数字电路与逻辑设计教程

谢声斌　编著

清华大学出版社

数字电路与逻辑设计教程

谢声斌　编著

清华大学出版社
北　京

内容简介

全书共分10章，内容包括数字电路的基础知识，组合逻辑电路的分析与设计、组合集成电路、时序电路的分析与设计、集成时序电路、存储器和可编程逻辑器件、脉冲信号的产生与整形、A/D与D/A转换等。

本书精简扼要，并深入浅出地阐述了数字、逻辑电路的工作原理和分析、设计方法。加强了对目前数字系统中常用的中、大规模集成部件的工作原理和应用的介绍。

本书可作为高等学校计算机、通信类专业的教材，也可作为工科院校通信类、计算机类、无线电类、自动化类等专业的教材，同时也可供有关工程技术人员学习和参考。

图书在版编目(CIP)数据

数字电路与逻辑设计教程/谢声斌编著. —北京：清华大学出版社，2004.10
ISBN 7-302-09222-2

Ⅰ.数…　Ⅱ.谢…　Ⅲ.数字电路　逻辑设计—高等学校—教材　Ⅳ.TN79

中国版本图书馆CIP数据核字(2004)第082219号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编**：100084
社 总 机：010-62770175　　**客户服务**：010-62776969
责任编辑：魏江江
封面设计：杨　兮
印 刷 者：北京密云胶印厂
装 订 者：北京鑫海金澳胶印有限公司
发 行 者：新华书店总店北京发行所
开　　本：185×260　**印张**：15.75　**字数**：390千字
版　　次：2004年10月第1版　2004年10月第1次印刷
书　　号：ISBN 7-302-09222-2/TN·206
印　　数：1～4000
定　　价：21.00元

前　言

本书是根据吉林大学教学大纲并结合近年来的教学实践编写的。在编写过程中尽量做到文字叙述通俗易懂、逻辑性强，内容安排由浅入深、循序渐近、理论联系实际，习题与课程内容相匹配，以便于自学。

本书共分 10 章，除介绍最基础的理论知识外，重点介绍了中、大规模集成电路。

第 1 章～第 3 章介绍了基础知识。内容包括数字与编码、逻辑代数和逻辑门电路。这些知识是进一步研究数字逻辑电路的组成和应用所必备的。

第 4 章是组合逻辑电路。主要介绍了分析和设计方法及常见的各种中规模集成电路的特性及应用。

第 5 章～第 7 章为时序电路。主要介绍了同步和异步电路的分析方法和设计方法。还介绍了各种常见的集成电路。

第 8 章介绍存储器和可编程逻辑器件。可编程器件(PLD)是近十年来发展起来的新型集成电路。一片 PLD 可代替几十、几百甚至上千个逻辑门，是集成电路的重要分支。

第 9 章介绍脉冲信号的产生与整形。

第 10 章介绍 A/D 和 D/A 变换。

全书采用国家标准的图形符号。

本书第 1 章由徐淳宁编写，第 2 章～第 4 章由黄玉兰编写，第 9 章和第 10 章由王凯编写，其余部分由谢声斌编写。全书由谢声斌负责统编定稿。

由于作者学识有限，书中必然存在不足和错误，敬请读者批评指正。

编　者

目　录

第 1 章　数制与编码

1.1　数制

1.1.1　进位计数制

所谓进位计数制是指按进位的方法来进行计数，简称进位制。

在进位计数制中，常常要用“基数”(或称底数)来区别不同的数制。而某进位制的基数就是表示该进位制所用字符或数码的个数。如十进制数有 0～9 共 10 个数码表示数的大小，故其基数为 10。为了区分不同的数制，可在数的下标注明基数。如$(65536)_{10}$表示以 10 为基数的数制，它是每计满 10 便向高位进 1，即“逢十进一”；当基数为 M 时，便是“逢 M 进一”。

