

The left side of the cover features a solid green background with several light green circles of varying sizes scattered across it.

Shuzi Dianzi Jishu

数字电子技术

路明礼/主编

武汉理工大学出版社

WUHAN UNIVERSITY OF TECHNOLOGY PRESS

高 职 高 专 教 材

数字电子技术

主 编 路明礼

副主编 卢秉娟

武汉理工大学出版社

内容简介

本书是依据教育部“电子技术基础课程教学基本要求(1995年修订版)”编写的。主要内容包括数字电路基础、逻辑门电路、组合逻辑电路、触发器、时序逻辑电路、存储器和可编程逻辑器件、ADC和DAC、数字集成电路的典型应用、数字电子电路读图及课程设计等,共分9章。本书“以能力为本,理论够用为度”的原则,压缩了集成电路电气特性和内部电路工作原理分析等,从应用的角度,更注重集成电路的外部特性。

本书简明扼要,深入浅出,便于自学,可作为高职高专电类及相关专业教材,也可供从事电子技术工作的工程技术人员学习参考。

图书在版编目(CIP)数据

数字电子技术/路明礼主编. —武汉:武汉理工大学出版社,2005

ISBN 7-5629-2317-5

I. 数…

II. 路…

III. 数字电路-电子技术

IV. TN79

中国版本图书馆 CIP 数据核字(2005)第 086666 号

出版者:武汉理工大学出版社(武汉市:武昌珞狮路 122 号 邮编:430070)

印刷者:武汉理工大印刷厂印刷

发行者:各地新华书店

开 本:787×1092 1/16

印 张:12.75

字 数:320 千字

版 次:2005 年 8 月第 1 版 2005 年 8 月第 1 次印刷

书 号:ISBN 7-5629-2317-5/TP·118

印 数:1~1500 册

定 价:22.00 元

(本书如有印装质量问题,请向承印厂调换)

前 言

结合高职高专教育专业培养目标,以职业为基础,以能力为根本,理论够用为度的原则,是本书编写的出发点。

本书的特点是考虑到高职高专教育的特点,降低理论深度,力求做到基本概念清楚,理论联系实际,语言简练通畅。始终贯彻以培养能力为主、以应用为目的的原则。为了适应数字电子技术特别是数字集成器件的飞速发展,在编写上注意知识的更新,淡化集成电路内部结构和内部工作原理的讲述,注重器件的外部功能特性和应用,尽可能多的介绍常用器件和新器件的应用,还注意了与微型计算机的密切联系。书中每章后有一定数量的习题,书中打*号的章节为选学内容,本教材的参考学时为50~60学时。

全书共分9章,内容为:数字电路基础、逻辑门电路、组合逻辑电路、触发器、时序逻辑电路、存储器和可编程逻辑器件、A/D、D/A转换器、数字集成电路的典型应用电路、数字电子电路读图及课程设计。第1、2章由卢秉娟编写,第3章由赵顺东编写,第4章、附录由黄慧敏编写,第5、6章由路明礼编写,第7、8、9章由江姝妍编写。姚正祥主任审阅并提出了许多宝贵意见,卢秉娟对全书进行了统稿,在此表示感谢。

由于编者经验不足,水平有限,书中难免会有许多缺点、错误和不妥之处,诚恳希望使用本教材的教师和读者提出宝贵意见。

编者

2005,5

目 录

1 数字电路基础	1
1.1 概述	1
1.2 计数进制及代码	2
1.2.1 各种进制及转换	2
1.2.2 二进制代码	3
1.3 逻辑关系和逻辑运算	5
1.4 逻辑代数的基本运算和逻辑函数	9
1.4.1 逻辑代数的基本运算	9
1.4.2 逻辑函数的表示方法	10
1.5 逻辑函数的化简	11
1.5.1 逻辑函数的公式化简法	11
1.5.2 逻辑函数的图形化简法	12
1.5.3 具有约束的逻辑函数及其卡诺图化简法	17
本章小结	19
习题	20
2 逻辑门电路	22
2.1 概述	22
2.2 逻辑门电路	22
2.2.1 二极管门电路	22
2.2.2 非门(反相器)	24
2.2.3 其它常用逻辑门电路	25
2.3 两大系列逻辑门电路及逻辑门 IC 简介	26
2.3.1 TTL 系列门电路	26
2.3.2 CMOS 系列门电路	35
2.3.3 门电路在使用中应注意的问题	37
本章小结	39
习题	39
3 组合逻辑电路	42
3.1 概述	42
3.2 组合逻辑电路基本分析方法和设计方法	42
3.2.1 组合电路的基本分析方法	42
3.2.2 组合电路的基本设计方法	44
3.3 常用的组合逻辑电路及 IC 简介	47
3.3.1 编码器和译码器	47

