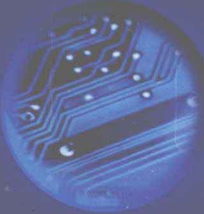
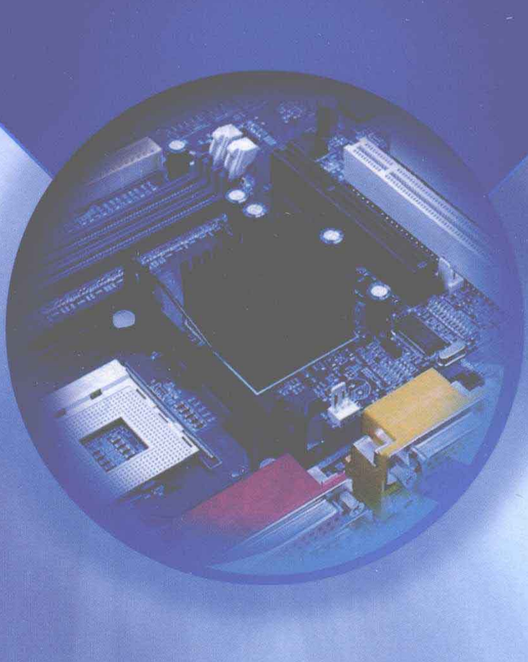
高职高专“十一五”规划教材

SHUZI DIANZI JISHU JICHU



数字电子技术基础

赵军 主编



化学工业出版社

高职高专"十一五"规划教材

数字电子技术基础

赵　军　主　编
柳明丽　高建新　副主编
张英囡　于宝琦　李青山　参　编

化学工业出版社
·北京·

本书是面向高职高专院校电子信息、通信、计算机、电气工程与自动化等专业的需要而编写的专业基础课教材。在内容编排上，教材以贴近工程实际中所需的数字电子技术基础知识和基本技能为主线，讲授了逻辑函数基础、逻辑门电路、组合逻辑电路、时序逻辑电路、存储器和可编程逻辑器件、脉冲的产生与整形及数/模和模/数转换等七方面的内容。全书体现了高职高专教育的特点，并且尽量做到以“必需、够用”为原则，以指导实践应用为目的，强调结论以及结论在实际中的应用，不强调公式的推导和理论的论证。在大多数章节中安排了实践练习内容，列举了与各章相对应的实用电路，使学生能在更接近实际的氛围中进行学习，培养学生分析问题、解决问题的能力和工程实践能力。

为了使读者更好地掌握和理解课程内容，书中配有较为丰富且贴近实际的例题、实践练习、思考题和习题，并在本书的最后附有部分习题的参考答案、数字电子技术基础常用中英文名词对照等内容。

本书简明扼要，深入浅出，便于自学，既可以作为高职高专院校和应用型本科院校电类相关专业的教材，也可以作为从事电子技术的工程技术人员的参考书。

图书在版编目（CIP）数据

数字电子技术基础/赵军主编.—北京：化学工业出版社，2009.1

高职高专“十一五”规划教材

ISBN 978-7-122-03937-8

Ⅰ.数…　Ⅱ.赵…　Ⅲ.数字电路-电子技术-高等学校：技术学院-教材　Ⅳ.TN79

中国版本图书馆 CIP 数据核字（2008）第 164374 号

责任编辑：王听讲　廉　静　　　文字编辑：王　洋

责任校对：边　涛　　　装帧设计：韩　飞

出版发行：化学工业出版社（北京市东城区青年湖南街 13 号　邮政编码 100011）

印　　装：北京白帆印务有限公司

787mm×1092mm　1/16　印张 13½　字数 331 千字　2009 年 3 月北京第 1 版第 1 次印刷

购书咨询：010-64518888（传真：010-64519686）　售后服务：010-64518899

网　　址：http://www.cip.com.cn

凡购买本书，如有缺损质量问题，本社销售中心负责调换。

定　　价：23.00 元

前　言

高等职业教育在我国高等教育中承担着重要角色，尤其工科的高等职业教育，更是担负着为我国工业、农业和国防的现代化建设培养应用型工程技术人才的重任。为了适应电子技术在工业生产中的广泛应用，各高等职业院校都开设了数字电子技术基础这门课程，并作为专业必修课引入教学。

