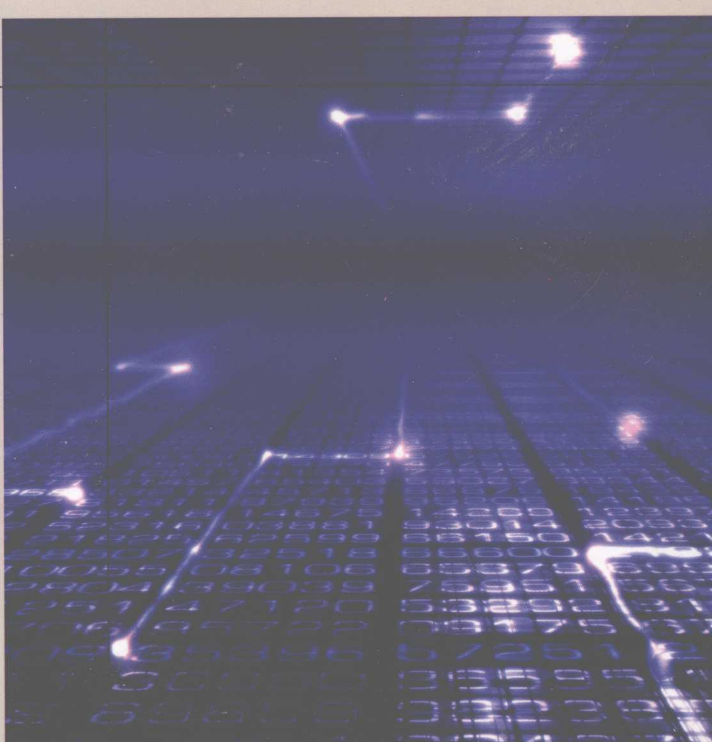
中等职业技术教育系列教材

数字电子技术基础

SHUZI DIANZI JISHU JICHU

主编　乐小勇　胡勋阳



華中师範大学出版社

中等职业技术教育系列教材

数字电子技术基础

主　编：乐小勇　胡勋阳
副主编：王仲良　江　炜　方向阳　孙宜潮
编　者：（排名不分先后）
程　琴　江　炜　胡勋阳
王仲良　汪　丹　晏玉华
刘金胜　胡雪梅　丁梅华
方向阳　乐小勇　孙宜潮

华中师范大学出版社

内容简介

本书根据教育部最新制定的《中等职业学校电工与电子技术教学大纲(试行)》并结合中职教育的现状,按照“必须、实用、够用”的原则编写。主要内容涉及数字电路基础、逻辑门电路、常用组合逻辑电路、触发器与时序逻辑电路、脉冲产生和整形电路、半导体存储器与可编程器件、数模与模数转换等。其中部分内容可根据各自情况作为选修内容。

本书主要供中等职业技术学校有关专业使用,也可供工程技术人员参考。

新出图证(鄂)

图书在版编目(CIP)数据

数字电子技术基础/乐小勇　胡勋阳 主编. —武汉:华中师范大学出版社,2008.9

(中等职业技术教育系列教材)

ISBN 978-7-5622-3476-0

Ⅰ.数…　Ⅱ.①乐…　②胡…　Ⅲ.数字电路—电子技术—专业学校—教材　Ⅳ.TN79

中国版本图书馆 CIP 数据核字(2008)第 134169 号

书　　名:数字电子技术基础
主　　编:乐小勇　胡勋阳
责任编辑:马知远　　责任校对:张　忠　　封面设计:罗明波
编 辑 室:第二编辑室　　电　　话:027—67867362
出版发行:华中师范大学出版社
社　　址:湖北省武汉市珞喻路 152 号
销售电话:027—67863040　67863426　67867076　67861549
邮购电话:027—67861321
传　　真:027—67863291
网　　址:http://www.ccnupress.com　　电子信箱:hscbs@public.wh.hb.cn
印　　刷:武汉市福成启铭彩色印刷包装有限公司　　督　　印:章光琼
字　　数:206 千字
开　　本:787mm×1092mm　　印　　张:12
版　　次:2008 年 9 月第 1 版　　印　　次:2008 年 9 月第 1 次印刷
印　　数:1—5000　　定　　价:19.50 元

