21st CENTURY
实用规划教材

21世纪全国高职高专
电子信息系列实用规划教材

数字电子技术

主 编 房永钢 王树红

北京大学出版社
PEKING UNIVERSITY PRESS

21世纪全国高职高专电子信息系列实用规划教材

数字电子技术

主　编　房永钢　王树红
副主编　杜保强　李　静

北京大学出版社
PEKING UNIVERSITY PRESS

内 容 简 介

本书是21世纪全国高职高专电子信息系列实用规划教材之一，其内容包括逻辑代数基础、逻辑门电路、组合逻辑电路、触发器、时序逻辑电路、D/A和A/D转换、脉冲波形的产生与整形电路和实验共8章。

本书可作为高职高专院校电子类专业电子技术基础课程的教材，也可作为高职高专院校非电子类专业电子技术基础课程的参考书，还可作为电大、函大、夜大等成人教育同类课程的教材或参考书。

图书在版编目(CIP)数据

数字电子技术/房永钢，王树红主编. —北京：北京大学出版社，2009.7

(21世纪全国高职高专电子信息系列实用规划教材)

ISBN 978-7-301-12391-1

Ⅰ. 数… Ⅱ. ①房…②王… Ⅲ. 数字电路—电子技术—高等学校：技术学校—教材 Ⅳ. TN79

中国版本图书馆CIP数据核字(2007)第083341号

书　　名：数字电子技术
著作责任者：房永钢　王树红　主编
责任编辑：赖　青
标准书号：ISBN 978-7-301-12391-1/TM・0015
出版者：北京大学出版社
地　　址：北京市海淀区成府路205号　100871
网　　址：http://www.pup.cn　http://www.pup6.com
电　　话：邮购部62752015　发行部62750672　编辑部62750667　出版部62754962
电子邮箱：pup_6@163.com
印刷者：河北滦县鑫华书刊印刷厂
发行者：北京大学出版社
经销者：新华书店
787毫米×1092毫米　16开本　14.5印张　336千字
2009年7月第1版　2009年7月第1次印刷
定　　价：24.00元

前　言

为贯彻高职高专教育由“重视规模发展”转向“注重提高质量”的工作思路，适应我国当前高职高专教育教学改革和教材建设的需要，培养以就业为导向的具备职业化特征的高等技术应用型人才，我们组织编写了本书，具体考虑了以下几个方面。

(1) 教材的基本内容紧扣教学大纲，对基本概念、基本原理和基本分析方法进行由浅入深、准确透彻的阐述。

(2) 在确保基础内容教学的同时，注重理论与实践相结合。

(3) 将有关技术进步的新成果、新应用纳入教学内容中，妥善处理传统内容的继承与现代内容的引入。

参加本书编写工作的有：河南职业技术学院的杜保强（第 1 章），太原大学的王树红（第 2、3 章），河南机电高等专科学校的李静（第 5 章），山东济宁学院的房永钢（第 4、6、7、8 章）。

本书建议总课时为 80 学时，各章建议课时分配见下表。

章　　节	建议课时
第 1 章	12 学时
第 2 章	8 学时
第 3 章	12 学时
第 4 章	10 学时
第 5 章	10 学时
第 6 章	8 学时
第 7 章	12 学时
第 8 章	8 学时

限于编者水平，加之时间仓促，书中不当之处在所难免，恳请读者批评指正。

编　者

2009 年 5 月

目　录

第 1 章　逻辑代数基础

教学提示：逻辑代数是分析和设计数字电路的基本数学工具。本章首先介绍数字电路的一些基本概念及数字电路中常用的数制与编码；然后讨论数字电路中二极管、三极管的工作方式、数字电路中的基本逻辑运算、逻辑函数及其表示方法；最后介绍逻辑代数的基本概念、公式、定理及逻辑函数的代数化简法和卡诺图化简法。

教学要求：通过本章学习，要了解数制和编码；掌握 3 种基本逻辑运算和常用逻辑运算；掌握逻辑代数的公式及定理；掌握逻辑函数的表示方法及相互间的转换方法；掌握最小项的定义、性质、表示方法及逻辑相邻的含义；掌握约束、约束条件和约束项的概念；掌握逻辑函数的标准与或式表示法；掌握代数化简法、卡诺图化简法及具有无关项的逻辑函数的化简方法。