1. 十进制数

十进制数是人们经常使用的一种数制，它有 10 个符号 0、1、2、3、4、5、6、7、8、9，我们叫这些数字符号为数码。十进制有 10 个数码，所以它的基数为 10，它的计数规则是“逢十进一”。数的表示法，一般采用位置记数法。每一个数码和数码所在的位置决定了该数的大小，即每一个数码的位置载有该数大小的一个特定值，这个数值称为“位权”。每个位置的“位权”，可以用基数的幂次来确定。在十进制中，整数的位权是 10^0(个位)、10^1(十位)、10^2(百位)等；小数的位权是 10^{-1}(十分位)、10^{-2}(百分位)等。如$(625.49)_{10}$用位置记数法表示如下：

$$(625.49)_{10}=6\times10^2+2\times10^1+5\times10^0+4\times10^{-1}+9\times10^{-2}$$

上式称为按位权展开式。

任意一个十进制数的按权展开式可表示为

$$\begin{aligned}N_{10}&=a_{n-1}\times10^{n-1}+a_{n-2}\times10^{n-2}+\cdots+a_1\times10^1+a_0\times10^0+\\&\quad a_{-1}\times10^{-1}+a_{-2}\times10^2+\cdots+a_{-m}\times10^{-m}\\&=\sum_{i=-m}^{n-1}a_i\times10^i\end{aligned}\tag{1-1}$$

2. 二进制数

进位计数制中最简单的是二进制，它只包括 0 和 1 两个不同的数码，即基数为 2，进位原则是“逢二进一”。二进制数也采用位置记数法，每个位置的位权是 2 的幂。

例如：二进制数 1101.11 的按位权展开式为

$$\begin{aligned}&1\times2^3+1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-2}\\&=8+4+1+0.5+0.25\\&=(13.75)_{10}\end{aligned}$$

二进制数的一般形式为

$$\begin{aligned}(N)_2 &= (b_{n-1}b_{n-2}\cdots b_1 b_0 \cdot b_{-1}b_{-2}\cdots b_{-m})_2 \\ &= (b_{n-1}\times 2^{n-1}+b_{n-2}\times 2^{n-2}+\cdots b_1\times 2^1+b_0\times 2^0+b_{-1}\times 2^{-1}+ \\ &\quad b_{-2}\times 2^{-2}+\cdots+b_{-m}\times 2^{-m})_{10} \\ &= \sum_{i=-m}^{n-1} b_i\times 2^i \end{aligned} \tag{1-2}$$

由上式可知，将二进制数化为十进制数，是把二进制的每一位数字乘以该位的权然后相加得到。实际上只需将为 1 的各位的权相加即可。

在数字系统中常常采用二进制数，这是因为二进制数的基数为 2，它只有 0 和 1 两个数字，运算规则简单，便于电路实现。

二进制数的缺点是，同一个数值用二进制数表示时，比十进制数表示时的位数要多，不便于记忆。为解决二进制数位数较多的问题，常采用以 2 的幂为基数的八进制数、十六进制数（注：二进制、八进制、十进制和十六进制常以 B、O、D 和 H 表示，它们是 Binary、Octal、Decimal 和 Hexadecimal 的字头）。

3. 八进制数

八进制数的基数为 8，用 0～7 八个不同的数码来表示数值，计数规则是“逢八进一”，任意一个八进制数的按位权展开式为

$$N_8 = \sum_{i=-m}^{n-1} a_i\times 8^i \tag{1-3}$$

4. 十六进制数

十六进制数是微机中最常用的一种进位制数，它在数据结构上类似于八进制数，易于与二进制数转换，且比八进制数更能简化数据的输入和显示，因而在计算机系统中得到了广泛的应用。