3.3.2	数据选择器和数据分配器	58
3.3.3	加法器和数据比较器	59
3.4	组合逻辑电路中的竞争冒险现象	63
	本章小结	65
	习题	66
4	触发器	68
4.1	触发器概述	68
4.2	基本 RS 触发器	68
4.2.1	电路结构及工作原理	68
4.2.2	逻辑功能及触发方式	69
4.3	时钟触发器	71
4.3.1	触发器的触发方式及几种触发器介绍	71
4.3.2	时钟触发器逻辑功能分类	77
4.4	触发器的逻辑功能转换	80
4.5	常用触发器 IC 简介	81
	本章小结	84
	习题	84
5	时序逻辑电路	87
5.1	概述	87
5.2	时序逻辑电路基本分析方法和设计方法	88
5.2.1	时序电路的基本分析方法	88
5.2.2	时序电路的基本设计方法 *	92
5.3	计数器	99
5.3.1	计数器的分类和基本原理	99
5.3.2	常用的计数器 IC 简介	101
5.3.3	用集成计数器实现任意进制计数器	110
5.4	寄存器	114
5.4.1	寄存器的功能	114
5.4.2	移位寄存器型计数器	117
	本章小结	118
	习题	118
6	存储器和可编程逻辑器件	121
6.1	概述	121
6.2	只读存储器(ROM)	121
6.2.1	ROM 的结构及工作原理	122
6.2.2	ROM 的应用和容量扩展	124
6.3	读写存储器(RAM)	128
6.3.1	RAM 的结构	128
6.3.2	RAM 的存储单元简介	129

6.3.3	RAM 的容量扩展	131
6.4 *	可编程逻辑器件(PLD)	132
6.4.1	PLD 的结构	133
6.4.2	PLD 的基本原理	135
	本章小结	141
	习题	142
7	A/D、D/A 转换器	143
7.1	概述	143
7.2	D/A 转换器基本原理	143
7.3	A/D 转换器基本原理	147
	本章小结	152
	习题	153
8	数字集成电路的典型应用电路	154
8.1	555 定时器及脉冲波形产生和整形电路	154
8.1.1	555 定时器	154
8.1.2	多谐振荡器	155
8.1.3	单稳态电路	158
8.1.4	施密特触发器电路	160
8.2	其它若干典型应用电路	163
	本章小结	165
	习题	166
9	数字电子电路读图及课程设计	167
9.1	数字电子电路读图方法	167
9.1.1	电子电路图种类及读图方法	167
9.1.2	读图举例	168
9.2	数字电子电路课程设计方法	171
9.2.1	数字系统设计介绍	171
9.2.2	数字电子电路课程设计方法举例	171
附录 1	TTL 集成电路系列介绍	176
附录 2	国产半导体集成电路型号命名方法(GB3430—89)	177
附录 3	部分集成逻辑门电路新、旧图形符号对照	179
附录 4	部分常用的集成电路芯片功能引出端排列图	180
	部分习题解答	186
	参考文献	194

1 数字电路基础

1.1 概 述

1. 数字电路及其特点

电子电路中的信号可分为两类,一类是在时间和大小上都是连续的模拟信号,由模拟电路来处理;另一类是在时间和大小上都是离散的数字信号,其大小变化是某个最小量的整数倍,由数字电路来处理。

数字电路又称为逻辑电路,是研究信号输入、输出之间逻辑关系的学科。与模拟电路相比较,它具有以下特点:

①模拟电路注重研究信号的放大、相位关系等,采用的晶体管一般工作在放大状态;数字电路注重研究信号输入、输出之间的逻辑关系,采用的晶体管一般工作在截止或饱和状态。

②数字电路分析和设计的数学工具是逻辑代数。逻辑代数以二进制为基础,逻辑代数中的变量称为逻辑变量,也是用英文字母 A 、 B 、 C 、……等表示的,其取值(逻辑值)只有 1 和 0 两种,这里的 1 和 0 并不表示具体数值大小,而是表示逻辑变量的两种相反的状态,如是和否、真和假、电平的高和低、三极管的(饱和)导通和截止、开关的接通和关断等。数字电路中的数字量或逻辑量均采用二进制来表示。