1.1　数字电路概述

1.1.1　模拟信号和数字信号

所谓信号，是指实际问题中遇到的运动或变化的各种量(电压、电流、光强、噪声、力、湿度、温度、位移等)。在电子电路系统中，可以通过合适的传感器将信号转变为随时间变化的电压或者电流。电子电路中的信号分类方法很多，可以根据信号的周期性、确定性、频率特性、能量特性、时域特性等分类，其中根据信号的时域特性可以分为两大类：模拟信号和数字信号。

1. 模拟信号

模拟信号是时间上连续、数值上也连续的电压或者电流信号，如音乐、图像、压力、温度等实际的物理信号经过相应的传感器转变为电信号。模拟信号一般用 $u(t)$或者 $i(t)$表示，其特点是任意时刻信号都有定义，即定义域是$(-\infty , +\infty)$。处理模拟信号的一类电路可以通称为模拟电路。

2. 数字信号

数字信号是时间上和数值上均是离散的电压或者电流信号，如电子表的秒信号、生产流水线上记录零件个数的计数信号等。这些信号的变化发生在一系列离散的瞬间，其值也是离散的，特点是信号仅在一些特定的时刻有意义。数字信号只有两个离散值，常用数字 0 和 1 来表示，注意，这里的 0 和 1 没有大小之分，只代表两种对立的状态，称为逻辑 0 和逻辑 1，也称为二值数字逻辑。

对信号的描述可以采用数学函数式或者图形。信号的图形也称为信号的波形，当用波形描述一个信号时，应注意在波形图上标出该信号的关键值，关键值包括信号的不连续点、零点、最大值点和最小值点等。直观地讲，连续即没有间断，也就是说信号的自变量可以

在其定义域内取任意值，如 0.00325、7 等；离散则是有间断，自变量只能定义在一组离散的规定值上，如整数值等，而不能在其定义域内取任意值，在规定值以外的数值上，自变量是没有意义的。显然，由于自变量的连续性，连续信号的波形是连续的，其函数值可以是连续的任意值，也可以是不连续的规定值。同样，对于数字信号而言，由于自变量的离散性，其信号波形是离散的，但信号的函数值可以是连续的任意值或不连续的规定值。数字信号在电路中往往表现为突变的电压或电流，如图 1.1 所示。该信号有以下特点。

(1) 信号只有两种电压值，5V 以上和 0V 以下。

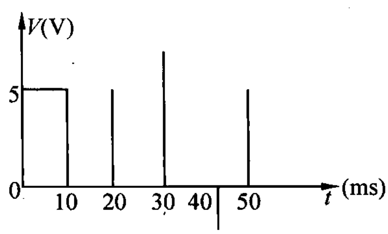


图 1.1　典型的数字信号

在数字逻辑电路中，经常用逻辑 1 来表示 5V 以上的电压，0V 以下的电压用逻辑 0 表示，这两个电压值又常被称为**逻辑电平**。5V 为高电平，0V 为低电平。当然也可以用逻辑 1 来表示 0V 以下的电压，用逻辑 0 来表示 5V 以上的电压。

(2) 信号从高电平变为低电平，或者从低电平变为高电平是一个突然变化的过程，期间无定义。

1.1.2　正逻辑与负逻辑

如上所述，数字信号是一种二值信号，用两个电平(高电平和低电平)分别表示两个逻辑值(逻辑 1 和逻辑 0)。在数字电路中，有两种逻辑体制：

(1) 正逻辑体制规定：高电平为逻辑 1，低电平为逻辑 0。

(2) 负逻辑体制规定：低电平为逻辑 1，高电平为逻辑 0。

1.1.3　数字电路的特点

数字电路(又称为数字电子技术)就是传递与处理(编码、译码、存储等)数字信号的电子电路，模拟电子技术是分析和处理模拟信号的技术。为了分析问题的方便起见，一般认为数字信号的典型代表是矩形脉冲信号。同时，信号的傅里叶分析证明，任何复杂的信号都是由不同频率的正弦信号叠加而成的，因此，正弦信号是模拟信号的典型代表。在数字电子技术中，使用的二极管、三极管等一般工作于开关状态，而在模拟电子技术中，使用的二极管、三极管等一般工作于放大状态。数字电路与模拟电路相比主要有下列优点。

(1) 由于数字电路是以二值数字逻辑为基础的，只有 0 和 1 两个基本数字，易于用电路来实现，比如可用二极管、三极管的导通与截止这两个对立的状态来表示数字信号的逻辑 0 和逻辑 1。

