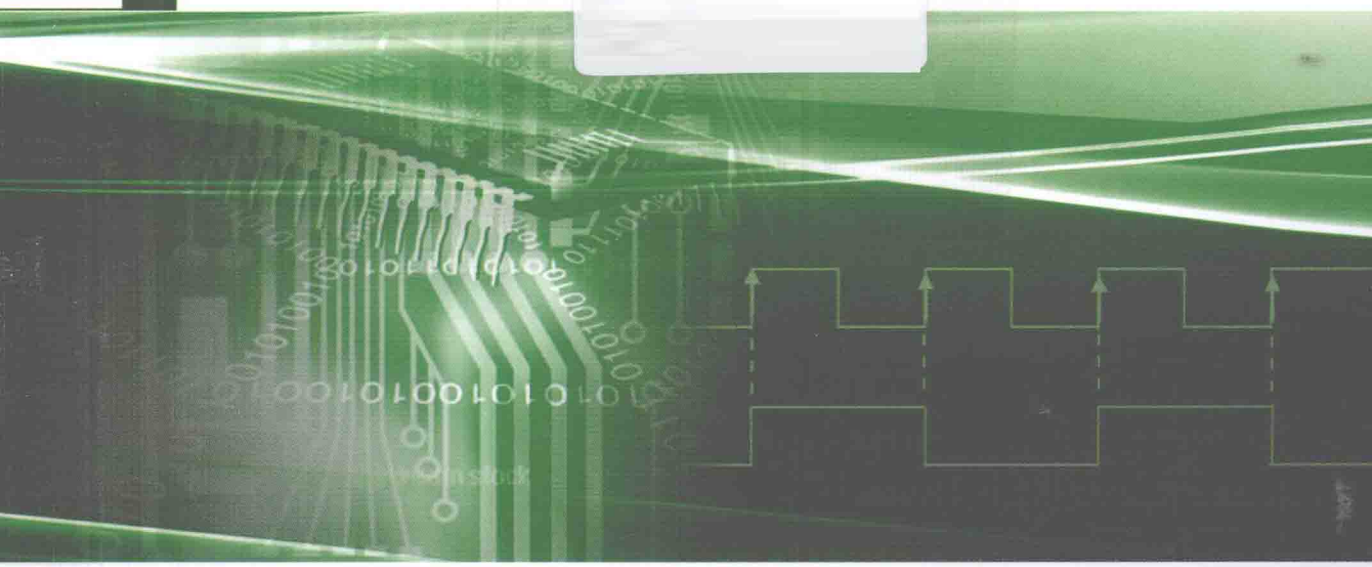
高等职业技术教育“十二五”规划教材
高等职业技术教育校企合作教材

数字电路技术基础

SHUZI DIANLU JISHU JICHU

主　编　鲁　军
副主编　马元凯　熊　英
　　况　君　刘　洋
主　审　梁卫华



西南交通大学出版社
Http://press.swjtu.edu.cn

高等职业技术教育“十二五”规划教材

数字电路技术基础

主　编　鲁　军
副主编　马元凯　熊　英
　　况　君　刘　洋
主　审　梁卫华

西南交通大学出版社
·成　都·

内容简介

本书介绍了数制与代码、常用逻辑门的符号及功能等数字电路技术的基础知识，各种常用集成门电路的基本特性及实际应用电路，同时给出了典型实训内容和项目设计。主要内容包括逻辑代数基础、门电路、组合逻辑电路、集成触发器、时序逻辑电路、半导体存储器与可编程逻辑器件、数-模与模-数转换器等内容，重点介绍了各种门电路的应用实例，以典型应用电路作为项目设计和实训内容。

本书可作为高等职业院校计算机、电子、通信等工科类专业的教材，也可作为大中专院校师生的专业参考书。

图书在版编目（CIP）数据

数字电路技术基础 / 鲁军主编. —成都：西南交通大学出版社，2014.2
高等职业技术教育“十二五”规划教材
ISBN 978-7-5643-2911-2

Ⅰ. ①数… Ⅱ. ①鲁… Ⅲ. ①数字电路－高等职业教育－教材 Ⅳ. ①TN79

中国版本图书馆 CIP 数据核字（2014）第 027473 号

高等职业技术教育“十二五”规划教材
数字电路技术基础
主编　鲁　军
*
责任编辑　李芳芳
助理编辑　宋彦博
特邀编辑　黄庆斌
封面设计　墨创文化
西南交通大学出版社出版发行
（四川省成都市金牛区交大路 146 号　邮政编码：610031　发行部电话：028－87600564）
http://press.swjtu.edu.cn
四川森林印务有限责任公司印刷
*
成品尺寸：185 mm × 260 mm　　印张：10.75
字数：269 千字
2014 年 2 月第 1 版　　2014 年 2 月第 1 次印刷
ISBN 978-7-5643-2911-2
定价：23.00 元