③数字电路在对信号处理时,是按照人们事先设计好的逻辑关系进行逻辑运算和逻辑判断的,也就是说,数字电路具有一定的运算和思维能力。因此,人们可以利用数字电路制造出具有一定智能的装置,如机器人和电子计算机等。

另外,数字电路具有集成度高,抗干扰能力强,工作可靠性高,芯片通用性强,信号便于存储等特点。

2. 数字电路举例

数字电路大致包括信号的产生、放大、整形、传送、控制、状态记忆、计数、数字显示等内容。以数字频率计为例可以说明其基本结构和内容。

如图 1.1 所示是一个数字频率计的简单框图,它可以用来测量周期信号的频率 f_x 。被测信号频率为 f_x ,通过放大整形后变换成频率仍为 f_x 的矩形脉冲信号 A ,秒脉冲电路产生宽度为 1s 的控制信号 B ,二者进入一个控制门(闸门),控制门输出的就是 1s 内的被测信号 Y 。计数器对 Y 的变化次数进行累计,并通过译码显示电路直接显示出 f_x 的值。

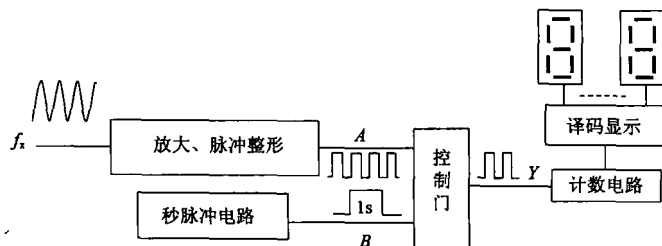


图 1.1 数字频率计的简单框图

1.2 计数进制及代码

1.2.1 各种进制及转换

1. 各种进制

(1) 十进制

十进制是我们日常生活和工作中最常使用的计数进制之一。在十进制中,有 0、1、2、3、4、5、6、7、8、9 共十个数码,计数基数为 10。超过 9 时要用多位数表示,其中低位数和相邻高位数之间的进位关系是“逢十进一,借一做十”。

例如: $648.72 = 6 \times 10^2 + 4 \times 10^1 + 8 \times 10^0 + 7 \times 10^{-1} + 2 \times 10^{-2}$

所以任何一个正的十进制数 D 均可按下式展开:

$$(D)_{10} = \sum k_i \times 10^i \quad (1.1)$$

其中 k_i 是第 i 位的系数,它可能是 0~9 十个数码中的任何一个。若整数部分的位数是 n ,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。

若以 N 取代式(1.1)中的 10,即可得到任意进制(N 进制)数的一般展开表达式

$$(D)_N = \sum k_i \times N^i \quad (1.2)$$

式中 N 称为计数制的基数; k_i 为第 i 位的系数; N^i 称为第 i 位的权。

(2) 二进制

在数字电路中应用最广的是二进制。在二进制数中,每位仅有 0 和 1 两种可能的数码,计数基数为 2。低位和相邻高位之间的进位关系是“逢二进一,借一做二”。

由式(1.2),任何一个二进制数 D 均可展开为

$$(D)_2 = \sum k_i \times 2^i \quad (1.3)$$

例如: $101.11 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$

(3) 八进制

在八进制数中,有 0、1、2、3、4、5、6、7 共八个数码,计数基数为 8。低位与相邻高位之间的进位关系是“逢八进一,借一做八”。

任何一个八进制数都可以按(1.2)展开为

$$(D)_8 = \sum k_i \times 8^i \quad (1.4)$$

例如: $(37.25)_8 = 3 \times 8^1 + 7 \times 8^0 + 2 \times 8^{-1} + 5 \times 8^{-2}$

(4) 十六进制

在十六进制数中,有 0~9、A(10)、B(11)、C(12)、D(13)、E(14)、F(15)共十六个数码,计数基数为 16。低位与相邻高位之间进位关系是“逢十六进一,借一做十六”。

任意一个十六进制数均可展开为

$$(D)_{16} = \sum k_i \times 16^i \quad (1.5)$$

例如: $2A.7F = 2 \times 16^1 + A \times 16^0 + 7 \times 16^{-1} + F \times 16^{-2} = 2 \times 16^1 + 10 \times 16^0 + 7 \times 16^{-1} + 15 \times 16^{-2}$