中等职业技术教育系列教材

前　言

为了响应国家大力发展职业教育的号召，为社会培养高素质、高技能的复合型人才，积极推进素质教育，适应构建和谐社会的需要，按照“以服务为宗旨，以就业为导向”的职业教育办学方针，根据职业教育的规律和中职学生的特点，紧密结合企业生产一线对人才的要求，突出职业技能和实践能力的培养，我们依据教育部最新制定的《中等职业学校电工与电子技术教学大纲（试行）》编写了本教材。本教材具有以下特点；

一、以大纲为基础，从电子类专业教学的需要、中职学生的文化基础、学生毕业后工作的实际出发，坚持“必须、实用、够用”的原则，删繁就简，使教材的难度、深度更加贴近学生，贴近岗位，使教材重点突出，实用性强。

二、本书吸收了大量长期处于中职学校教学一线教师的经验，内容的编排和讲解更加符合学生的认知规律，努力实现理论和实践的有机结合，便于教师的教和学生的学。

三、本教材能与时俱进，具有时代特征，在教材中既充实了新知识、新技术、新理论、新工艺，又采用了新规范、新标准，具有超前性、先进性。

四、为了适应不同专业、不同基础学生的学习，本教材将卡诺图及卡诺图化简法、组合逻辑电路的竞争冒险现象、计数器的设计等内容作为选学内容，教师可根据本校实际选用。

本教材由乐小勇老师、胡勋阳老师担任主编，王仲良老师、江炜老师、方向阳老师、孙宜潮老师担任副主编。参加本书编写的有程琴（第一章），江炜（第二章），胡勋阳、孙宜潮（第三章），乐小勇、王仲良（第四章），汪丹、晏玉华（第五章），方向阳、刘金胜、胡雪梅（第六章），丁梅华（第七章），周万平老师审阅了全部书稿。

本教材在编写过程中得到了黄冈电子信息学校、黄冈劳动技工学校、罗田理工中等专业学校、宜昌一技校等单位的大力支持，在此一并表示诚挚的谢意。

由于时间仓促，加之编者水平有限，书中难免有疏漏之处，恳请各位读者予以批评指正。

编者

2008 年 6 月

目　　录

第 1 章　数字电路基础

学习目标

☞熟悉数字电路中常用数制的计数特点，了解不同数制之间的转换方法，重点掌握二进制和十六进制以及它们之间的转换。

☞理解三种基本逻辑关系的逻辑特点，掌握三种逻辑运算的表示方法、逻辑符号和运算规律。

☞熟悉逻辑代数的基本定律和基本法则。

☞熟悉逻辑问题的三种表示方式。

☞熟悉公式化简法，掌握卡诺图化简法。

1.1　数字电路概述

1.1.1　模拟信号与数字信号

在电子线路中，有两种迥然不同的信号表示方式，一种是用电压的高低不同来形象地模拟自然界中的各种连续变化的信息，这种连续变化的电信号称为模拟信号；另一种是用一系列高低电平的不同组合来代表各种不同的信息，这种离散的带有一定数字编码信息的电信号称为数字信号。

自然界中的大部分信号，如声音、图像、温度和压力等，都是连续变化的信息，通过各种传感器能够较容易地转换成连续变化的模拟信号，但是模拟信号在处理和传递的过程中容易失真，也容易被干扰。

数字信号用两种高低明显不同的电平的不同组合方式代表不同的信息，因为这两种高低不同的电平有明显的差别，当波形因为失真或干扰而有所变形时，仍然可以保持较大的差别来避免出现信号之间的混淆而产生信号失真或受到干扰，所以数字信号在处理和传输的过程中，具有更高的精度和更好的抗干扰能力。譬如数字式移动电话、数字式电视机、数码照相机和摄像机等数字化产品比传统的模拟手机、模拟电视、感光相机以及磁带摄像机等模拟式产品拥有更高的清晰度和稳定性。