(2) 由数字电路组成的数字系统工作可靠，精度较高，抗干扰能力强。它可以通过整形很方便地去除叠加于传输信号上的噪声与干扰，还可利用差错控制技术对传输信号进行查错和纠错。

(3) 数字电路不仅能完成数值运算，而且能进行逻辑运算，这在控制系统中是不可缺少的。

(4) 数字信息便于长期保存，比如可将数字信息存入磁盘、光盘中长期保存。

(5) 数字电路产品系列多、通用性强、成本低。

由于具有这一系列优点，数字电路在电子设备或电子系统中得到了越来越广泛的应用，

计算机、计算器、电视机、音响系统、视频记录设备、光碟、长途电信及卫星系统等，无一不采用数字系统。

1.1.4 数字电路的发展与分类

数字电路的结构是以二值数字逻辑为基础的，其中的工作信号是离散的数字信号。电路中的电子器件，如二极管、三极管(BJT、FET)处于开关状态，时而导通，时而截止。

数字电路的发展历史与模拟电路一样，经历了由电子管、半导体分立器件到集成电路的过程，但数字集成电路比模拟集成电路的发展更快。从 20 世纪 60 年代开始，数字集成器件以双极型工艺制成了小规模逻辑器件，随后发展到中规模逻辑器件；20 世纪 70 年代末，微处理器的出现使数字集成电路的性能产生了质的飞跃。

数字集成器件所用的材料以硅为主，在高速电路中，也使用化合物半导体材料，例如砷化镓等。

逻辑门是一种重要的逻辑单元电路。TTL 逻辑门电路问世较早，其工艺经过不断改进，至今仍为主要的基本逻辑器件之一。随着 MOS 工艺特别是 CMOS 工艺的发展，TTL 的主导地位有被 CMOS 器件取代的趋势。

近年来，可编程逻辑器件(PLD)特别是现场可编程门阵列(FPGA)的飞速发展，为数字电子技术开创了新局面。PLD 不仅规模大，而且将硬件与软件相结合，使器件的功能更加完善，使用也更加灵活。

从集成度来说，数字集成电路可分为小规模、中规模、大规模、超大规模和甚大规模 5 类。所谓集成度，是指每个芯片所包含的三极管(BJT 或 FET)的个数。表 1-1 列出了 5 类数字集成电路的分类依据。

从表 1-1 可以看到，存储器是基本数字器件之一，集成度也很高。利用存储器可以记忆或存储二值数字 1 或 0。存储的数字信息可以被取出来分析或直接利用，例如，打印机可从计算机的存储器里取出信息并打印在纸上。通常数字信息的存储视为将信息写入存储器，而信息恢复则理解为从存储器读出信息。

表 1-1 数字集成电路的分类

分　类	三极管的个数	典型集成电路
小规模	最多 10 个	逻辑门电路
中规模	10～100	计数器、加法器
大规模	100～1000	小型存储器、门阵列
超大规模	1000～10^6	大型存储器、微处理器
甚大规模	10^6 以上	可编程逻辑器件、多功能集成电路

1.2 数制及转换

在数字电路和计算机中，只用 0 和 1 两种符号来表示信息，参与运算的数也是由 0 和 1 构成的，称为二进制数。考虑到人类计数习惯，计算机在进行操作时，一般都要把输入的十进制数转换为二进制数后再来处理；而计算机处理的二进制结果也需要转换为便于人

类识别的十进制数后显示出来。因此，需要学习不同的数制及其转换方法。

1.2.1 几种常用的数制

用数字量表示物理量的大小时，一位数码往往不够用，因此经常需要用进位计数的方法组成多位数码使用。把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称为数制。常用的数制有十进制、二进制、八进制和十六进制。

1. 十进制(Decimal)

在十进制中，每个数位规定使用的数码为 0~9，共 10 个数，故其进位基数 R 为 10。其计数规则是"逢十进一"。各位的权值为 10^i，i 是各数位的序号。十进制数用下标 D 表示，也可省略。例如

$$(143.75)_D = 1\times10^2 + 4\times10^1 + 3\times10^0 + 7\times10^{-1} + 5\times10^{-2}$$

所以，任意一个正的十进制数 N 都可以展开成

$$(N)_D = \sum k_i 10^i$$

其中 k_i 是第 i 位的系数，它可能是 0～9 这 10 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。不失一般性，对于任意 R 进制数，其展开式的一般形式为

$$(N)_R = \sum k_i R^i$$

2. 二进制(Binary)