十六进制与二进制数之间的转换简单,在微机系统中常用十六进制符号书写程序。

2. 进制之间的转换

(1) 十进制数转换成二进制数

对于十进制数的整数部分,转换成为二进制数的口诀为“除 2 取余,逆序排列”;

例如:将 $(25)_{10}$ 转换为二进制数。转换结果为 $(25)_{10} = (11001)_2$ 运算过程如下。

$$\begin{array}{r|l}
 2 & 25 \\
 \hline
 2 & 12 \cdots 1 \\
 2 & 6 \cdots 0 \\
 2 & 3 \cdots 0 \\
 2 & 1 \cdots 1 \\
 & 0 \cdots 1
 \end{array}$$

对于十进制数的小数部分,转换成为二进制数的口诀为“乘 2 取整,正序排列”。即将十进制数的小数部分乘以 2 后,将整数部分取出后,再将剩下的小数部分乘以 2,再次取整,以此类推。

例如 $(0.625)_{10} = (0.101)_2$ 运算过程如下。

$$\begin{array}{c}
 0.625 \xrightarrow[\text{取 1}]{\times 2} 0.25 \xrightarrow[\text{取 0}]{\times 2} 0.50 \xrightarrow[\text{取 1}]{\times 2} 0.00 \\
 \xrightarrow{\hspace{10em}} \text{取 整}
 \end{array}$$

有些小数可能乘 2 乘不尽,转换时会有误差,可以按要求转换到保留小数点之后若干位。另外若整数非常大,除 2 操作不方便,可以采用相同的方法除以 8 或者除以 16 得到八进制或者十六进制数,然后再转换成二进制数。

(2) 二进制数转换成十进制数

转换很简单,任意进制数,只需按照式(1.3)展开求值即可得到十进制数。

例如将 $(1001.11)_2$ 转换为十进制数。

$$\begin{aligned}
 (1001.11)_2 &= 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} \\
 &= 8 + 0 + 0 + 1 + 0.5 + 0.25 = (9.75)_{10}
 \end{aligned}$$

(3) 二进制数与八进制、十六进制数的转换

二进制数与八进制、十六进制数的关系为:

三位二进制数可以对应写为一位八进制数,即 $(000)_2 \sim (111)_2 = (0)_8 \sim (7)_8$;

四位二进制数可以对应写为一位十六进制数,即 $(0000)_2 \sim (1111)_2 = (0)_{16} \sim (F)_{16}$ 。

在转换时,二进制的整数部分从小数点向左每三位或者每四位地分组,小数部分从小数点向右每三位或者每四位地分组,即可实现二进制数转换为八进制或者十六进制数。掌握这一关系就能很容易地实现它们之间的转换。

例如 $(10011010111)_2 = (10'011'010'111)_2 = (010'011'010'111)_2 = (2327)_8$

$(10011010111)_2 = (100'1101'0111)_2 = (0100'1101'0111)_2 = (4D7)_{16}$

$(2AF)_{16} = (0010'1010'1111)_2 = (1010101111)_2$

$(100110111.101011)_2 = (467.53)_8 = (137.AC)_{16}$

1.2.2 二进制代码

在数字系统中,为了表示文字、符号等信息,往往采用一定位数的二进制数码,这种具有特

定含义的二进制数码称为二进制代码。建立这种代码与十进制数码、字母、符号的一一对应关系的过程叫做编码。若所需编码的信号对象有 N 个,则需用二进制数代码的位数 n 应满足下面关系

$$N \leq 2^n$$

例如 $N=10$,则由 $N \leq 2^n$ 取 $n=4$ 。

下面介绍几种常见的编码,见表 1.1。

表 1.1 常用各种代码的编码表

十进制数	自然码	循环码	BCD 码				
			8421BCD 码	十进制数	余 3 码	余 3 循环码	十进制数
0	0000	0000	0000	0			
1	0001	0001	0001	1			
2	0010	0011	0010	2			
3	0011	0010	0011	3	0011	0010	0
4	0100	0110	0100	4	0100	0110	1
5	0101	0111	0101	5	0101	0111	2
6	0110	0101	0110	6	0110	0101	3
7	0111	0100	0111	7	0111	0100	4
8	1000	1100	1000	8	1000	1100	5
9	1001	1101	1001	9	1001	1101	6
10	1010	1111			1010	1111	7
11	1011	1110			1011	1110	8
12	1100	1010			1100	1010	9
13	1101	1011					
14	1110	1001					
15	1111	1000					