但是，自然界中的大部分信息不能直接转换成数字信号，往往需要通过各种传感器把它们转化成模拟信号，再通过模/数转换器转换成数字信号；同样，数字信号通常也不能直接被识别，通常也需要通过数/模转换器转换成模拟信号，再通

过相应的设备输出。

1.1.2 模拟电路和数字电路

处理模拟信号的电路称为模拟电路。它是传统经典的电子技术，其研究热点主要在传感器、高速模/数转换和数/模转换等方面。处理数字信号的电路称为数字电路。在数字电路中，除所有的信号都被数字化以外，其输入信号和输出信号之间又具有一定的逻辑关系，所以又称之为数字逻辑电路，简称数字电路。通过学习我们会发现，数字电路的工作方式、工作任务与模拟电路几乎完全不同，它主要研究的是数字信号的编码、运算、记忆、计数、存储、分配、测量和传输等。从性能上讲，数字电路具有抗干扰性强、精度高、结构简单以及便于集成等优点，它在信号处理和传递、自动控制系统和测量设备等领域获得了广泛的应用。

1.2 数字电路中的常用数制与码制

在分析和设计数字电路的过程中，常常会用到除十进制外的二进制、八进制和十六进制等其他数制。本节将以十进制为参照，重点介绍二进制和十六进制的计数方式、作用和它们之间的转换，同时也将介绍几种常用的编码方法。

1.2.1 数制

数制是人们利用符号来计数的方法，数制的种类较多，但最常用的有十进制，最适宜于数字电路识别和处理的二进制以及为了方便二进制数读数而引入的八进制和十六进制等。

1. 十进制数(Decimal Number)

(1) 组成十进制数的符号有 0,1,2,3,4,5,6,7,8,9，这些符号称为数码。

(2) 十进制在计数时采用“逢十进一”的规律进行累计，当某一位上的数累计到了 10 时，就要向前一位进 1，该位重新回到 0。如 9+1=10，99+1=100。

(3) 十进制数中，不同位置上的数码拥有不同的权值，从右边开始第 n 位数的权为 10^{n-1}，10 为十进制数的基数。每一位数码所代表的大小为该数码乘以它所对应位置的权。例如十进制数 6834，它的每个数码所代表的大小如图 1.2.1 所示。

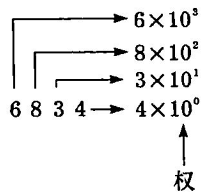


图 1.2.1 十进制数 6834 的每个数码所代表的大小

(4) 任一十进制数可表示为：

$$N_{10} = a_{n-1} \times 10^{n-1} + a_{n-2} \times 10^{n-2} + \cdots + a_1 \times 10^1 + a_0 \times 10^0 + a_{-1} \times 10^{-1} + a_{-2} \times 10^{-2} + \cdots + a_{-m} \times 10^{-m}$$

$$= \sum_{i=-\infty}^{\infty} (a_i \times 10^i)$$

式中：a_i为 0 ～ 9 中的任一十进制数码，10 为十进制数的基数，10^i 为第 i 位数的权，n 为整数部分的位数，m 为小数部分的位数，m，n 为正整数。

2. 二进制数(Binary Number)

(1) 二进制只有两个不同的数码，即 0 和 1。

(2) 二进制采用“逢二进一”的规律进行累计，当某一位上的数累计到 2 时，就要向前一位进 1，该位重新回到 0。如二进制数 1＋1＝10，11＋1＝100，计算方法如图 1.2.2 所示。

$$\begin{array}{r} 1 \\ +1 \\ \hline 1\ 0 \end{array} \qquad \begin{array}{r} 1\ 1 \\ +\quad 1 \\ \hline 1\ 0\ 0 \end{array}$$

图 1.2.2　二进制加法计算方法

(3) 二进制数的基数为 2，从右边开始第 n 位数的权为 2^{n-1}。与十进制数相类似，任意一个二进制数可表示为：

$$N_2 = a_{n-1} \times 2^{n-1} + a_{n-2} \times 2^{n-2} + \cdots + a_1 \times 2^1 + a_0 \times 2^0 + a_{-1} \times 2^{-1}$$
$$+ a_{-2} \times 2^{-2} + \cdots + a_{-m} \times 2^{-m} = \sum_{i=-\infty}^{\infty} (a_i \times 2^i)$$

