广泛采用二进制。

例1-1 写出数值为$[10110101]_2$与之对应的十进制数。

解 由二进制数的一般表达式可知，只要将它们按权展开，各位加权数字基本值相加，即可得到对应的十进制数。

$$\begin{aligned}[10110101]_2&=1\times2^7+1\times2^5+1\times2^4+1\times2^2+1\times2^0\\&=128+32+16+4+1=[181]_{10}\end{aligned}$$

从上例可知，三位十进制数$[181]_{10}$就用了八位二进制数$[10110101]_2$表示。可以想象，如果数值再大，位数会更多。既难记忆，又不便读写。为此，在实际应用中，又经常使用八进制和十六进制，有时还会遇到二-十进制。

3．八进制、十六进制和二-十进制

（1）八进制

在八进制中，基为8，它有0，1，2，3，4，5，6，7八个数字符号，它是逢八进位的，各位数的权是8的幂。n位八进制正整数的表达式可写成：

$$[N]_8=a_{n-1}\times8^{n-1}+a_{n-2}\times8^{n-2}+\cdots+a_1\times8^1+a_0\times8^0=\sum_{i=0}^{n-1}a_i\times8^i \qquad (1\text{-}3)$$

例1-2 求三位八进制数$[N]_8=[221]_8$所对应的十进制数的值。

解 $(221)_8 = 2\times 8^2 + 2\times 8^1 + 1\times 8^0 = 128 + 16 + 1 = [145]_{10}$

在十进制数中，权值为十的幂，如10^3，10^2等。而八进制数涉及的权值为8^3，8^2等。因为8是2的整数次幂（$8=2^3$），因此从二进制到八进制以及从八进制到二进制的转换，是个很简单的过程。如：

八进制到二进制	二进制到八进制
$[4\quad 5\quad 7]_8$	$[100\ 101\ 111]_2$
$[100\ 101\ 111]_2$	$[4\quad 5\quad 7]_8$

（2）十六进制

在十六进制中，基为16。它有0，1，2，3，4，5，6，7，8，9，A，B，C，D，E，F十六个数字符号。它是逢十六进位的，各位数的权为16的幂。n位十六进制正整数的一般表达式为：

$$[N]_{16} = \sum_{i=0}^{n-1} a_i \times 16^i \tag{1-4}$$

例1-3 一个十六进制正整数$[3BF]_{16}$所对应的十进制数是多少？

解 $[3BF]_{16} = 3\times 16^2 + 11\times 16^1 + 15\times 16^0 = [959]_{10}$

16也是2的整数次幂（$16=2^4$），所以一个十六进制的数字其值和四位二进制数字相等：

十六进制到二进制	二进制到十六进制
$[D\quad 8\quad 7]_{16}$	$[1101\ 1000\ 0111]_2$
$[1101\ 1000\ 0111]_2$	$[D\quad 8\quad 7]_{16}$

（3）二-十进制

在数字电路中，经常还采用二-十进制。这是由于计算机系统内应用的是二进制，而人们又习惯用十进制表示法，所以将十进制数的每一位分别用四位二进制数表示出来，例如$[39]_{10}$表示为$[0011]_2$ $[1001]_2$，并将两组四位二进制数按原来十进制数的顺序排列起来，所构成的就是二-十进制数，简称BCD进制。上例中：

$[39]_{10} = [0011\ 1001]_{BCD}$。可见在这种进制中，每组四位数是二进制，而组与组之间是十进制。

为了表示以上各种进制数之间的数值关系，列出表1-1常用计数进制表。

二、码制

数字电路中，信息的传递通常是经由电平高低或脉冲有无来实现的。电路中的这两种相反状态正好与数制中的二进制数的状况相吻合。二进制数中的“0”和“1”两个数字可与电路中的两种状态一一对应，因此数字电路中大量使用二进制数是很自然的了。

在日常生活中人们常用十进制数，有时也使用八进制或十六进制等，使用这些数制或其它的文字符号时，根本不可能简单地用两种状态加以描述。在数字电路中为了要表示它们，就要用一组二进制数与之对应。这组二进制数就是代码。一般说来，用某种文字、符号或数字表示特定对象的过程，就叫作编码。如上所述，在数字电路中通常采用二进制编码。如果要求表示的对象多，可以用增加二进制代码的位数来解决。一位二进制代码有“0”和“1”两种状态，可以用来表示两个对象：两位二进制数有00，01，10，11四种组合状态，可以表