1. 自然码

自然码是一种四位二进制代码,可以看作一个四位二进制数,各位的权依次是 8、4、2 和 1,每个代码的数值恰好等于它所表示的十进制数的大小。因此,将这种代码又叫做 8421 编码。

2. 循环码

循环码,又叫格雷码(Gray Code),也是一种四位二进制代码。它的每 1 位代码都是按一定的周期规律进行循环。应用较多的是三位和四位的循环码,对于四位循环码,循环规律为:

右起第 1 位的循环周期是 0110;

第 2 位的循环周期是 00111100;

第 3 位的循环周期为 0000111111110000;

第 4 位的循环周期为 000000001111111100000000。

若每相邻的两个代码组合相比较,只有一位不同,其它位都相同,则称为这种代码具有逻辑相邻性。循环码就具有逻辑相邻性的特点。

3. 二 - 十进制编码(BCD 码; Binary Coded Decimal code)

用 4 位二进制代码表示一位十进制数 0~9 的代码叫做 BCD 码。它包括 8421BCD 码、余 3 码和余 3 循环码。

(1)8421BCD 码

8421BCD 码是最常用的一种编码。它是取 8421 码(共 16 种)的前十种组合 0000~1001 来对应表示一位十进制数 0~9 的。后六种组合 1010~1111 不用,作为无效代码或伪码。它属于一种有权码,各位代码的权依次为 8、4、2 和 1。

(2)余 3 码

余 3 码是用 8421 码的中间十种 0011~1100 来对应表示一位十进制数的一种编码。它将 8421 码的前 3 种、后 3 种代码组合作为其无效代码。它属于一种无权码。

(3)余 3 循环码

余 3 循环码是用 4 位从 0000 开始的循环码的中间十种组合(0010~1010)对应表示一位十进制数 0~9 的一种编码。

另外,还有恒权码 2421 码、5211 码,这种代码各位的权是恒定的;还有字符、字母编码,主要有 ASCII 和 ISO 码,这里就不再介绍了。

1.3 逻辑关系和逻辑运算

在数字电路中,有算术运算和逻辑运算。算术运算也有加、减、乘、除运算,与普通代数中的运算方法相同。这里主要讲逻辑运算。

在逻辑代数中,有三种最基本的逻辑关系:与、或、非。在这三种最基本逻辑关系的基础上,又有复合逻辑关系:与非、或非、与或非、异或、同或等。对于这些逻辑关系都对应有各自的逻辑运算。

1. 与逻辑关系和与运算

当决定一件事情的所有条件都具备时,这件事情才会发生。这种关系称为与逻辑关系。

如图 1.2(a)所示是一个简单的与逻辑关系电路。两个开关 A、B 的状态决定灯泡 Y 的状态,只有开关 A、B 全部闭合,灯才会亮;否则灯泡灭。

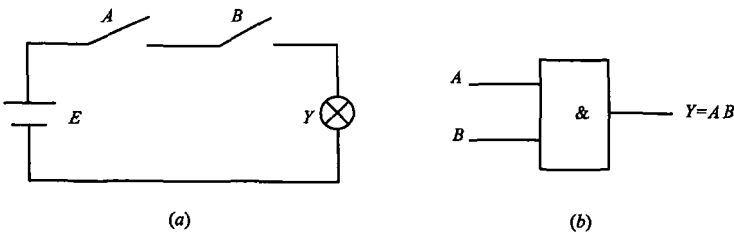


图 1.2 与逻辑关系及与运算逻辑符号

(a)电路;(b)与运算逻辑符号

表 1.2 是这个电路的功能表。

表 1.2 图 1.2 电路的功能表

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

现把开关 A、开关 B、灯泡 Y 抽象出来作为逻辑变量 A、B、Y，其中 A、B 属于输入变量，Y 属于输出变量。并且这里我们规定用二进制数码 0 表示开关断开、灯泡灭，用 1 表示开关闭合、灯泡亮，这一替换过程叫做逻辑状态赋值，由此将得到另一个数学表格如表 1.3 所示，这个表格叫做逻辑真值表，简称真值表。