例如：$(10011)_2 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$

(4) 数字电路采用二进制的优点

①数制简单、容易表示

二进制数只有“**0**”和“**1**”两种数码，任何具有两个不同稳定状态的元件，都可以用来表示二进制数的每一位。而制造具有两个稳定状态的元件要比制造多稳定状态(如 10 个稳定状态)的元件容易得多，如晶体管的导通和截止、电容的充电和放电、磁芯两个不同状态的磁化等。在数字电路中通常采用电平的“高”、“低”或脉冲的“有”、“无”来表示“**1**”和“**0**”。这种简单的工作状态可靠，且抗干扰能力强。

②运算规则简单

二进制的运算规则非常简单，这使得在计算机中实现二进制运算的线路大为简化。

③节省设备

若采用十进制数，则有 0～9 十个数码，表示一个数位共需 10 个完全不同的设备状态。而采用二进制数来表示十进制数，则一位十进制数可用 4 位二进制数来表示，二进制数的每一位只有两个状态，共有 8 个设备状态，其表示范围为 **0000** ～**1111**，即 0～15。这说明采用二进制数，可以节省设备。

尽管在计算机内部采用二进制操作，但使用二进制并不方便，其书写冗长，阅读也不方便，为此通常用八进制或十六进制数来表示二进制数。

3. 十六进制(Hexadecimal Number)

十六进制是计算机系统中除二进制数以外使用较多的进制，与前面两种数制相比有三个方面的特点：

(1) 十六进制有 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F 共 16 个数码，其分别对应于十进制数的 0～15；

(2) 十六进制采用“逢十六进一”的规律进行累计，当某一位上的数累计到 F 时，就要向前一位进 1，该位重新回到 0。如十六进制数 F+1=10，FF+1=100。

(3) 十六进制数的基数为 16，从右边开始第 n 位数的权为 16^{n-1}。十六进制数所代表的大小也可以用下面的式子表示：

$$N_{16}=a_{n-1}\times 16^{n-1}+a_{n-2}\times 16^{n-2}+\cdots+a_1\times 16^1+a_0\times 16^0$$
$$+a_{-1}\times 16^{-1}+a_{-2}\times 16^{-2}+\cdots+a_{-m}\times 16^{-m}=\sum_{i=-\infty}^{\infty}a_i\times 16^i$$

在使用不同数制时，常将各种数制用简码来表示：如十进制数用 D 表示或省略；二进制数用 B 来表示；十六进制数用 H 来表示。

如：十进制数 123 表示为：123D 或者 123；二进制数 1011 表示为：1011B；十六进制数 3A4 表示为：3A4H。

在计算机中除上面讲到的二进制、十进制、十六进制外，常常还会用到八进制数，这里就不讨论了。

十进制数 0～15 对应的二进制数和十六进制数如表 1.2.1 所示。

表 1.2.1　0～15 十进制数、二进制数、十六进制数

十进制数	二进制数	十六进制数
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

1.2.2　各种进制的相互转换

1. 其他进制转换为十进制

方法是：将其他进制按权位展开，然后各项相加，就得到相应的十进制数。

[例 1.2.1]　$N=(10110.101)_B=(\quad)_D$

解　按权展开 $N=1\times2^4+0\times2^3+1\times2^2+1\times2^1+0\times2^0+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}=16+4+2+0.5+0.125=(22.625)_D$