前　言

“数字电路技术基础”是通信、电子及计算机等专业必修的一门基础课，也是从电学基础知识向专业知识过渡的重要课程之一。本书根据高等职业教育电子信息工程、通信技术及相近专业的教学要求编写而成。在编写过程中，主要考虑到要符合高职高专教育的特点，以应用为目的，以必须、够用为尺度，以基本概念、基本电路、基本方法为重点，因此在教材结构及内容安排上更加注重了集成电路的应用，将“教”“学”“做”“评”融为一体，以培养学生的实践技能。

全书包括 7 章、两个综合实训及附录。第 1 章为逻辑代数基础，主要介绍数制与代码、常用逻辑门的符号及功能、逻辑函数的描述方法及化简；第 2 章为门电路，主要介绍 TTL、CMOS 门电路及其应用；第 3 章为组合逻辑电路，主要介绍组合逻辑电路的分析和设计、常见组合逻辑模块的功能和应用；第 4 章为集成触发器，主要介绍 *RS*、*JK*、*D*、*T* 等常用触发器及其应用；第 5 章为时序逻辑电路，主要介绍时序逻辑电路的分析和设计、常见时序逻辑模块的功能和应用；第 6 章为半导体存储器与可编程逻辑器件，主要介绍半导体存储器的特点及可编程逻辑器件 PLD 的发展概况；第 7 章为数-模和模-数转换器，主要介绍集成数-模转换器和模-数转换器的基本概念、基本原理和典型电路。

本书除第 1 章讲逻辑代数基础外，其他各章都把相关应用电路部分内容单独成节，这样既有利于加强学生学习针对性、激发学习兴趣，也便于授课老师根据具体情况进行内容拓展。此外，每章都配有思考和练习题，安排有技能实训和综合实训，可以提高学生对数字电路的实际操作能力和综合应用能力。

为了全面落实教育部的“十二五”教育规划纲要，以服务为宗旨、以就业为导向，遵循技能型人才成长规律，适应经济发展方式、产业发展水平、岗位对技能型人才的要求，本书坚持行业指导、企业参与、校企合作的教材开发机制。因此，特别邀请了重庆普天普科通信技术有限公司总经理马元凯参与教材的开发和编写工作，切实反映了职业岗位能力对专业基础课程的要求，对接企业用人需求。

本书由鲁军副教授担任主编，并负责全书的统稿；由马元凯、熊英、况君、刘洋担任副主编；由重庆电讯职业学院通信技术系梁卫华主任担任主审，并为本书的编写提出了很多指导性意见。

本书在编写过程中参考了众多专家学者的研究成果，在此，向所有作者表示深深的谢意。

由于编者水平有限，加之时间仓促，书中难免存在不妥之处，诚望读者批评指正。

编　者

2013 年 4 月于重庆

目　录

逻辑代数基础

数字电路主要研究数字信号的产生、存储、变换及运算等，其分析及设计方法是电子工程技术人员所必备的基础知识。二进制和逻辑代数是目前数字逻辑电路中进行算术运算及逻辑运算的主要数学工具。

本章主要讨论数字电路的计数体制、逻辑代数、逻辑函数及其化简。

1.1 数制与代码

1.1.1 数　制

数制就是计数的制度，也称为计数方式或位置计数制；代码则是一种符号，指特殊的数码。

人们在生活中经常使用的是十进制计数方式；而在数字电路中，由于经常使用电路的通、断或电平的高、低来表示“1”和“0”，因此采用二进制计数方式更加方便和实用。此外，为了读写和操作方便，在数字电路中还经常使用八进制和十六进制计数方式。不同的数制之间可以相互转换，为了区别不同的计数方式，通常在数的左右两边加括号，并在右括号的下标处指明进制。