十六进制数的基数为 16，即由 16 个不同的数码组成，且“逢十六进一”。这种数制中有 16 个不同的数字：除了 0～9 十个数字外，还用字母 A、B、C、D、E、F 分别表示数 10、11、12、13、14、15。例如，将十六进制数 4E9 转换为十进制数：

$$(4E9)_H = 4\times 16^2+14\times 16^1+9\times 16^0 = (1257)_D$$

任意一个十六进制数写成按位权展开式为

$$(N)_H = \sum_{i=-m}^{n-1} a_i\times 16^i \tag{1-4}$$

十六进制、十进制、八进制和二进制数之间的关系如表 1-1 所示。

表 1-1 8421 码、2421 码、余 3 码的编码

二进制码				代码对应的十进制数			
				自然二进制码	二—十进制编码		
B3	B2	B1	B0		8421 码	2421 码	余 3 码
0	0	0	0	0	0	0	冗余码组
0	0	0	1	1	1	1	
0	0	1	0	2	2	2	

续表

二进制码				代码对应的十进制数			
				自然二进制码	二—十进制编码		
B3	B2	B1	B0		8421 码	2421 码	余 3 码
0	0	1	1	3	3	3	0
0	1	0	0	4	4	4	1
0	1	0	1	5	5		2
0	1	1	0	6	6	冗余码组	3
0	1	1	1	7	7		4
1	0	0	0	8	8		5
1	0	0	1	9	9		6
1	0	1	0	10	冗余码组		7
1	0	1	1	11		5	8
1	1	0	0	12		6	9
1	1	0	1	13		7	冗余码组
1	1	1	0	14		8	
1	1	1	1	15		9	

1.1.2　不同进位数制之间的转换

1. 各种进制数转换成十进制数

人们在日常生活中已习惯于使用十进制数，但计算机中的算术运算和逻辑运算却是以二进制数的形式进行的。这样就涉及不同数制之间的转换。

将二进制、八进制、十六进制以及 r 进制数转换为十进制数是用各进制数的按位权展开式求得的，即用

$$N_r = \sum_{i=-m}^{n-1} a_i \times r^i \tag{1-5}$$

a_i——系数，取值范围 0～(r－1)；

r——r 进制数的基数；

i 标明系数 a_i 所在位置。

例 1-1

$$(10110)_B = 1\times2^4+0\times2^3+1\times2^2+1\times2^1+0\times2^0 = (22)_D$$

例 1-2

$$\begin{aligned}(E5D7.A3)_H &= 14\times16^3+5\times16^2+13\times16^1+\\ &\quad 7\times16^0+10\times16^{-1}+3\times16^{-2}\\ &= (58839.63671875)_D\end{aligned}$$