在二进制中，每个数位规定使用的数码为 0 和 1，共两个数码，其进位基数 R 为 2，计数规则是"逢二进一"，即"1+1=10"。各位的权值为 2^i。二进制数用下标 B 表示。例如

$$(101.11)_B = 1\times2^2 + 0\times2^1 + 1\times2^0 + 1\times2^{-1} + 1\times2^{-2}$$

这里，式子中的 1 可以省略。由于二进制数只需两个状态，计算机容易实现，因而二进制数码是数字系统唯一认识的代码，但其不足是书写太长。

3. 十六进制(Hexadecimal)与八进制(Octal)

在十六进制(或者八进制)中，每个数位上规定使用的数码为 0~9，和 A、B、C、D、E、F 共 16 个数(或者 0~7 共 8 个数)，故其进位基数 R 为 16。其计数规则为"逢十六进一"(或者"逢八进一")，即"1+F=10"(或者"1+7=10")。八进制数用下标 O 表示，十六进制数用下标 H 表示。例如

$$(AB68)_H = 10\times16^3 + 11\times16^2 + 6\times16^1 + 8\times16^0$$

$$(37.41)_O = 3\times8^1 + 7\times8^0 + 4\times8^{-1} + 1\times8^{-2}$$

因为 $2^4=16$，所以 4 位二进制数可用一位十六进制数表示。在计算机应用系统中，二进制主要用于内部的数据处理，八进制和十六进制主要用于书写程序，十进制主要用于运算最终结果的输出。

1.2.2 不同数制之间的相互转换

由于目前的计算机内部处理数据只用二进制数，因此，应该知道不同数制之间的相互转换。

数制转换一般是二进制与十进制的相互转换、二进制与十六进制的相互转换、十六进制与十进制的相互转换、八进制与十进制的相互转换。

1. 二进制转换成十进制

把二进制数转换为等值的十进制数称为二-十转换。具体方法举例说明。

【例 1.1】 将二进制数 10011.101 转换成十进制数。

解：将每一位二进制数乘以位权，然后相加，可得

$$(1011.101)_B=1\times2^3+0\times2^2+1\times2^1+1\times2^0+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}=(11.625)_D$$

2. 十进制转换成二进制

把十进制数转换为等值的二进制数称为十-二转换。具体方法举例说明。

【例 1.2】 将十进制数 23 转换成二进制数。

解：(1) 整数部分的转换：可用"除 2 取余"法将十进制的整数部分转换成二进制数。根据"除 2 取余"法的原理，按如下步骤转换

$$\begin{array}{r|l}
2 & 23 \quad \cdots\cdots 余1 \quad b_0 \\ \hline
2 & 11 \quad \cdots\cdots 余1 \quad b_1 \\ \hline
2 & 5 \quad \cdots\cdots 余1 \quad b_2 \\ \hline
2 & 2 \quad \cdots\cdots 余0 \quad b_3 \\ \hline
2 & 1 \quad \cdots\cdots 余1 \quad b_4 \\ \hline
 & 0
\end{array}
\quad \Big\uparrow \text{读取次序}$$

则$(23)_D=(10111)_B$

(2) 纯小数部分转换：可用"乘 2 取整"的方法将任何十进制数的纯小数部分转换成二进制数。

【例 1.3】 将十进制数$(0.562)_D$转换成误差ε不大于2^{-6}的二进制数。

解：用"乘 2 取整"法，按如下步骤转换

$$\begin{array}{lcl}
 & & \text{取整} \\
0.562\times2=1.124 & \cdots\cdots & 1 \ \cdots\cdots b_{-1} \\
0.124\times2=0.248 & \cdots\cdots & 0 \ \cdots\cdots b_{-2} \\
0.248\times2=0.496 & \cdots\cdots & 0 \ \cdots\cdots b_{-3} \\
0.496\times2=0.992 & \cdots\cdots & 0 \ \cdots\cdots b_{-4} \\
0.992\times2=1.984 & \cdots\cdots & 1 \ \cdots\cdots b_{-5}
\end{array}$$