1. 十进制

在十进制数中，每个数位可用的数码为 0～9，共十个数码。其计数规则是“逢十进一”。通常我们把每位可用数码的个数称为该进制数的“基数”。如十进制数的“基数”是“10”。十进制数用下标“10”表示，也可用英文大写字母“D”表示。

另外，在这种位置计数制中，同一个数码在不同的数位上所表示的数值是不同的。例如，十进制数 $(911.1)_{10}$，小数点前第二位的“1”代表 10^1，第一位的“1”代表 10^0，而小数点后第一位的“1”代表 10^{-1}。通常我们把某位数码为 1 时所代表的十进制数值称为该位的“权”。十进制整数个位、十位、百位……的权分别为 10^0、10^1、10^2…。一般地说，第 i 位的权为基

数的 i 次幂，其中 $i=\cdots2、1、0、-1、-2\cdots$，也称 i 为各个数位的序号。

运用“权”的概念，我们可以将任意一个十进制数表示成每位数码乘以该位的“权”值，然后相加的形式。例如：

$$(368.25)_{10}=3\times10^{2}+6\times10^{1}+8\times10^{0}+2\times10^{-1}+5\times10^{-2}$$

十进制数人们最熟悉，但在数字电路中实现起来比较困难。

2. 二进制

在二进制数中，每个数位可使用的数码为 0、1，共 2 个数码，故其基数为 2。其计数规则是“逢二进一”。二进制数的下标用“2”表示，也可用英文大写字母“B”表示。各位的权值为 2^i，其中 i 是各个数位的序号。我们也可运用“权”的概念，将一个二进制数表示成每位数码乘以该位的“权”值，然后相加的形式。例如：

$$(1011.01)_2=1\times2^{3}+0\times2^{2}+1\times2^{1}+1\times2^{0}+0\times2^{-1}+1\times2^{-2}=(11.25)_{10}$$

可见，二进制数变为十进制数只需要按权展开相加即可。此外二进制数由于只需两个状态，机器实现容易，但有时二进制数位数太多，不便书写和记忆。

由于二进制可以很方便地用具有两个稳定状态的电子器件（如晶体管）的饱和及截止来实现（其中一个状态表示“0”，另一个状态表示“1”），而且二进制运算简单，因此在数字电路及计算机中得到了广泛应用。

3. 八进制

在八进制数中，每个数位上可使用的数码为 0~7，共 8 个，故其基数为 8。其计数规则为“逢八进一”。八进制数的下标用“8”表示，也可用英文大写字母“O”表示。各位的权值为 8^i，其中 i 是各个数位的序号。例如：

$$(752.34)_8=7\times8^{2}+5\times8^{1}+2\times8^{0}+3\times8^{-1}+4\times8^{-2}=(490.4375)_{10}$$

因为 $2^3=8$，即 3 位二进制数可用一位八进制数表示，所以八进制数便于表示较长的二进制数。

4. 十六进制

在十六进制中，每个数位上规定使用的数码符号为 0~9 和 A~F，共 16 个，故其进位基数为 16。其计数规则是“逢十六进一”。十六进制数的下标用“16”表示，也可用英文大写字母“H”表示。各位的权值为 16^i，其中 i 是各个数位的序号。例如：

$$(2B.4)_{16}=2\times16^{1}+11\times16^{0}+4\times16^{-1}=(43.25)_{10}$$

因为 $2^4=16$，4 位二进制数可用一位十六进制数表示，所以可用十六进制数表示较长的二进制数。在数字系统中，二进制主要用于机器内部的数据处理，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。