2. 十进制数转换成其他进制数

十进制数转换成等值的其他进制数时，需要对十进制数的整数部分和小数部分分别进行转换。

(1) 整数部分的转换

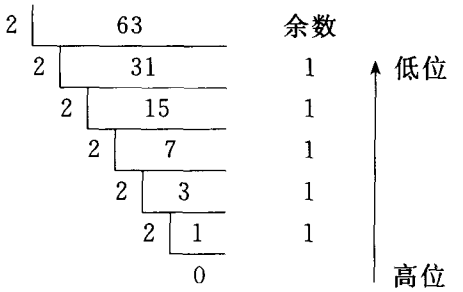
十进制数转换为 r 进制整数时，通常采用“除 r 取余”法，其步骤如下：

① 用给定的十进制数除以基数 r，所得的余数即为所求的等值 r 进制数的最低位。

② 将上一步所得的商再除以基数 r，所得余数为所求的 r 进制数的次低位。

③ 重复步骤②，一直进行到商为 0 为止，末次所得的余数为所求 r 进制数的最高位。

例 1-3　将十进制数 63 转换为等值的二进制数。



于是得　　$(63)_{10}=(11111)_2$

(2) 小数部分的转换

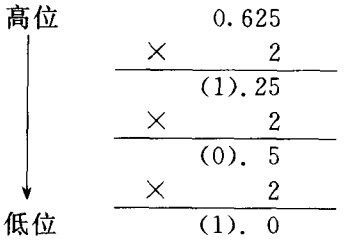
将十进制小数转换为 r 进制小数，通常采用“乘 r 取整”法，其步骤如下：

① 将给定的十进制数的小数部分乘以基数 r，所得乘积的整数部分便是等值 r 进制数小数的最高位。

② 用 r 乘步骤①所得乘积的小数部分，所得乘积的整数部分即为所求 r 进制数小数的次高位。

③ 重复步骤②，直至所得乘积的小数部分为 0，或已达到要求的精度为止，各次乘积的整数部分，便是 r 进制小数部分的各位，最后一次乘积的整数部分是最低位。

例 1-4　将十进制小数 0.625 转换成等值的二进制数。



于是得　$(0.625)_{10}=(0.101)_2$

应该指出的是，不是所有的十进制小数都能转换成有限位的 r 进制小数，因为用 r 去乘乘积的小数部分，若小数部分不为 0，则演算将一直进行下去。在用计算机进行转换的情况下，由于计算机的字长有限，当演算到最低位时，往往采用“四舍五入”的方法处理，这将出现误差，误差的大小(或称转换精度)与计算机的字长有关。

例 1-5　将$(0.513)_{10}$转换为八进制数，转换精度小于8^{-6}。

高位

$$
\begin{array}{rr}
 & 0.513 \\
\times & 8 \\
\hline
 & (4).104 \\
\times & 8 \\
\hline
 & (0).832 \\
\times & 8 \\
\hline
 & (6).656 \\
\times & 8 \\
\hline
 & (5).248 \\
\times & 8 \\
\hline
 & (1).984 \\
\times & 8 \\
\hline
 & (7).872
\end{array}
$$

低位

于是得　$(0.513)_{10}=(0.406520)_8$

3. 二进制数与八进制、十六进制数的相互转换

(1) 二进制数转换成八进制数

由于八进制数的基数$8=2^3$，所以，三位二进制数恰好等于一位八进制数。因此，二进制数转换成等值的八进制数的规则是，将二进制数由小数点开始分别向左和向右每三位划分为一组，不足三位的，整数部分可在最高位的左边添 0 补齐，小数部分可在最低位的右边添 0 补齐，缺几位添几位，每组用一位等值的八进制数来代替，即可得到相应的八进制数。

例 1-6　将二进制数 10011101.01 转换为八进制数。

$$(10011101.01)_2=0\ 1\ 0\ 0\ 1\ 1\ 1\ 0\ 1.\ 0\ 1\ 0=(235.2)_8$$

(2) 二进制数转换为十六进制数

同理，由于十六进制数的基数$16=2^4$，故可将二进制数由小数点开始分别向左和向右按四位为一组划分，不足四位时分别在最高位前和最低位后添 0 补齐。每组用一位等值的十六进制数来代替，即可得到相应的十六进制数。

例 1-7　将二进制数 10101100001.110101 转换为十六进制数。

$$(10101100001.110101)_2=0\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 1.\ 1\ 1\ 0\ 1\ 0\ 1\ 0\ 0=(561.D4)_{16}$$

(3) 八进制数转换为二进制数

把每位八进制数用三位二进制数表示即得对应的二进制数

例 1-8　将八进制数 257.3 转换为二进制数。

$$(257.3)_8=(010101111.011)_2=(1010111.011)_2$$

(4) 十六进制转换为二进制数

把每位十六进制数用四位二进制数表示即得对应的二进制数

例 1-9　将$(A3.B6)_{16}$转换为二进制数。

$$(A3.B6)_{16}=(10100011.10110110)_2=(10100011.1011011)_2$$

1.2 编码

在数字系统中，除了用数值表示信息外，还常常用字符表示信息，字符信息也采用由若干位的二进制数码来表示，这样就要为不同的字符信息分配不同的二进制代码，称其为对信息的编码。本节介绍一些常用的编码及它们的特性。