由于最后的小数 0.984＞0.5，根据“四舍五入”的原则，b_{-6}应为 1。因此

$$(0.562)_D=(0.100011)_B$$

其误差$\varepsilon<2^{-6}$。

3. 二进制转换成十六进制

由于十六进制基数为 16，而$16=2^4$，因此，4 位二进制数就相当于 1 位十六进制数，故可用“4 位分组”法将二进制数化为十六进制数。

注意：对于不足 4 位的二进制数，整数最高位前面补零，小数最低位后面补零处理。

【例 1.4】 将二进制数 1001101.100111 转换成十六进制数。

解：$(1001101.100111)_B=(0100\ 1101.1001\ 1100)_B=(4D.9C)_H$

同理，若将二进制数转换为八进制数，可将二进制数分为 3 位一组，不足 3 位的二进制数处理方法类似于 4 位二进制数。再将每组的 3 位二进制数转换成 1 位八进制数即可。

4. 十六进制转换成二进制

由于每位十六进制数对应于 4 位二进制数，因此，十六进制数转换成二进制数，只要将每一位变成 4 位二进制数，按位的高低依次排列即可。

【例 1.5】 将十六进制数 6E.3A5 转换成二进制数。

解：$(6E.3A5)_H=(110\quad 1110.0011\quad 1010\quad 0101)_B$

同理，若将八进制数转换为二进制数，只需将每一位变成 3 位二进制数，按位的高低依次排列即可。

5. 十六进制转换成十进制

可用“按权相加”法将十六进制数转换为十进制数。

【例 1.6】 将十六进制数 7A.58 转换成十进制数。

解：
$$\begin{aligned}(7A.58)_H&=7\times16^1+10\times16^0+5\times16^{-1}+8\times16^{-2}\\&=112+10+0.3125+0.03125=(122.34375)_D\end{aligned}$$

同样的方法可以将十进制转换成十六进制，这里不再讨论。

1.3 二-十进制码

由于数字系统是以二值数字逻辑为基础的，因此，数字系统中的信息(包括数值、文字、控制命令等)都是用一定位数的二进制码表示的，这个二进制码称为代码。

二进制编码方式有多种，常见的有二-十进制编码、可靠性编码(格雷码、奇偶校验码)、字符编码(ASCII 码)等。二-十进制码(用二进制代码来表示十进制的 0～9 这 10 个数)又称为 BCD 码(Binary-Coded-Decimal)，它是数字编码中常用的一种编码方式。要用二进制代码来表示十进制的 0～9 这 10 个数，至少要用 4 位二进制数。4 位二进制数有 16 种组合，可从这 16 种组合中选择 10 种组合分别来表示十进制的 0～9 这 10 个数。选哪 10 种组合，有多种方案，这就形成了不同的 BCD 码。具有一定规律的常用 BCD 码有 8421 码、2421 码、5421 码和余 3 码等，具体见表 1-2。

表 1-2　常用 BCD 码一览表

十进制数	8421 码	2421 码	5421 码	余 3 码	$G_3G_2G_1G_0$ 格雷码
0	0000	0000	0000	0011	0000
1	0001	0001	0001	0100	0001
2	0010	0010	0010	0101	0011
3	0011	0011	0011	0110	0010
4	0100	0100	0100	0111	0110
5	0101	1011	1000	1000	0111
6	0110	1100	1001	1001	0101
7	0111	1101	1010	1010	0100
8	1000	1110	1011	1011	1100
9	1001	1111	1100	1100	1101
10					1111
11					1110
12					1010
13					1011
14					1001
15					1000

注意：BCD 码用 4 位二进制码表示的只是十进制数的一位。如果是多位十进制数，应首先将每一位用 BCD 码表示，然后再组合起来。

(1) 8421BCD 码的规律：最高位是 8，次高位是 4，次低位是 2，最低位是 1。编码按照自然二进制数的规律编排。

(2) 5421BCD 码的规律：最高位是 5，次高位是 4，次低位是 2，最低位是 1。数 0～4 的编码与 8421BCD 码的规律相同，5～9 的编码按照最高位优先的规律编排。例如，7 的编码是 1010。

(3) 2421BCD 码的规律：最高位是 2，次高位是 4，次低位是 2，最低位是 1。数 0～4 的编码与 8421BCD 码的规律相同，5～9 的编码按照最高位优先的规律编排。例如，7 的编码是 1101。