1.1.2 数制的转换

由于二进制数不便于读写，而人们又习惯使用十进制数，因此二进制数往往只在数字电路、计算机内部使用，这就需要在机器的输入端及输出端进行不同数制的转换。

1. 二进制数转换为十进制数

转换原则：按权展开相加。前面在讲数制时，已经举例说明了此方法，这里不再赘述。

2. 十进制数转换为二进制数

把十进制数转换成二进制数常采用基数乘除法。下面分别就整数部分和小数部分的转换加以说明。

（1）整数转换。整数转换采用基数连除法。把十进制整数 N 转换成二进制数的步骤如下：

① 将数 N 除以 2，记下所得的商和余数。

② 将上一步所得的商再除以 2，记下所得商和余数。

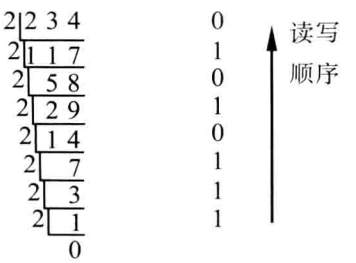
③ 重复做上一步，直到商为 0。

④ 将各步骤中求得的余数按照与运算过程相反的顺序从下向上排列起来，即为所求二进制数的整数部分。

【例 1-1】 $(234)_{10}=(\ ?\ )_2$

解：

余数



所以，$(234)_{10}=(11101010)_2$

（2）纯小数转换。纯小数转换采用基数连乘法。把十进制的纯小数 M 转换成二进制数的步骤如下：

① 将 M 乘以 2，记下整数部分。

② 将上一步乘积中的小数部分再乘以 2，记下整数部分。

③ 重复做上一步，直到小数部分为 0 或者满足精度要求为止。

④ 将各步骤中求得的整数按照与运算过程相同的顺序从上向下排列起来，即为所求二进制数的小数部分。

【例 1-2】$(0.6875)_{10}=(?)_2$

解： 整数

$$
\begin{array}{r}
0.6875 \\
\times\quad 2 \\
\hline
1.3750 \qquad 1 \\
0.3750 \\
\times\quad 2 \\
\hline
0.7500 \qquad 0 \\
\times\quad 2 \\
\hline
1.5000 \qquad 1 \\
0.5000 \\
\times\quad 2 \\
\hline
1.0000 \qquad 1
\end{array}
$$

读写顺序 ↓

所以，$(0.6875)_{10}=(0.1011)_2$

注意： 小数转换不一定能算尽，只能算到一定精度的位数为止，故要产生一些误差。

如果一个十进制数既有整数部分又有小数部分，可将整数部分和小数部分分别按要求进行等值转换，然后合并就可得到结果，这种方法称作基数乘除法。

【例 1-3】$(11.375)_{10}=(?)_2$

解： 余数

$$
\begin{array}{r|l}
2 & 11 \\
\hline
2 & 5 \quad \cdots 1 \\
2 & 2 \quad \cdots 1 \\
2 & 1 \quad \cdots 0 \\
 & 0 \quad \cdots 1
\end{array}
\quad \uparrow
$$

整数

$$
\begin{array}{r}
0.375 \\
\times\quad 2 \\
\hline
0.750 \qquad 0 \\
\times\quad 2 \\
\hline
1.500 \qquad 1 \\
0.500 \\
\times\quad 2 \\
\hline
1.000 \qquad 1
\end{array}
\quad \downarrow
$$

所以，$(11.375)_{10}=(1011.011)_2$

3. 二进制数转换成八进制数或十六进制数

二进制数转换成八进制数（或十六进制数）时，其整数部分和小数部分可以同时进行转换。其方法是：以二进制数的小数点为起点，分别向左、向右，每三位（或四位）分一组。对于小数部分，最低位一组不足三位（或四位）时，必须在有效位右边补 0，使其足位；对于整数部分，最高位一组不足位时，可在有效位的左边补 0，也可不补。然后，把每一组二进制数转换成八进制（或十六进制）数，并保持原排序。

【例 1-4】 $(1011011111.10011)_2=(?)_8=(?)_{16}$

解： $(\underline{001}\ \underline{011}\ \underline{011}\ \underline{111}.\underline{100}\ \underline{110})_2=(1337.46)_8$

$(\underline{0010}\ \underline{1101}\ \underline{1111}.\underline{1001}\ \underline{1000})_2=(2DF.98)_{16}$

4. 八进制数或十六进制数转换成二进制数

八进制（或十六进制）数转换成二进制数时，只要把八进制（或十六进制）数的每一位数码分别转换成三位（或四位）的二进制数，并保持原有排序即可。整数最高位一组左边的 0 和小数最低位一组右边的 0，可以省略。

【例 1-5】 $(36.24)_8=(?)_2$

解： $(36.24)_8$ = （011 110.010 100）$_2$ = （11110.0101）$_2$

3 6.2 4

【例 1-6】（3DB.46）$_{16}$ = （?）$_2$

解：（3DB.46）$_{16}$ = （0011 1101 1011. 0100 0110）$_2$ = （1111011011.0100011）$_2$

3 D B . 4 6

由上可见，对于几种常用进制数的转换，非十进制数转换成十进制数可采用按权展开法；十进制数转换成二进制数时可采用基数乘除法，即整数采用连续“除 2 取余”，小数转换采用连续“乘 2 取整”；二进制数与八进制数、十六进制数转换时可采用分组转换的方法。