真值表是反映输入变量的所有可能的取值组合与输出变量的函数值一一对应关系的表格。它是数字电路中非常重要的概念。

表 1.3 与逻辑关系真值表

输 入		输 出
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

逻辑与也叫逻辑乘，在数字电路中，用与运算逻辑符号来表示，如图 1.2(b)。这个符号也叫与门。

与逻辑关系除用真值表来描述外，还可以用与运算逻辑函数表达式来描述：

$$Y = A \cdot B = AB$$

式中的“ $\cdot$ ”为逻辑与运算符，可以省略不写。读做 Y 等于 A 与 B，或者 A 乘 B。

2. 或逻辑关系和或运算

当决定一件事情的所有条件中，只要有一个具备时，这件事情就会发生。这种关系称为或逻辑关系。

如图 1.3(a)所示是一个简单的或逻辑关系电路。两个开关 A、B 的状态决定灯泡 Y 的状态，只要开关 A、B 有一个闭合，灯就会亮；只有全断开灯泡才会灭。

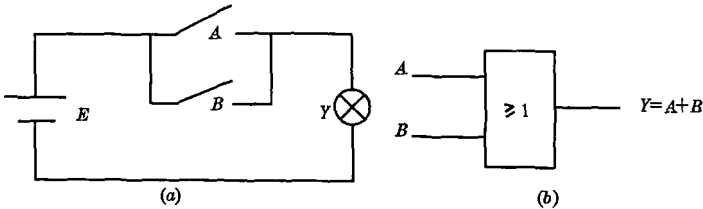


图 1.3 或逻辑关系及或运算逻辑符号

(a) 电路；(b) 或运算逻辑符号

表 1.4 是这个电路的功能表。

表 1.4 图 1.3 电路的功能表

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

按前面的同样规定进行状态赋值,将得到其真值表如表 1.5 所示。

逻辑或也叫逻辑加,在数字电路中,用或运算逻辑符号来表示,如图 1.3(b)。这个符号叫或门。

或逻辑关系除用真值表来描述外,同样可以用或运算逻辑函数表达式来描述:

$$Y = A + B$$

式中的“+”为逻辑或运算符,读做 Y 等于 A 或 B,或者 A 加 B。

表 1.5 或逻辑关系真值表

输 入		输 出
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

真值表中,出现“1+1=1”,要说明的是,这是逻辑运算,这里的“1”并不代表具体的数,没有数的概念,而是代表一种逻辑状态,这在前面已经规定了。

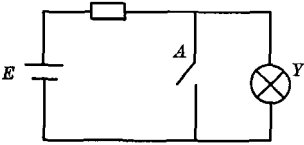
3. 非逻辑关系和非运算

如图 1.4(a)所示是一个简单的非逻辑关系电路。开关 A 的状态决定灯泡 Y 的状态,开关 A 闭合,灯就灭;A 断开灯泡就会亮。

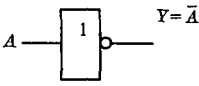
真值表如表 1.6 所示,其逻辑符号如图 1.4(b)所示,叫做非门。非运算逻辑函数表达式为:

$$Y = \bar{A}$$

式中变量字母上的“—”为逻辑非运算符,读做 Y 等于 A 非,或者 A 反。非就是否定。非门也叫反相器。



(a)



(b)

表 1.6 非逻辑关系真值表

A	Y
0	1
1	0

图 1.4 非逻辑关系及逻辑符号
(a)电路;(b)非运算逻辑符号

4. 逻辑变量、逻辑函数及复合逻辑关系

(1) 逻辑变量、逻辑函数

以上已经碰到逻辑变量和逻辑函数的概念,这里重申一下。

逻辑变量也是用英文字母表示的变量,其取值只有两种,或者是 1 或者是 0,没有第三种可能。而且,这里的 1 和 0 没有数值大小的含义,仅用来表示事物相互对立而又联系的两个方面,即两种对立的状态。

前面介绍的几种逻辑关系表达式,均抽象地表达了输入变量和输出变量的函数关系。一般地讲,如果输入逻辑变量 A、B、C、…的取值确定之后,输出逻辑变量 Y 的值也被唯一的确定