表 1-1 常用计数进制表

十进制	二进制	八进制	十六进制	二-十进制
0	0	0	0	0000
1	1	1	1	0001
2	10	2	2	0010
3	11	3	3	0011
4	100	4	4	0100
5	101	5	5	0101
6	110	6	6	0110
7	111	7	7	0111
8	1000	10	8	1000
9	1001	11	9	1001
10	1010	12	A	00010000
11	1011	13	B	00010001
12	1100	14	C	00010010
13	1101	15	D	00010011
14	1110	16	E	00010100
15	1111	17	F	00010101
16	10000	20	10	00010110
20	10100	24	14	00100000
32	100000	40	20	00110010

示四个对象。一般说来，如果使用 n 位二进制数代码，它最多可以表示2^n个不同对象。如果已知需要表示的对象为M，可以根据$2^n > M$这一关系式来确定最少需要用的二进制数代码位数 n。

二进制数代码与它所表示的对象之间的对应关系并不是唯一的。例如，为了要表示一位十进制数，由于它有0到9这样10个数字，据$2^n > 10$这一关系式可知，n 最小也应当是 4 。也就是说至少需要四位二进制数代码。至于二进制数代码组合与十进制数字之间的对应关系完全是人为规定的。例如我们可以规定用二进制数代码0000表示十进制数字 0 ，但也可以用二进制代码0011或其它组合来表示它。需要指出的是，从理论上讲，编码虽允许有随意性，但由于这种编码一旦确定，便要用它进行随后的运算与处理，此外人们还要用这些代码交换信息。因此，编码的方式应当是科学的，能被大家所接受的，有的甚至还是标准化了的。下面介绍几种常用的代码。

1、常用的二-十进制编码（binary-coded decimal，BCD码）

用四位二进制数代码表示十进制数字0到9叫做二-十进制编码，常用的二-十进制编码见表1-2。

（1）8421码

在这种编码中，其值为 1 那些位的权值之和就等于它们所表示的十进制数。在这种编码中，由左至右每一位1分别代表其权值8、4、2、1。所以把这种编码称为8421码。由于在8421码中每一位的权都是固定的，所以它是一种恒权代码。

（2）余 3 码

在这种编码中每一组码所表示的二进制数要比它所对应的十进制数多 3 ，故这种码称余 3 代码。余 3 代码不是恒权码。

表 1-2 二-十进制编码表

编码种类 / 十进制数	8421码	余3码	2421码(A)	2421码(B)
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
5	0101	1000	0101	1011
6	0110	1001	0110	1100
7	0111	1010	0111	1101
8	1000	1011	1110	1110
9	1001	1100	1111	1111
权	8421		2421	2421

（3）2421码

表1-2中列出了两种2421码，它们都是恒权码，但所取的状态和编码的顺序不完全一样。

2．循环码

循环码又称反射码或格雷码（Gray code）。表1-3是四位循环码的编码表。由表可见，循环码的编码顺序具有如下特点，即：循环码中每一位码从上到下的排列顺序都是以固定周期进行循环的。如表中右起第一位的循环周期是0110，第二位的循环周期是00111100等。这种编码的主要优点是相邻的两个编码中只有一位状态不同。这在数字电路中是有一定用处的，以后将要提到。循环码是一种变权代码。

表 1-3 循环码编码表

码序	循环码	码序	循环码
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

〔练习与思考〕

1．数的常用进制有哪几种？什么叫基？权值和加权数字值？

2．码制是什么？二进制数代码同二进制数有什么区别？

3．将下列二进制数分别转换成十进制数、八进制数和十六进制数。

$[1011]_2$；$[10101101]_2$；$[110110]_2$；$[1110010001]_2$。

4．指出表1-2和表1-3的编码各有什么特点。

1.2 逻辑代数的基本知识

逻辑代数又称布尔代数，它得名于英国数学家乔治·布尔（George Boole），他在1854年首先提出这种代数的基本概念和性质，并用它作为研究思维逻辑的工具。1938 年克劳德·香农（Claude）开始用它来研究开关电路，自此，逻辑代数与开关数字电路结下了不解之缘。

和普通代数比较起来，虽然逻辑代数也用字母表示变量，但是它比普通代数简单得多。因为它的变量取值非常简单，不是“0”就是“1”，没有第三种值。值得注意的是，逻辑变量的值已不再表示具体数量的大小，它表示的是两种不同的逻辑状态（例如：用“1”和“0”表示是或非，真和假，高和低，有和无，开和关等等）。因此，逻辑代数中的变量就称为逻辑变量。由于逻辑变量本身所具有的特殊性质，所以逻辑代数的运算规则也与普通代数有不同之处，在学习过程中应注意加以区别。