5. 二进制移位

二进制数移位可分为左移和右移。左移时，若低位移进位为 0，相当于该二进制数乘 2；右移时，若高位移进位为 0，移出位作废，相当于该二进制数除以 2。

例如，1010B 左移后变为 10100B，10100B = 1010B × 2；1010B 右移后变为 0101B，0101B = 1010B/2。

1.1.3 代 码

不同的数码不仅可以表示数量的不同大小，而且还能表示不同的事物。前面我们讨论的二进制数就是用多位“0-1”序列表示数的大小，而“0-1”序列也可作为“二元符号”来表示十进制数码、英文字母或其他符号，此时的 0、1 已没有数量大小的含义，只是表示不同事物的符号而已。如键盘上的“Enter”键在计算机中用七位“0-1”序列 0001101 来表示。通常我们把这种代表一定意义的二元符号组称为代码。下面介绍一下最常用的几种代码。

1. 自然二进制码

自然二进制码的形式与二进制数相同，但它已经没有数的大小概念，如 4 位自然二进制码只是作为代表“0 ~ 15”的 16 个 4 位二进制符号而已。

2. 二-十进制码（BCD 码）

二-十进制码是用二进制码元来表示十进制数符“0 ~ 9”的代码，简称 BCD（Binary Coded Decimal）码。用二进制码元来表示“0 ~ 9”这 10 个数符，必须用 4 位二进制码元来表示，而 4 位二进制码元共有 16 种组合，从中取出 10 种组合来表示“0 ~ 9”的编码方案有很多种，几种常用的 BCD 码如表 1.1 所示。若某种代码的每一位都有固定的“权值”，则称这种代码为有权码；否则叫无权码。最常用的有权码和无权码分别是 8421BCD 码和余 3BCD 码。8421BCD 码各位的权值分别为 8、4、2、1；余 3 码是 8421BCD 码的每个码组加 0011 形成的，余 3 码各位无固定权值，故属于无权码。大家可以用二-十进制码（BCD 码）来表示十进制数。

表 1.1 几种常用的 BCD 码

十进制数	8421 码	5421 码	2421 码	余 3 码	BCD Gray 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	1000	1011	1000	0111
6	0110	1001	1100	1001	0101
7	0111	1010	1101	1010	0100
8	1000	1011	1110	1011	1100
9	1001	1100	1111	1100	1000

【例 1-7】$(246.1)_{10}$ = （？）$_{8421BCD}$

解： $(246.1)_{10}$ = （0100 0100 0110.0001）$_{8421BCD}$

在这里，要注意每一位十进制数必须与 4 位 BCD 码对应，即便是整数的最高位或小数的最低位为“0”也不能省略，这与二进制数变十六进制数时是不同的。同时，如果用 BCD 码来表示非十进制数，必须先将非十进制数转换成十进制数，再求其对应的 BCD 码。

【例 1-8】$(10000100.1100)_{余3BCD}$ = （？）$_{8421BCD}$

解： $(01001000.1011)_{余3BCD} = (51.9)_{10} = (01010001.1001)_{8421BCD}$

两种 BCD 码相互转换时，通常将一种 BCD 码先转换成十进制数。

3. ASCII 码

除了上面介绍的代码以外，还有另一大类的信息传输代码，最典型的就是美国国家信息交换标准代码 ASCII 码。它用七位二进制码来表示某些数字、英文字母、数学符号和某些图形。

例如，数字“0~9”的 ASCII 代码是 30H~39H；大写英文字母 A ~ Z 的 ASCII 代码是 41H~5AH；小写英文字母 a ~ z 的 ASCII 代码是 61H~7AH；“？”的 ASCII 码是 3FH；“%”的 ASCII 码是 25H，等等。