2. 将十进制转换成其他进制

方法是：将十进制数分成整数部分和小数部分分别进行。

整数部分：（基数除法）

(1) 把要转换的数除以新进制的基数，把余数作为新进制数的最低位；

(2) 把上一次得的商再除以新进制的基数，把余数作为新进制数的次低位；

(3) 重复上一步骤，直到最后的商为零，这时的余数就是新进制数的最高位。

小数部分：（基数乘法）

(1) 把要转换数的小数部分乘以新进制的基数，把得到的整数部分作为新进制小数部分的最高位；

(2) 把上一步得的小数部分再乘以新进制的基数，把整数部分作为新进制小数部分的次高位；

(3) 重复上一步骤，直到小数部分变为零，或者达到预定的精度要求。

[例 1.2.2]　$N=(47.37)_D=(\quad)_B$，保留小数点后四位。

解　整数部分

取余数部分

$$\begin{array}{r|l l l}
2 & \underline{47} & & \\
2 & \underline{23} & \cdots 1 & (低位) \\
2 & \underline{11} & \cdots 1 & \\
2 & \underline{5} & \cdots 1 & \uparrow 读数方向 \\
2 & \underline{2} & \cdots 1 & \\
2 & \underline{1} & \cdots 0 & \\
 & 1 & \cdots 1 & (高位)
\end{array}$$

小数部分

取整数部分

$$\begin{array}{r l l}
0.37 & & \\
\times\quad 2 & & \\
\hline
0.74 & \cdots 0 & (高位) \\
\times\quad 2 & & \\
\hline
1.48 & \cdots 1 & \\
0.48 & & \downarrow 读数方向 \\
\times\quad 2 & & \\
\hline
0.96 & \cdots 0 & \\
\times\quad 2 & & \\
\hline
1.92 & \cdots 1 & (低位)
\end{array}$$

所以 $N=(47.37)_D=(101111.0101)_B$

［**例 1.2.3**］ $N=(347.78125)_D=(\quad)_H$

解 整数部分　　　　　　　　　　小数部分

取余数部分

$$
\begin{array}{r|l}
16 & 347 \\
16 & 21 \cdots 11(B) \\
16 & 1 \cdots 5 \\
 & 1 \cdots 1
\end{array}
$$

（低位）↑ 读数方向（高位）

取整数部分

$$
\begin{array}{r}
0.78125 \\
\times \quad 16 \\
\hline
12.5 \\
0.5 \\
\times \quad 16 \\
\hline
8.0
\end{array}
\quad
\begin{array}{l}
\cdots 12(C) \\
\\
\\
\cdots 8
\end{array}
$$

（高位）↓ 读数方向（低位）

所以 $N=(347.78125)_D=(15B.C8)_H$

3. 二进制与十六进制的互换

二进制数转换成十六进制数的方法：每四位二进制数与一位十六进制数互换。

(1) 将二进制数从小数点处开始，分别向两边每四位分成一小节，小数点左边最后一节不够四位时，向数的最左边添 0 凑成四位，小数点右边最后一节不够四位时，向数的最右边添 0 凑成四位。

(2) 将每一小节按表 1.2.1 的对应关系变成一个十六进制数，将这些数按顺序排列即可。

将十六进制数转换成二进制数的方法：

(1) 将每一位十六进制数按表 1.2.1 的关系变成四位二进制数；

(2) 将整数最左边和小数最右边的 0 去掉即可。

［**例 1.2.4**］ 完成下列数制的转换。

1. $N=(3D5.C)_H=(\quad)_B$

解 $(3D5.C)_H=(0011,1101,0101.1100)_B=(1111010101.11)_B$

2. $N=(101011.111)_B=(\quad)_H$

解 $N=(101011.111)_B=(0010,1011.1110)_B=(2B.E)_H$

1.2.3 常用编码

在数字系统中，各种数据要转换为二进制代码才能进行处理，而人们习惯了使用十进制数，所以在数字系统的输入输出中仍采用十进制数，这样就产生了用四位二进制数表示一位十进制数的方法，人们将用于表示十进制数的二进制代码称为二一十进制代码(Binary Coded Decimal)，简称为 BCD 码。它具有二进制数的形式以满足数字系统的要求，又具有十进制数的特点（只有十种有效状态）。在某些情况下，计算机也可以对这种形式的数直接进行运算。常见的 BCD 码表示有两类，一类是有权码，一类是无权码。