一、三种基本逻辑关系和运算

在逻辑代数中，最基本的逻辑运算只有三种：与运算——逻辑乘法运算；或运算——逻辑加法运算；非运算——逻辑求反运算。下面就分别叙述它们。

1．与逻辑关系和运算

日常事物中往往会有这种情况，要得到某种“结果”，必须同时满足几个“条件”。而这种“条件”和“结果”的关系就是与逻辑关系。为了进一步说明上述关系，结合图1-1（a）中的电路加以说明。

上面说过，逻辑变量只有两种相反的状态，对于这两种状态我们分别取值为“1”或“0”，在图1-1电路中，我们规定开关接通为“1”，断开为“0”；电灯亮为“1”，暗为“0”，并据此进行讨论。

图1-1（a）电路中，开关A和B串联，只有当A和B同时接通时（条件同时满足），电灯Y才亮（得到结果）。电灯的亮暗和开关的通断之间是一种逻辑与的关系。

（a）逻辑“与”；（b）逻辑“或”；（c）逻辑“非”

如果用逻辑变量符号表示上述关系，可用A和B代表开关通断的情况，同时用Y代表灯亮或暗这一结果。这种与的逻辑关系可以写成下面的逻辑变量表达式：

$$Y=A\cdot B \tag{1-5}$$

在逻辑代数中，式（1-5）被称为与运算，也称逻辑乘法运算，其中Y称逻辑积。

为了详细描述与逻辑关系，经常把“条件”和“结果”的各种可能性，列成表格对应地表示出来，见表1-4。这种表虽然能够直观地反映出各种条件下对应的结果，可是书写起来比较麻烦，而且对于不同的事物要列不同的表格。为了书写方便并能通用起见，一般用逻辑变量列表，并将它们真正取值1或0填入表格中，如表1-5所示。这种表称为真值表。由此表不难看出，A、B两个变量有四种取值情况，只有A、B均为1时，Y才为1，也就是$1\cdot1=1$。可见逻辑乘法和普通代数的乘法所得结果是一样的。但是，两者却有本质上的区别。这里1和0不是数量的大小，而是两种相反的逻辑状态。$1\cdot1=1$，表示两个开关都闭合灯才亮。此表也可以代表任意两变量事件的与逻辑关系。可见真值表是表示逻辑关系的重要方法之一。

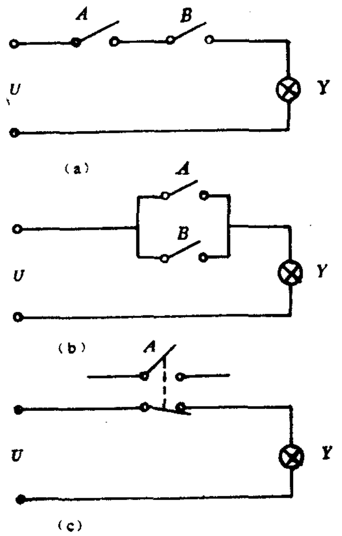

图 1-1 由开关组成的逻辑电路

表 1-4 与逻辑关系表

开关 A	开关 B	灯
断	断	暗
断	通	暗
通	断	暗
通	通	亮

表 1-5 与逻辑真值表

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

此外，与逻辑关系还可以用逻辑符号表示。图1-2是与逻辑符号。

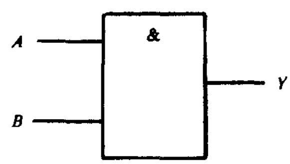

图 1-2 与逻辑符号

2．或逻辑关系和运算

在几个条件中，只要满足一个或一个以上条件，则能得到某一“结果”。这种“条件”和“结果”的关系就是或逻辑关系。

对于图1-1(b)电路，只要闭合任意一个开关或两个开关都闭合，灯都能亮。两个开关的开关状态是“条件”，灯能否亮是结果，这种“条件”对结果来说是或逻辑关系。

用逻辑变量A和B分别代表两个开关的状态。假设开关闭合为1，开关断开为0。用逻辑变量Y代表灯是否亮，假设灯亮为1，灯暗为0。那么在逻辑代数中，这种或逻辑关系可以写成下面逻辑变量表达式

$$Y = A + B \tag{1-6}$$

式(1-6)称为或运算，也称逻辑加法运算，其中Y为逻辑和。

如果将逻辑变量A，B分别代入它们的真正取值1或0，则可得到逻辑变量Y所对应的取值。将结果列成真值表如表1-6所示。从表中可以看出，只要变量A，B中一个为1或两个同时为1，则Y都为1。应当注意逻辑加法和普通二进制数的加法并不完全相同。普通代数二进制加法$1+1=10$。而逻辑加$1+1=1$。前者是数量之和，而后者表示当两个“条件”同时满足时，“结果”能实现，是逻辑关系。