1.2 逻辑代数

逻辑代数又称为布尔代数或开关代数，它是研究开关理论和分析、设计数字逻辑电路的数学基础。

1.2.1 逻辑变量与逻辑函数

逻辑代数中的变量称为逻辑变量，常用英文大写字母 A、B、C 等表示。逻辑变量只有“0”与“1”两种可能的取值，它们没有“大小”的含义，也无“数量”的概念，分别代表两种对立的状态，比如开关器件的通断、信号的有无、电平的高低以及事件的真假等。当两个二进制数码表示不同的逻辑状态时，它们之间可以按照指定的某种因果关系进行推理运算，我们将这种运算称为逻辑运算。逻辑运算是逻辑思维和逻辑推理的数学描述。

在数字电路中，我们约定：用“1”表示高电平，“0”表示低电平，叫做“正逻辑”；而用“0”表示高电平，“1”表示低电平，叫做“负逻辑”。在以后的讨论中，如无特殊说明都是针对正逻辑而言。

1.2.2　基本逻辑关系

逻辑代数中有与（AND）、或（OR）、非（NOT）三种基本逻辑运算，它们是所有逻辑运算的基础。

1. 与逻辑运算（逻辑乘）

决定某一事件的所有条件同时成立，该事件才发生，这种因果关系叫“与”逻辑，也叫与运算或逻辑乘。

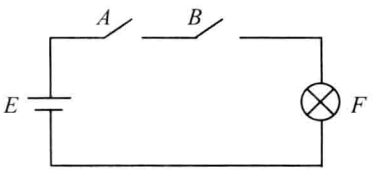


图 1.1　与逻辑电路示意图

表 1.2　与逻辑的真值表

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

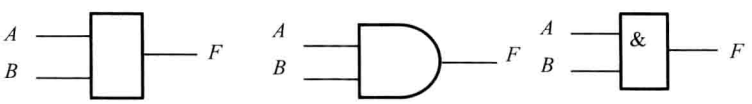
例如，如图 1.1 所示电路中的 A、B 为开关，用来控制灯 F 的亮与灭，如果把开关“闭合”状态记作“1”，开关“断开”状态记作“0”，而把灯“亮”状态记作“1”，灯“灭”状态记作“0”，那么灯的状态与开关的状态之间的关系有四种可能的情况出现，如表 1.2 所示。这种表称为真值表。所谓真值表，就是将输入变量的所有可能的取值组合所对应的输出变量的值一一列出来的表格。它是描述逻辑功能的一种重要形式。

“与”逻辑除了用表格形式表示外，还可以用表达式把 F 与 A、B 之间的关系表示为

$$F = A \cdot B$$

这种表示形式称为逻辑函数式。它也是描述逻辑功能的一种重要形式。此式中的“$A \cdot B$”读作“A 与 B”或“A 逻辑乘 B”；式中的“$\cdot$”是与运算符，通常可省略，即 $A \cdot B = AB$。

实现“与运算”的电路叫做与门，其逻辑符号如图 1.2 所示。



（a）常用传统符号　（b）国外流行符号　（c）国际标准符号

图 1.2　与门的逻辑符号

由表 1.2 可知，逻辑乘的基本运算规则为

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

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

2. 或逻辑运算（逻辑加）

决定某一事件的所有条件中，只要有一个成立，则该事件就发生，这种因果关系叫“或”逻辑，也叫或运算或逻辑加。

例如，如图 1.3 所示电路中的 A、B 为开关，用来控制灯 F 的亮与灭，若约定逻辑假设不变，那么灯的状态与开关的状态之间的关系有四种可能的情况出现，如表 1.3 所示。

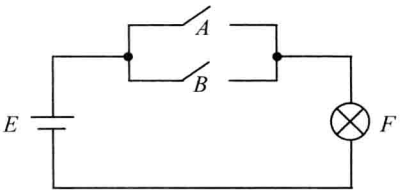


图 1.3　或逻辑电路示意图

表 1.3　或逻辑的真值表

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

“或”逻辑除了用表格形式表示外，还可以用表达式把 F 与 A、B 之间的关系表示为

$$F = A + B$$

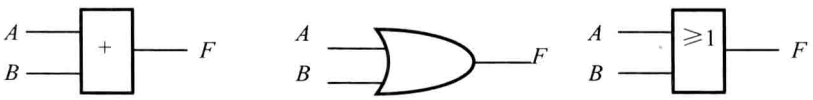
此式中的“$A + B$”读作“A 或 B”或“A 逻辑加 B”；式中的“+”是或运算符。