1.2.1 二—十进制编码

用四位二进制数表示一位十进制数所得的编码，称为二—十进制编码(Binary Coded Decimal Codes，简称为 BCD 码)。当用二进制代码表示 N 个不同信息时，所需的二进制代码的最少位数 n 应满足下式：

$$N \leqslant 2^n$$

一位十进制数，有 0～9 个不同的信息，$N=10$，可求得 $n=4$，也就是说，至少需要四位二进制数才能表示一位十进制数。

用四位二进制数可以组成 $2^4=16$ 个不同的编码组合，用其中的十个码组分别代表十进制中 0～9 个数，剩下六个多余的码组，称为冗余码组。

由于从 16 个二进制码组中任选十个码组的方案有很多种，因而产生了多种 BCD 码。下面着重讨论 8421 码、2421 码和余 3 码三种常用的 BCD 码。它们的编码如表 1-1 所示。

1. 8421BCD 码

从表 1-1 中可看出，8421BCD 码的编码特点是，十进制数 0～9 所对应的四位二进制代码，就是与该十进制数等值的二进制数，即四位二进制数 0000～1111 共 16 种码组中的前十种码组 0000～1001。剩余的六个码组 1010～1111 在 8421BCD 码中是不允许出现的，称为冗余码组。在这种二—十进制码中，每位的权值是固定的，故是一种有权码。从左向右每位的值分别是 8、4、2、1，所以，称这种二—十进制码为 8421BCD 码。

8421BCD 码具有奇偶特性，即凡是奇数(1、3、5、7、9)码组的最低位二进制数皆为 1，而偶数(0、2、4、6、8)码组的最低位二进制数皆为 0，因此，采用 8421 码容易判别它的奇偶性。

用 8421BCD 码对一个多位十进制数进行编码，只要把十进制数的各位数字编成对应的 8421 码。

$$(523)_{10}=(010100100011)_{8421BCD}$$

2. 2421BCD 码

2421 码也是一种有权码，它有四位，从左向右每位的权值分别为 2、4、2、1。表 1-1 中 2421BCD 码的特点是：它的十个编码组合中，0 与 9、1 与 8、2 与 7、3 与 6、4 与 5 之间的关系是自身按位取反，即互为反码。例如数码 4 的 2421BCD 的编码是 0100，而数码 5 的 2421BCD 编码是 1011，恰好是 0100 的反码。我们把具有这种特性的 BCD 码，称为自补码。自补码的优点是求补简便，所以 2421BCD 码在计算机系统中得到了广泛的应用。

应该指出，2421BCD 的编码方案不是惟一的。因为在 2421 码中，有部分数码存在两种加权方法。例如，十进制数 7，既可编成 0111，也可编成 1101。

3. 余 3BCD 码

余 3BCD 码也有四位，但每位的权值是不固定的，故是无权码。它是由每个 8421BCD 码加上“3”得到的，因此得名为余 3 码。

例如十进制数 6 的 8421BCD 码为 0110，则它的余 3BCD 码应为

$$0110+0011=1001$$

余 3BCD 码，也是一种自补码，它的 0～4 的代码分别与 9～5 的代码互为反码，所以余 3 码在计算机系统中也得到了广泛的应用。

1.2.2　循环码

循环码也叫余 3 格雷码。它的编码特点是，任何两个相邻码组（包括首尾两个码组）之间仅有一位不同。例如，十进制数 3 和 4 是相邻的两个码组，它们的循环码分别为 0101 和 0100，两者仅最右位不同，其他各码位均相同。循环码这一特性，能避免在码组转换的过渡过程中产生瞬时误码。因此，循环码在通信和测量技术中得到了广泛的应用，四位循环码的编码如表 1-2 所示。

表 1-2　四位循环码的编码

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

由表 1-2 还可看出，循环码具有反射特性，即它的最高位的 0 和 1 只改变一次，若在最高位 0 和 1 的交界处设一镜面（表中虚线所示），则其他各位的编码以镜面为界形成镜像反射。

用循环码表示十进制数时，为使 0 和 9 的码组也只能有一个码位不同，常采用如表 1-2 所示的余 3 循环码。它是从四位循环码的 16 个码组中去除首尾各三个码组而构成的。