(4) 余 3 码的规律：无权码，编码按照 8421BCD 码加 3 的规律编排。

(5) 格雷码的规律：无权码，编码按照逻辑相邻的规律编排。见表 1-2 和图 1.2 所示。

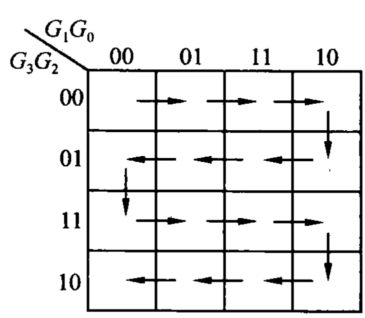


图 1.2　四变量卡诺图

【例 1.7】　将十进制数 83 分别用 8421 码、2421 码和余 3 码表示。

解：由表 1-2 可得

$(83)_D=(1000\ 0011)_{8421}$

$(83)_D=(1110\ 0011)_{2421}$

$(83)_D=(1011\ 0110)_{余3}$

编码表 1-2 中 4 位无权码叫格雷码(Gray)，看似无规律，实际上，它是按照“相邻性”编码原则实现的，即相邻两码之间只有一位数字不同。

格雷码常用于模拟量的转换中，当模拟量发生微小变化而可能引起数字量发生变化时，格雷码仅改变一位，这样与其他码同时改变两位或多位的情况相比更为可靠，可减少出错的可能性。可用如图 1.2 所示的四变量卡诺图帮助记忆格雷码的编码方式。

1.4　基本逻辑运算

利用二值数字逻辑中的 1 和 0 不仅可以表示二进制数，还可表示许多对立的逻辑状态。在分析和设计数字电路时，所使用的数学工具是英国数学家乔治·布尔(1849 年建立布尔代数)提出的描述逻辑关系的数学方法——逻辑代数。逻辑代数是按一定的逻辑规律进行运算的代数，虽然它和普通代数一样也是用字母表示变量，但两种代数中变量的含义是完全不同的，它们之间有着本质的区别，逻辑代数中的变量(逻辑变量)只有两个值，即 0 和 1，而没有中间值。0 和 1 并不表示数量的大小，而表示对立的逻辑状态。

1.4.1　基本逻辑运算

在逻辑代数中，只有与、或、非 3 种基本逻辑运算。运算是一种函数关系，它可以用语句描述，亦可用逻辑表达式描述，还可用表格或图形来描述。描述逻辑关系的表格为真值表，用规定的图形符号来表示逻辑运算称为逻辑符号。下面分别讨论这 3 种基本的逻辑运算。

1. 与逻辑运算

只有当决定一件事情的条件全部具备之后，这件事情才会发生，把这种因果关系称为与逻辑。

实际生活中与逻辑的问题很多，举例如下。

(1) 为了安全，保险柜上安装了指纹识别系统，当甲、乙两人同时触摸保险柜的铁门时，铁门将自动打开。铁门被打开与甲、乙两人之间就构成了与逻辑问题。做表 1-3(a)描述如下，当然也可以用字母 A、B、F 分别表示条件甲、乙两人和结果“保险柜”，又可以做表 1-3(b)。

(2) 如图 1.3 所示，灯泡 F 亮、暗和开关甲 A、乙 B 构成与逻辑问题。表 1-3 同样反映了灯泡 F 和开关甲 A、乙 B 遵从的逻辑关系。

由表可知，逻辑与的基本运算规则为

$$0\cdot 0=0,0\cdot 1=0,1\cdot 0=0,1\cdot 1=1$$

$$0\cdot A=0,1\cdot A=A,A\cdot A=A$$

即：有零必为零，全 1 才为 1。

表 1-3　逻辑真值表

(a)			(b)		
甲	乙	保险柜	A	B	F
假	假	闭	0	0	0
假	真	闭	0	1	0
真	假	闭	1	0	0
真	真	开	1	1	1

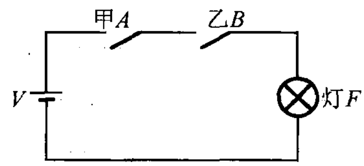
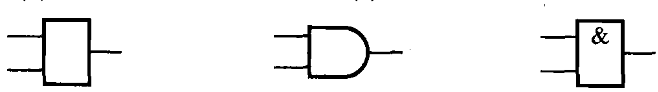


图 1.3　逻辑电路示意图

用逻辑表达式(也叫逻辑函数式)表示为：$F = A \cdot B$ 或者 $F = AB$ 。

实现与逻辑运算的电路称为与门，其逻辑符号如图 1.4 所示，其中图 1.4(a)是我国常用的传统符号，图 1.4(b)为国外流行符号，图 1.4(c)为国家标准符号。