1. 8421BCD 编码

这是一种使用最广的有权 BCD 码，其各位的权分别是（从最高有效位开始到最低有效位）8，4，2，1。其编码规则如表 1.2.2 所示。

表 1.2.2　常见 BCD 编码表

十进制数	8421BCD 码	2421BCD 码	余 3 码
0	0000	0000	0011
1	0001	0001	0100
2	0010	0010	0101
3	0011	0011	0110
4	0100	0100	0111
5	0101	1011	1000
6	0110	1100	1001
7	0111	1101	1010
8	1000	1110	1011
9	1001	1111	1100
10	0001 0000	0001 0000	0100 0011

[例 1.2.5]　写出十进制数$(563.97)_D$对应的 8421BCD 码。

解　$(563.97)_D=(0101\ 0110\ 0011.1001\ 0111)_{8421BCD}$

[例 1.2.6]　写出 8421BCD 码$(1101001.01011)_{8421BCD}$对应的十进制数。

解　$(1101001.01011)_{8421BCD}=(0110\ 1001.0101\ 1000)_{8421BCD}=(69.58)_D$

在使用 8421BCD 码时一定要注意其有效的编码仅有十个，即：0000～1001。四位二进制数的其余六个编码 1010,1011,1100,1101,1110,1111 不是有效编码。

2. *2421BCD 编码*

2421BCD 码也是一种有权码，从高位到低位的权分别为 2,4,2,1，它也可以用四位二进制数来表示一位十进制数。其编码规则如表 1.2.2 所示。

3. *余 3 码*

余 3 码也是一种 BCD 码，但它是无权码，由于每一个码与对应的 8421BCD 码之间相差 3，故称为余 3 码，这种编码一般较少使用，故只作一般性了解，具体的编码规则如表 1.2.2 所示。

4. *格雷码*

格雷码是一种无权码，其特点是任意两个相邻的码之间只有一个数不同。另外由于最大数与最小数之间也仅一个数不同，故通常又叫格雷反射码或循环码。格雷码与十进制数和二进制数之间的对应关系如表 1.2.3 所示。

表 1.2.3　格雷码与十进制数、二进制数的对应关系

十进制数	二进制数	格雷码	十进制数	二进制数	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

1.3 逻辑代数基础

逻辑代数是分析和设计逻辑电路的重要理论基础，它是由英国科学家乔治·布尔(George Boole)于1956年创立的，故又称为布尔代数。逻辑代数是借用代数的形式来反映事物之间的逻辑关系，与数学中的代数没有任何实质上的联系。

逻辑关系是指客观事物的内在因果关系，或者说条件和结果的关系，这些因果关系在逻辑代数中可以用特定的逻辑表达式来描述。逻辑代数中，将影响事情结果的每一个条件设为输入逻辑变量，常用 A、B、C、D 等大写字母表示，结果设为输出逻辑变量，常用大写字母 F、F_1、F_2、F_3、…或 Y、Y_1、Y_2、Y_3、…表示。事物的每一个条件和结果往往存在两种对立的状态，在逻辑代数中可以抽象地表示为逻辑0和逻辑1。在二值逻辑中，逻辑变量的取值只有两种，即逻辑0和逻辑1。它们并不表示数量的大小，而是表示两种对立的逻辑状态。

在逻辑代数中，所有的逻辑关系都可以用与、或、非三种基本逻辑关系或他们的组合来表示，通过整理和化简之后，可以用相应的数字逻辑电路来实现，所以数字电路的输入输出信号之间往往存在一定的逻辑对应关系，并且都是建立在三种最基本的逻辑电路(逻辑“与”门电路、逻辑“或”门电路和逻辑“非”门电路)基础上，他们分别反映了最基本的三种逻辑关系：逻辑“与”、逻辑“或”和逻辑“非”。

1.3.1 逻辑“与”(AND Logic)

1. 逻辑“与”关系

如果决定某一事件发生的多个条件必须同时具备时事件才能发生，则称这种因果关系为“与”逻辑。

例如，用两个开关串联控制同一盏电灯的照明控制电路如图1.3.1(a)所示。显然，仅当两个开关均闭合时，灯才能亮，这个电路的两个开关与灯之间的关系就是逻辑“与”的关系。