分析余 3 循环码的编码特点，可知它也具有反射特性。

循环码的各位没有确定的权值，它属于无权码。

1.2.3 字符代码

在数字系统中，0 和 1 不仅可以代表数，还可用 0、1 组成表示字母和符号的代码。利用字符代码可以进行十进制数、字母和各种符号的信息交换。例如我国广泛使用的 ASCII（American Standard Code for Information Interchange 的缩写，称为美国标准信息交换码），就是用七位二进制数的编码组合来表示一个字符。

习题

1-1 把下面十进制数转换成二进制数。

(1) 2006 (2) 12.543 (3) 425.865 (4) 0.6875

1-2 把下列二进制转换为十进制数

(1) 1001101.001 (2) 11011101.111

(3) 10.10001 (4) 0.10110

1-3 把下面已知基数的数变换成指定基数的数

(1) 十进制数 225.25 变成二进制、八进制和十六进制数。

(2) 二进制数 11010111.11 变成二进制、八进制和十六进制数。

(3) 十六进制数 2AC5.D6 变成十进制、二进制和八进制数。

1-4 将下列十进制数分别转换成 8421BCD 码和余 3BCD 码。

(1) 384 (2) 7625 (3) 957 (4) 204

1-5 把下列 8421BCD 码译成十进制数

(1) 0110,0010,1000 (2) 0011,0100,0111,0101

(3) 1001,0001,0110

第2章 逻辑代数

逻辑代数是按一定逻辑规律进行运算的代数，它是分析和设计逻辑电路的一种数学工具，可用来描述数字电路和数字系统的结构和特性，逻辑代数又称为布尔代数，也叫开关代数。

2.1 逻辑代数的基本运算

2.1.1 基本逻辑运算

逻辑代数有“0”和“1”两种逻辑值，用来表示两种相互对立的逻辑状态，而没有赋予任何数量的概念。例如，它们可以表示电位的高低、信号的有无或者晶体管的导通、截止，也可以表示事件的真伪、对错、是否等，所以逻辑“1”和逻辑“0”与自然数的 1 和 0 有本质上的区别，决不能混淆。

生活中逻辑问题千变万化，但都可以用逻辑乘（与运算）、逻辑加（或运算）和逻辑非（非运算）三种基本逻辑运算来描述，它们可以由相应的逻辑电路实现。

1. 逻辑加（或运算）

逻辑加是描述或逻辑关系的，也称或运算。其运算符是“+”。实现或逻辑的电路通常称为或门。

或逻辑的定义是，当决定事件发生的各种条件（A、B、C、…）中，只要有一个或一个以上的条件具备，事件（Y）就发生。其运算关系式为：

$$Y=A+B+C+\cdots \tag{2-1}$$

逻辑变量 A 和 B 的或逻辑运算的规则如表 2-1 所示。

从表 2-1 可以看出，只要输入变量中有一个为 1，输出逻辑函数 Y 就为 1。

表中最后一行中 1+1=1，不同于普通的算术加法，它不表示数值的大小，仅表示一种逻辑判断。

实现逻辑加的电路称为或门（OR gate），图 2-1 显示了三种或门的逻辑符号，图中(a)为我国常用的符号；(b)为美日等国常用的符号；(c)为我国国家标准 GB 4728.12-85规定的符号。

表 2-1 或逻辑运算规则

A	B	$Y=A+B$
0	0	0
0	1	1
1	0	1
1	1	1

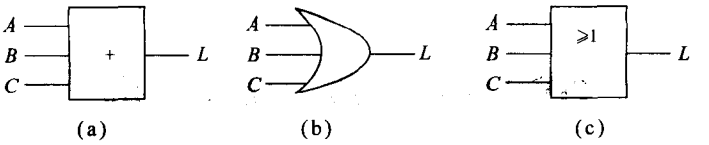


图 2-1 或门符号

2. 逻辑乘(与运算)