(a)　传统符号　　(b)　国外流行符号　　(c) 国家标准符号

图 1.4　与逻辑符号

这里，关于与逻辑的问题共采用了表格、运算规则、表达式、逻辑符号 4 种方法进行了描述。实际上，其他的逻辑问题也可以采用上述 4 种方法进行描述。

2. 或逻辑运算

当决定一件事情的几个条件中，只要有一个或一个以上条件具备，这件事情就会发生，这种因果关系称为或逻辑。

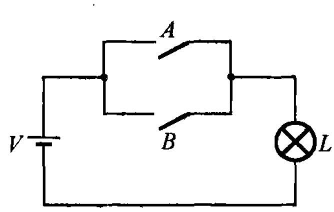
图 1.5(a)是实际生活中的或逻辑问题。或逻辑运算的关系表如图 1.5(b)所示，逻辑真值表如图 1.5(c)所示。若用逻辑表达式来描述，则可写为

$$L=A+B$$

其运算规则是：0+0=0，0+1=1，1+0=1，1+1=1，0+A=1，A+A=A，1+A=1

因此，或逻辑运算的规则可以表述为：输入有 1，输出为 1；输入全 0，输出为 0。

在数字电路中能实现或逻辑运算的电路称为或门，其逻辑符号如图 1.5(d)所示。或逻辑运算也可以推广到多变量：$L = A + B + C + \cdots$



(a) 电路图

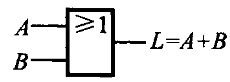
开关A　开关B	灯L
不闭合　不闭合	不亮
不闭合　闭合	亮
闭合　不闭合	亮
闭合　闭合	亮

(b) 逻辑关系表

图 1.5　或逻辑运算

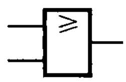
A	B	$L=A+B$
0	0	0
0	1	1
1	0	1
1	1	1

(c) 逻辑真值表

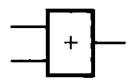


(d) 逻辑符号

图 1.5　或逻辑运算(续)



(a) 国家标准符号



(b) 常用符号

(c) 国际常用符号

图 1.6　或逻辑符号

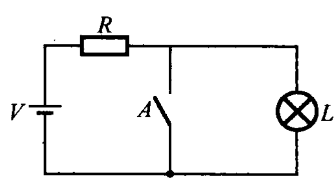
3. 非逻辑运算

某事情发生与否，仅取决于一个条件，即是对该条件的否定。条件具备时事情不发生，条件不具备时事情才发生。

例如图 1.7(a)所示的电路，当开关 A 闭合时，灯不亮；而当 A 不闭合时，灯亮。其逻辑关系表如图 1.7(b)所示，逻辑真值表如图 1.7(c)所示。若用逻辑表达式来描述，则可写为:$L=\overline{A}$。

非逻辑运算的规则为：$\overline{0}=1$；$\overline{1}=0$。

在数字电路中实现非逻辑运算的电路称为非门，其逻辑符号如图 1.7(d)所示。



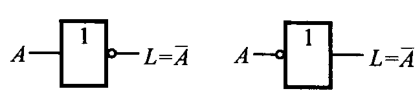
(a) 电路图

开关 A	灯 L
不闭合	亮
闭合	不亮

(b) 逻辑关系表

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

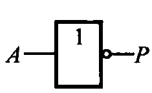
(c) 逻辑真值表



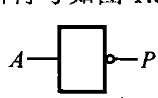
(d) 逻辑符号

图 1.7　非逻辑运算

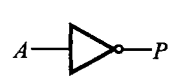
非逻辑的国家标准符号和常用逻辑符号如图 1.8 所示。



(a) 国家标准符号



(b) 常用符号



(c) 国际常用符号

图 1.8　非逻辑符号

1.4.2 其他常用逻辑运算

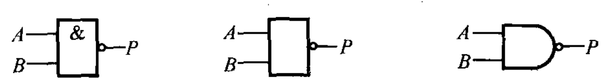
实际的逻辑问题往往比与、或、非基本逻辑复杂，不过它们都可以用与、或、非组合成的复合逻辑来实现。在实际应用中为了减少逻辑门的数目，使数字电路的设计更方便，还常常使用其他几种常用逻辑运算。生产厂家为此也设计出了常用的复合逻辑，有与非、或非、与或非、异或和同或等。