了,那么就称 Y 是 $A、B、\cdots$ 的逻辑函数,并写成为

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

一般,任一逻辑函数均可以用对应的真值表直观表达出来。

(2) 复合逻辑关系

除了以上三种最基本的逻辑运算外,还有一些其他常用的复合逻辑运算,如图 1.5 所示是几种常用复合逻辑关系的逻辑符号。

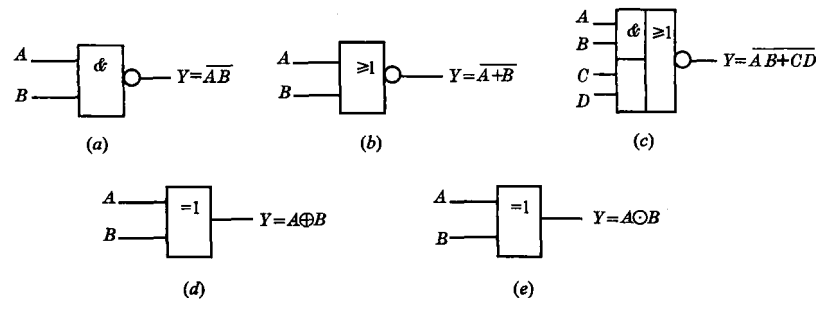


图 1.5 几种常用的逻辑符号

(a)与非门;(b)或非门;(c)与或非门;(d)异或门;(e)同或门

① 与非逻辑关系

与非逻辑关系是对与逻辑关系的否定,逻辑表达式为: $Y = \overline{AB}$,读做 A 与 B 的非。逻辑符号如图 1.5(a)所示,叫做与非门,与非逻辑关系真值表如表 1.7 所示。从真值表可以总结出与非逻辑关系为:有 0 出 1,全 1 出 0。

② 或非逻辑关系

或非逻辑关系是对或逻辑关系的否定,逻辑表达式为: $Y = \overline{A+B}$,读做 A 加 B 的非。或非逻辑关系真值表如表 1.7 所示,逻辑符号如图 1.5(b)所示,叫做或非门。从真值表可以总结出或非逻辑关系为:有 1 出 0,全 0 出 1。

③ 与或非逻辑关系

与或非逻辑关系逻辑表达式为:

$$Y = \overline{AB + CD}$$

其逻辑符号如图 1.5(c)所示,叫做与或非门。

表 1.7 常用复合逻辑关系的真值表

输入		几种逻辑输出			
A	B	$\overline{AB}$	$\overline{A+B}$	$\overline{AB+AB}$	$\overline{AB+AB}$
0	0	1	1	0	1
0	1	1	0	1	0
1	0	1	0	1	0
1	1	0	0	0	1

④ 异或逻辑关系、同或逻辑关系

所谓异或,是指两个输入逻辑变量相比较,相同则输出 0;相异则输出 1。

异或逻辑关系逻辑表达式为:

$$Y = \overline{AB} + A\overline{B} = A \oplus B$$

其逻辑符号如图 1.5(d)所示,叫做异或门。

所谓同或,是指两个输入逻辑变量相比较,相同则输出 1;相异则输出 0。所以它是异或逻辑关系的非。

其逻辑符号如图 1.5(e)所示,叫做同或门。

其逻辑关系逻辑表达式为:

$$Y = \overline{AB} + \overline{AB} = \overline{AB} + AB = \overline{A \oplus B} = A \odot B$$

实际的逻辑问题,往往变量多,非常复杂,但是它们都是通过基本逻辑关系的组合或复合来实现的。在复杂的逻辑运算中要特别注意运算的优先顺序。按优先顺序为:①括号;②非运算;③与运算;④异或运算;⑤或运算。

1.4 逻辑代数的基本运算和逻辑函数

1.4.1 逻辑代数的基本运算

数字电路的基本数学工具就是逻辑代数。逻辑代数中的变量称为逻辑变量,其取值只有 0 和 1。逻辑运算法则中有公理、定理、规则和公式。

1. 公理

① $0 \cdot 0 = 0; 1 + 1 = 1$

② $0 + 0 = 0; 1 \cdot 1 = 1$

③ $0 \cdot 1 = 0; 1 + 0 = 1$

④ $\overline{0} = 1; \overline{1} = 0$

2. 定理

① $A \cdot 1 = A; A + 0 = A$

② $A + 1 = 1; A \cdot 0 = 0$

③ $A \cdot \overline{A} = 0; A + \overline{A} = 1$

④ $A \cdot A = A; A + A = A$

⑤ 交换律: $AB = BA; A + B = B + A$

⑥ 结合律: $A(BC) = (AB)C; A + (B + C) = (A + B) + C$

⑦ 分配律: $A(B + C) = AB + AC; A + (BC) = (A + B)(A + C)$

⑧ 德·摩根定理: $\overline{AB} = \overline{A} + \overline{B}; \overline{A + B} = \overline{A} \cdot \overline{B}$