逻辑乘是描述与逻辑关系的,也称与运算,其运算符是"×"或"·"。实现与逻辑的电路通常称为与门。

与逻辑的定义是:只有决定事件的全部条件(A、B、C、…)都具备时,该事件(Y)才发生。其运算关系式为

$$Y=A\cdot B\cdot C\cdot\cdots \tag{2-2}$$

逻辑变量 A 和 B 的与逻辑运算的规则如表 2-2 所示。

表 2-2 与逻辑运算规则

A	B	$Y=A\times B$
0	0	0
0	1	0
1	0	0
1	1	1

从表 2-2 所示的运算结果可以看出,只要输入变量中有一个为 0,输出逻辑函数 Y 就为 0;只有当全部输入变量均为 1 时,输出函数 Y 才为 1。

实现逻辑乘的电路称为与门(AND gate),图 2-2 显示了三种与门逻辑符号。

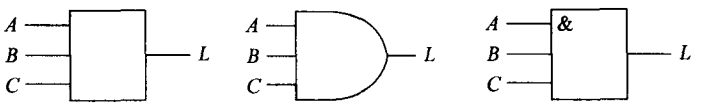


图 2-2 与门符号

有的书上用"∧"或"∩"表示与运算符号。

3. 逻辑非(非运算)

逻辑非是对一个变量的逻辑否定,当一条件不成立时,与其相关的事件才发生。逻辑非也是描述一个电路输出和输入之间的反相关系。其逻辑表达式为

$$Y=\overline{A} \tag{2-3}$$

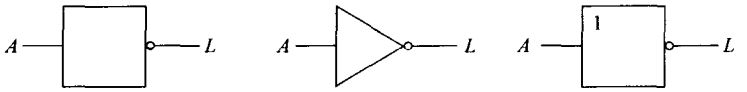
表 2-3 非逻辑运算规则

A	$Y=\overline{A}$
0	1
1	0

非逻辑的运算规则如表 2-3 所示。

通常,A 称为原变量,$\overline{A}$称为反变量。

实现逻辑非的电路称为非门(NOT gate)。图 2-3 显示了非门的三种逻辑符号。



(a) 常用符号　　(b) 美、日常用符号　　(c) 国标符号

图 2-3　非门符号

2.1.2　逻辑函数及其描述方法

1. 逻辑函数

逻辑函数用来描述一个逻辑系统中输入变量与输出变量之间的关系。例如，一个数字逻辑系统有三个输入变量 A、B、C 和一个输出变量 Y，当输入变量 A、B、C 的一组取值确定之后，输出变量 Y 的取值就被惟一地确定，它们之间就构成一种函数关系，这种函数关系被称之为逻辑函数，可表示为

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

在这里 A、B、C 和 Y 的取值都只有“0”和“1”两种逻辑值，不可能有其他取值。

2. 逻辑函数的描述方法

逻辑函数通常有三种描述方法，即逻辑表达式、真值表和逻辑图。

(1) 逻辑表达式

逻辑表达式是描述逻辑函数的最基本的方法，是用逻辑函数来描述一个数字逻辑系统输入输出变量之间的关系。

(2) 真值表

对于给定的逻辑问题，将所有输入变量的各种可能的取值和与之相对应的输出函数值排列成一个表格，这种表格称为真值表。

真值表的结构由两部分组成，左边一列为输入变量的全部可能的组合。例如，两个输入变量有 $2^2=4$ 种组合，n 个输入变量则有 2^n 种组合，右边一列为对应的输出逻辑函数的值。如逻辑表达式 $Y=A+B$，则其相应的真值表如表 2-4 所示。

表 2-4　真值表

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

此表描述了当输入变量 A 和 B 中有一个为 1 时，输出函数 Y 即为“1”这一逻辑加问题，由此可见，真值表同逻辑表达式一样都可用来描述具体的逻辑问题。

(3) 逻辑图

逻辑图是用逻辑符号来描述逻辑问题而形成的逻辑电路图。逻辑图能直观形象地描述逻辑系统输入和输出之间的逻辑关系，最基本的逻辑符号就是前面介绍的与门、或门和非