2. “与”运算

在逻辑代数中，“与”逻辑关系用“与”运算来描述。两个变量的“与”运算关系可表示为：

$$F=A\cdot B$$

其中 A,B 为输入逻辑变量，代表客观事物的两个条件，F 为输出逻辑变量，代表事物的结果。如果用逻辑**1**代表成立，逻辑**0**则代表不成立。用**1**和**0**表示所有可能出现的输入情况和对应的输出结果的表格称为真值表，如图1.3.1(b)所示。

在“与”逻辑的真值表中，F 与 A,B 之间的关系与数学中的乘法运算关系相

似，于是"与"逻辑关系就借用数学里面的乘法表达式 $F = A \cdot B$ 来表示。正因如此，人们又把逻辑"与"称为逻辑乘。

3. "与"运算的运算法则

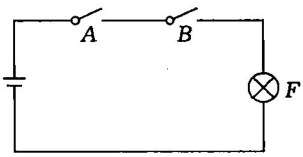
根据逻辑"与"的运算特点可以推导出下面的逻辑运算法则：

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

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

4. "与"门电路的逻辑符号

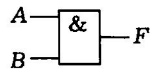
数字电路中，实现"与"运算关系的逻辑电路称为"与"门电路，简称"与"门，其逻辑符号如图 1.3.1(c)所示。



(a)电路

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

(b)真值表



(c)逻辑符号

图 1.3.1　"与"逻辑

1.3.2　逻辑"或"(OR Logic)

1. 逻辑"或"关系

如果决定某一事件是否发生的多个条件中只要有一个或一个以上条件成立事件便发生，则称这种因果关系为"或"逻辑。

例如，用两个开关并联控制一盏电灯的照明控制电路如图 1.3.2(a)所示。显然，两个开关只要有一个闭合，灯就能亮，这个电路的两个开关与灯的状态之间的关系就是逻辑"或"的关系。

2. "或"运算

在逻辑代数中，两变量"或"运算关系可表示为：

$$F = A + B$$

其真值表如图 1.3.2(b)所示。

从真值表中可以看出：除了第四种情况外，F 与 A，B 之间的关系正好与数学中的加法相似，所以逻辑"或"关系借用了数学中加法的表达式 $F = A + B$。

3. "或"运算的运算法则

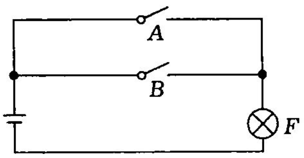
根据逻辑"或"的关系特点可以推导出下面的逻辑运算法则：

$0 + 0 = 0 \quad 1 + 0 = 1 \quad 0 + 1 = 1 \quad 1 + 1 = 1$

$0 + A = A \quad 1 + A = 1 \quad A + A = A$

4. “或”门电路的逻辑符号

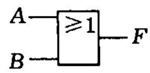
实现“或”运算关系的逻辑电路称为“或”门电路，简称“或”门，其逻辑符号如图 1.3.2(c)所示。



(a)电路

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

(b)真值表



(c)逻辑符号

图 1.3.2 “或”逻辑

1.3.3 逻辑“非”(NOT Logic)

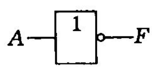
结果与条件相反的逻辑关系称为逻辑“非”。在逻辑代数中，“非”运算的表达式为：

$$F=\overline{A}$$

其真值表如图 1.3.3(a)所示。

A	F
0	1
1	0

(a)真值表



(b)逻辑符号

图 1.3.3 “非”逻辑

逻辑“非”的运算法则如下：

$$\overline{1}=0 \quad \overline{0}=1 \quad \overline{\overline{A}}=A \quad A+\overline{A}=1 \quad A\cdot\overline{A}=0$$

实现“非”运算功能的逻辑电路称为“非”门电路，简称“非”门，有时又称为“反相器”，其逻辑符号如图 1.3.3(b)所示。

1.4 逻辑代数的基本定律和运算规则

逻辑代数的基本定律是分析、设计逻辑电路，化简和变换逻辑函数式的重要工具。这些定律和普通代数看起来相似，但更有其独特性。

1.4.1 与普通代数相似的定律——交换律、结合律和分配律

逻辑代数的基本定律如表 1.4.1 所示。