⑨ 还原律: $\overline{\overline{A}} = A$

证明以上定理的基本方法就是看等号两边的逻辑表达式对应的真值表是否完全相同,若相同,则等式成立。证明(略)。

3. 规则

①代入规则

在任何逻辑等式的两边出现同一逻辑变量的地方,都代之以一个逻辑函数,等式依然成立。这个规则称为代入规则。

例如,在等式 $A(B + C) = AB + AC$ 中,将所有的 C 都用 $C + D$ 替代,则等式

$$A(B + C + D) = AB + AC + AD \quad \text{依然成立。}$$

②反演规则

对于任意一个逻辑函数 Y , 将其中的逻辑运算作变换: “ $\cdot$ ” $\rightarrow$ “ $+$ ”, “ $+$ ” $\rightarrow$ “ $\cdot$ ”; “ 0 ” $\rightarrow$ “ 1 ”, “ 1 ” $\rightarrow$ “ 0 ”; 原变量 $\rightarrow$ 反变量, 反变量 $\rightarrow$ 原变量, 则得到新的表达式是函数 Y 的反函数 $\bar{Y}$, 这个规则称为反演规则。利用这一规则可以很容易直接求出一个函数的反函数。在变换时应注意两点: 一是注意保持原逻辑函数的运算优先次序不变; 二是单变量上的非号应保持不变。

例如, 若函数 $Y = \bar{A}B + \bar{C}\bar{D} + E + 0$, 则反函数 $\bar{Y} = (A + \bar{B}) \cdot \overline{C + \bar{D}} \cdot \bar{E} \cdot 1$

③对偶规则

对于任一个逻辑函数 Y , 若将其中的逻辑运算作变换: “ $\cdot$ ” $\rightarrow$ “ $+$ ”, “ $+$ ” $\rightarrow$ “ $\cdot$ ”; “ 0 ” $\rightarrow$ “ 1 ”, “ 1 ” $\rightarrow$ “ 0 ”, 则得到新的表达式是 Y 的对偶式 Y' 。

前面给出的公理、定理, 全是按照对偶规则列出的。如, $A \cdot \bar{A} = 0$ 的对偶式是 $A + \bar{A} = 1$; $A(B + C) = AB + AC$ 的对偶式是 $A + (BC) = (A + B)(A + C)$ 等。

如果两个逻辑表达式相等, 则其对偶式也相等。即若 $Y = X$, 则 $Y' = X'$ 。

4. 常用公式

利用上面的公理、定理、规则可以得到一些常用公式。这些公式将用于逻辑函数化简。

① 吸收公式 $A + A \cdot B = A$

② 消因子公式 $A + \bar{A} \cdot B = A + B$

③ 并项公式 $AB + A\bar{B} = A$

④ 多余项公式 $AB + \bar{A}C + BC = AB + \bar{A}C$
 $AB + \bar{A}C + BCD = AB + \bar{A}C$

证明: $AB + \bar{A}C + BC = AB + \bar{A}C + BC(A + \bar{A})$
 $= AB + ABC + \bar{A}C + \bar{A}BC$
 $= AB(1 + C) + \bar{A}C(1 + B)$
 $= AB + \bar{A}C$

⑤ 异或运算公式 $A \oplus B = B \oplus A = A\bar{B} + \bar{A}B$

$$A \oplus B \oplus C = (A \oplus B) \oplus C = A \oplus (B \oplus C) = (A \oplus C) \oplus B$$

$$A \oplus 1 = \bar{A}$$

$$A \oplus 0 = A$$

$$A \oplus A = 0$$

$$A \oplus \bar{A} = 1$$

1.4.2 逻辑函数的表示方法

逻辑函数的表示方法有逻辑表达式、真值表、卡诺图、逻辑图等方法。

1. 逻辑表达式

用与、或、非等逻辑运算表示逻辑函数的各变量之间关系的代数式, 称为逻辑表达式。例如, $Y = A + B$, $L = ab + \bar{a}c$, $C = Q_3 Q_0$ 等。

2. 真值表

前述中已经用到真值表, 并给出了真值表的定义。真值表中, 每个输入变量只有 0、1 两种取值, n 个变量就有 2^n 个不同的取值组合, 而每种组合都有对应的输出逻辑值。一个确定的