实现“或运算”的电路叫或门，其逻辑符号如图 1.4 所示。

由表 1.3 可知，逻辑加的基本运算规则为

$0 + 0 = 0$　　$0 + 1 = 1$　　$1 + 0 = 1$　　$1 + 1 = 1$

$0 + A = A$　　$1 + A = 1$　　$A + A = A$



（a）常用传统符号（b）国外流行符号（c）国际标准符号

图 1.4　或门的逻辑符号

3. 非逻辑运算（逻辑反）

若前提条件为“真”，则结论为“假”；若前提条件为“假”，则结论为“真”。即结论是对前提条件的否定，这种因果关系叫做“非”逻辑，也叫非运算或逻辑反。

例如，如图 1.5 所示电路中的 A 为开关，用来控制灯 F 的亮与灭，若约定逻辑假设不变，那么灯的状态与开关的状态之间的关系有两种可能的情况出现，如表 1.4 所示。

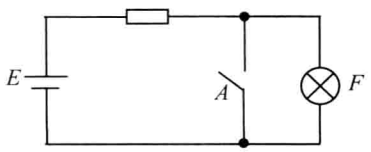


图 1.5　非逻辑电路示意图

表 1.4　非逻辑的真值表

A	F
0	1
1	0

“非”逻辑除了用表格形式表示外，也可以用逻辑函数式的形式把 F 与 A 之间的关系表示为

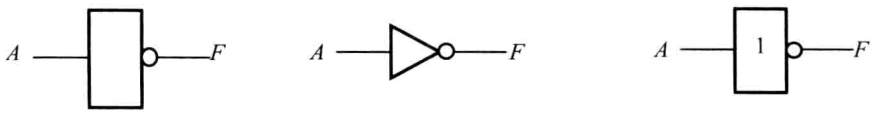
$$F=\overline{A}$$

此式中的“$\overline{A}$”读作“A 非”；式中的“－”表示非运算符。

由表 1.4 可知，非逻辑的基本运算规则为

$$\overline{0}=1 \qquad \overline{1}=0$$

实现“非运算”的电路叫非门，其逻辑符号如图 1.6 所示。



（a）常用传统符号　　（b）国外流行符号　　（c）国际标准符号

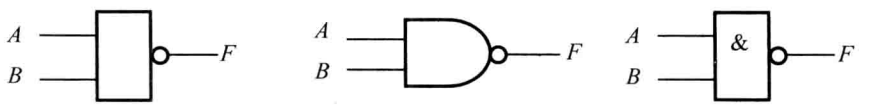
图 1.6　非门的逻辑符号

4.“与非”逻辑运算

“与非”逻辑运算是“与”逻辑和“非”逻辑的组合。先“与”再“非”，表达式为

$$F=\overline{A\cdot B}$$

实现“与非”逻辑运算的电路叫“与非门”。其逻辑符号如图 1.7 所示。



（a）常用传统符号　　（b）国外流行符号　　（c）国际标准符号

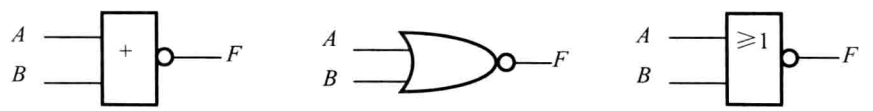
图 1.7　与非门的逻辑符号

5.“或非”逻辑运算

“或非”逻辑运算是“或”逻辑和“非”逻辑的组合。先“或”后“非”，其表达式为

$$F=\overline{A+B}$$

实现“或非”逻辑运算的电路叫“或非门”。其逻辑符号如图 1.8 所示。



（a）常用传统符号　　（b）国外流行符号　　（c）国际标准符号

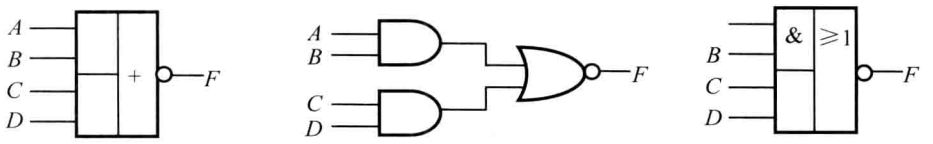
图 1.8　或非门的逻辑符号

6. “与或非”逻辑运算

“与或非”逻辑运算是“与”、“或”、“非”三种基本逻辑的组合。先“与”再“或”最后“非”。其表达式为

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

实现“与或非”逻辑运算的电路叫做“与或非门”。其逻辑符号如图 1.9 所示。



（a）常用传统符号　　（b）国外流行符号　　（c）国际标准符号

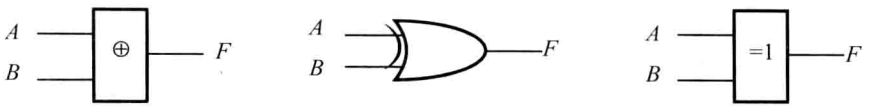