或逻辑也可以用逻辑符号表示，如图1-3所示。

3．非逻辑关系和运算

当一条件满足时，却得不到某一结果，当条件不满足时，才能得到结果。这种“条件”和“结果”的关系就是非逻辑关系。

从图1-1(c)我们可以看出，开关A闭合了，灯却灭了。而开关A断开了，灯却亮了。

如果用逻辑变量A代表开关，假设1为开关闭合，0为开关断开，同时用Y代表电灯，灯

表 1-6 或逻辑真值表

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

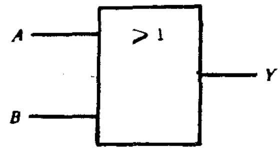

图 1-3 或逻辑符号

亮表示为1，灯灭表示为0。那么在逻辑代数中，常把非逻辑关系写成逻辑变量表达式

$$Y=\overline{A} \tag{1-7}$$

式(1-7)称为非运算，也称求反运算。为了表示非（或称反），我们在变量上加一横，例如$\overline{A}$为变量A的逻辑非，读作A非或A反。

非逻辑的真值表和逻辑符号分别如表1-7和图1-4所示。

表 1-7 逻辑非真值表

A	Y
0	1
1	0

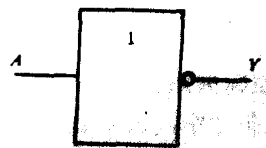

图 1-4 非逻辑符号

二、逻辑函数和逻辑图

三种基本逻辑关系反映在逻辑代数中是三种最基本的逻辑运算。但在实际许多逻辑关系中，只用这三种关系来表示还远远不够，一般是以这三种基本逻辑运算组合起来，以一种更加复杂的运算形式来表示新的逻辑关系。因此，常用逻辑函数来描述这种关系。

1. 逻辑函数

在式(1-5)到(1-7)中，逻辑变量A，B表示某种输入（条件），Y表示输入逻辑变量运算的结果（输出），或者说，Y是变量A，B的逻辑函数，例如，对于式(1-5)，Y就是A，B的与函数，对于式(1-6)，Y就是A，B的或函数。

一般地说，某逻辑变量Y如果是由若干其它逻辑变量A，B，$C\cdots$经过若干个基本逻辑运算确定的，那么Y就称作是A，B，$C\cdots$的逻辑函数。逻辑函数的一般表达式可写成

$$Y=F(A,B,C\cdots) \tag{1-8}$$

例如$Y=\overline{A}\cdot B+A\cdot\overline{B}$就是一个逻辑函数表达式。

2. 逻辑图

所谓逻辑图就是用逻辑符号表示出来的逻辑函数。

在数字电路中，对应于各种逻辑符号，一般都有实现其功能的电路。因此，要完成逻辑电路的设计，必须将逻辑函数的关系转换成逻辑图的形式，以确定电路的结构。可见逻辑图是分析设计数字电路的重要工具。

除去与、或、非三种基本逻辑函数及其逻辑符号外，还有由这些基本函数及其符号组合而成的一些常用复合逻辑函数及其逻辑符号。下面再介绍几种常用的复合逻辑函数及其逻辑符号。

（1）与非逻辑

与非逻辑函数表达式为

$$Y=\overline{A\cdot B} \tag{1-9}$$

表1-8是它的真值表，图1-5是它的逻辑符号。

表 1-8 与非逻辑真值表

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

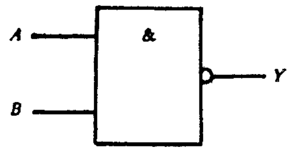

图 1-5 与非逻辑符号

（2）或非逻辑

或非逻辑函数表达式为

$$Y=\overline{A+B} \tag{1-10}$$

表1-9为或非逻辑的真值表。图1-6为或非逻辑符号。

表 1-9 或非逻辑真值表

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

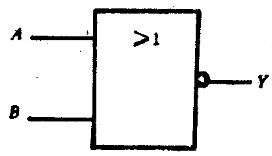

图 1-6 或非逻辑符号

（3）异或逻辑

异或逻辑函数表达式为

$$Y=\overline{A}\cdot B+A\cdot\overline{B} \tag{1-11}$$

表1-10为异或逻辑的真值表。图1-7为异或逻辑符号。

表 1-10 异或逻辑真值表

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

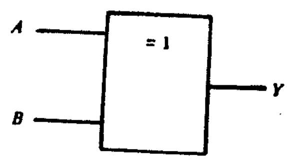

图 1-7 异或逻辑符号

熟悉了常用的逻辑符号，就可以根据逻辑函数式画逻辑图。

由于逻辑代数的运算顺序和普通代数一样，先括号，再算逻辑乘，然后做逻辑加法，所