1. 与非逻辑

与非逻辑的真值表见表 1-4。由表 1-4 可见，与非逻辑是将 A、B 进行与运算，然后将其结果求反得到，因此可以把与非运算看做是与运算和非运算的组合，并用逻辑符号上的小圆圈代表非运算。逻辑符号如图 1.9 所示。与非运算的表达式为 $P=\overline{AB}$。

表 1-4　与非逻辑真值表

A	B	P
0	0	1
0	1	1
1	0	1
1	1	0



(a) 国家标准符号　(b) 常用符号　(c) 国际常用符号

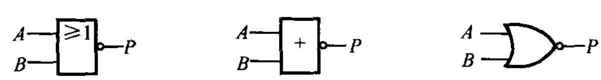
图 1.9　与非逻辑符号

2. 或非逻辑

或非逻辑的真值表见表 1-5，逻辑符号如图 1.10 所示。由表 1-5 可见，或非逻辑是将 A、B 进行或运算，然后将其结果求反得到，因此可以把或非运算看做是或运算和非运算的组合。或非运算的表达式为 $P=\overline{A+B}$。

表 1-5　或非逻辑真值表

A	B	P
0	0	1
0	1	0
1	0	0
1	1	0



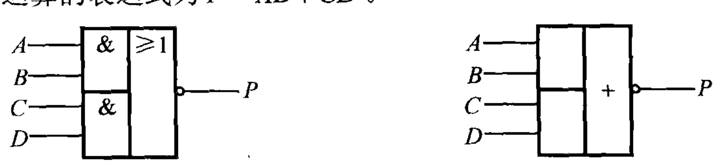
(a) 国家标准符号　(b) 常用符号　(c) 国际常用符号

图 1.10　或非逻辑逻辑符号

3. 与或非逻辑

与或非逻辑的逻辑符号如图 1.11 所示，真值表略。在与或非逻辑中，A、B 之间以及 C、D 之间都是与关系，然后把它们与的结果进行或运算，最后再进行非运算。

与或非逻辑运算的表达式为 $P=\overline{AB+CD}$。



(a) 国家标准符号　(b) 常用符号

图 1.11　与或非逻辑符号

4. 异或逻辑和同或逻辑

异或是一种二变量逻辑运算，当两个变量取值相同时，逻辑函数值为 0；当两个变量取值不同时，逻辑函数值为 1。

异或逻辑的真值表见表 1-6(a)，逻辑符号如图 1.12(a)所示。异或逻辑是这样一种逻辑关系，当 A、B 不同时，输出 P=1；而当 A、B 相同时，输出 P=0，“$\oplus$”是异或运算符号，异或逻辑也是与、或、非逻辑的组合，其逻辑表达式为

$$P = A\bar{B} + \bar{A}B = A \oplus B$$

同或逻辑的真值表见表 1-6(b)，逻辑符号如图 1.12(b)所示。同或逻辑和异或逻辑相反，当 A、B 相同时，输出 P=1；而当 A、B 不同时，输出 P=0。“$\odot$”是同或运算符号，同或逻辑的逻辑表达式为

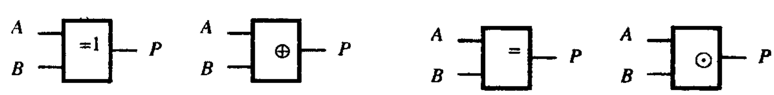
$$P = \bar{A}\bar{B} + AB = A \odot B$$

表 1-6(a)　异或逻辑真值表

A	B	P
0	0	0
0	1	1
1	0	1
1	1	0

表 1-6(b)　同或逻辑真值表

A	B	P
0	0	1
0	1	0
1	0	0
1	1	1



(a) 异或　　(b) 同或

图 1.12　异或和同或逻辑符号

1.5 逻 辑 代 数

逻辑代数和普通代数一样，有一套完整的运算规则，包括公理、定理和定律，用它们对逻辑函数式进行处理，可以完成对电路的化简、变换、分析与设计。

1.5.1　逻辑代数的运算公式

逻辑代数的运算公式包括 9 个定律，见表 1-7，其中有的定律与普通代数相同，有的定律与普通代数不同，使用时切勿混淆。

表 1-7　逻辑代数的基本公式

名　称	公　式　1	公　式　2
0-1 律	$A \cdot 1 = A$ $A \cdot 0 = 0$	$A + 0 = A$ $A + 1 = 1$