图 1.9　与或非门的逻辑符号

7. “异或”逻辑运算和“同或”逻辑运算

若两个输入变量 A、B 的取值相异，则输出变量 F 为 1；若 A、B 的取值相同，则 F 为 0。这种逻辑关系叫“异或”逻辑，其逻辑表达式为

$$F=A\oplus B=\overline{A}B+A\overline{B}$$

式中的“$A\oplus B$”读作“A 异或 B”。实现“异或”运算的电路叫“异或门”。其逻辑符号如图 1.10（a）、（b）、（c）所示。



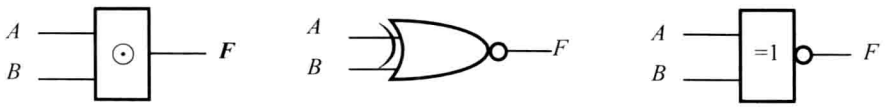
（a）常用传统符号　　（b）国外流行符号　　（c）国际标准符号

图 1.10　异或门的逻辑符号

若两个输入变量 A、B 的取值相同，则输出变量 F 为 1；若 A、B 的取值相异，则 F 为 0。这种逻辑关系叫“同或”逻辑，其逻辑表达式为

$$F=A\odot B=\overline{A}\overline{B}+AB$$

式中的“$A\odot B$”读作“A 同或 B”。实现“同或”运算的电路叫“同或门”。其逻辑符号如图 1.11 所示。



（a）常用传统符号　　（b）国外流行符号　　（c）国际标准符号

图 1.11　同或门的逻辑符号

两变量的“异或”及“同或”逻辑的真值表如表 1.5 所示。

对于两个逻辑函数 F_1 和 F_2，若输入变量的所有取值组合，函数 F_1 和 F_2 的取值总是相反，则称 F_1 和 F_2 互为反函数。记作

$$F_1=\overline{F_2} \text{ 或} F_2=\overline{F_1}$$

由表 1.5 可知，两变量的“异或逻辑”和“同或逻辑”互为反函数。即

$$A\odot B=\overline{A\oplus B}$$

由后面介绍的对偶定理可知，$A\oplus B$ 和 $A\odot B$ 还互为对偶式。

表 1.5 异或逻辑和同或逻辑的真值表

A	B	$F_{异}$
0	0	0
0	1	1
1	0	1
1	1	0

A	B	$F_{同}$
0	0	1
0	1	0
1	0	0
1	1	1

在实际应用中，常利用异或门（或同或门）来控制信号的传送。如将信号接到 A 端，B 作为控制端，当 $B=0$ 时，$F=A$，信号可以直接传送；当 $B=1$ 时，$F=\overline{A}$，信号取反传送，相当于非门的功能。

1.2.3 基本公式、定理和常用规则

前面已经介绍了基本逻辑运算和常用逻辑运算。英国数学家布尔首先提出的布尔代数被广泛应用于解决开关电路和数字逻辑电路的分析与设计中，因此也将布尔代数称为开关代数或逻辑代数。下面就介绍逻辑代数中常用的基本公式和定理。

1. 基本公式

逻辑代数的基本公式是由若干常量、变量和与、或、非三种基本运算组成的逻辑等式，也称为布尔恒等式。逻辑代数的运算服从如表 1.6 所示的基本公式。

表 1.6 逻辑代数的基本公式

公式名称	公式	
1. 0—1 律	$A\cdot 0=0$	$A+1=1$
2. 自等律	$A\cdot 1=A$	$A+0=A$
3. 互补律	$A\cdot\overline{A}=0$	$A+\overline{A}=1$
4. 重叠律	$A\cdot A=A$	$A+A=A$
5. 交换律	$A\cdot B=B\cdot A$	$A+B=B+A$