本书是根据教育部最新制定的《高职高专教育基础课程教学基本要求》，并结合高等职业院校电子信息工程、通信、计算机、电气工程与自动化等专业的要求编写的专业基础课教材。它重视理论教学，更重视实践环节，主要任务是通过各个教学环节，运用各种教学手段和方法，使学生在数字电子技术方面获得知识、素质和技能，并为以后学习各专业知识和高一级的职业技能培训打下良好的基础。本书以培养应用型工程技术人才为目标，实用性强。

本书的建议学时为 80～100 学时。书中注有“*”号的部分为选讲内容，可根据学时多少或专业需要决定。

本书由赵军主编，负责全书内容的组织和定稿，柳明丽、高建新担任副主编，张英囡、于宝琦，李青山参与编写。本书共有 7 章。第 1 章由于宝琦编写；第 2、6 章由张英囡编写；第 3、4、5 章由柳明丽编写，第 7 章由李青山编写。刘永波审阅了全书，并对全书的内容和形式提出了许多宝贵建议。

本书在编写过程中得到了许多同事和朋友的支持和帮助，在此一并表示感谢。

由于编者水平有限，书中可能会有疏漏之处，恳请读者给予批评指正，并将意见及时反馈给我们，以便帮助我们不断改进。

编者

2008 年 10 月

目　　录

第 1 章 逻辑函数基础

【内容提要】

逻辑代数是分析和设计数字逻辑电路的基本工具。本章将首先介绍数字系统中分析和设计逻辑电路的基础知识，逻辑代数的基本概念、公式和定理，数字逻辑电路中的基本逻辑运算，然后介绍逻辑函数化简的基本方法，最后介绍几种常用逻辑函数的表示方法及其相互间的转换。

1.1 数字电路的特点

电子电路分为两大类：模拟电路和数字电路。在模拟电路中被传输、加工和处理的是模拟信号，这类信号的特点是在时间上和幅度上都是连续变化的，如电视图像信号以及传感器测量的温度、压力信号等。在数字电路中被传输、加工和处理的是数字信号，这类信号的特点为在时间上和幅度上都是离散的，是随时间不连续变化的脉冲信号。

与模拟电路相比，数字电路具有以下特点。

① 数字信号只有两种可能情况：有信号或者无信号，因此，数字电路只需要能够正确反映信号的有无，所以允许数值上存在一定范围的误差。组成数字电路的元件数值允许有较大的偏差，特别适宜集成化。

② 在数字电路中，晶体管工作在开关状态，即交替地处于饱和与截止两种状态；在模拟电路中，晶体管工作在放大状态。

③ 数字电路主要研究输入与输出之间的逻辑关系，采用的是逻辑代数、真值表、逻辑函数表达式、波形图和卡诺图等方法。

数字电路是计算机技术和各种数控、数显以及测量技术的基础。随着集成技术的发展，数字电路和计算机技术在各个领域都得到了广泛的应用，通信、控制和各种电器产品的数字化早已是大势所趋，数字照相机、DVD 和数字电视等数字化电子产品已进入寻常百姓人家。

【思考题】

1-1-1 和模拟电路相比，数字电路具有哪些优点？

1-1-2 数字信号和模拟信号各有什么特点？

1.2 数制和码制

1.2.1 数制

用数字量表示物理量的大小时，只用一位数码常常不够用，因而经常需要采用进位计数

的方法组成多位数码使用。将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。

数字电路采用的是二进制，只用 **1** 和 **0** 两个数表示，而这里的 **1** 和 **0** 并不表示数值的大小，它们所代表的是两种不同的逻辑状态，例如，用 **1** 和 **0** 分别表示一件事的是与非、真与假；开关的闭合与断开；晶体管的饱和导通与截止；电灯的亮与灭等。为了使读写方便简单，计算机汇编语言中经常采用十六进制来描述数据。

1. 十进制

在日常生活和工作中最常用的进位计数制是十进制。十进制数有 10 个数码：0～9，计数基数是 10 。超过 9 的数必须采用多位数来表示，低位与相邻高位之间的进位关系为“逢十进一”，故称为十进制。

一个十进制数可这样描述：

$$(756.56)_{10}=7\times10^2+5\times10^1+6\times10^0+5\times10^{-1}+6\times10^{-2}$$

其中 10^2、10^1、10^0、10^{-1} 和 10^{-2} 称为各位的位权，简称权。

上式中的下角 10 表示括号里的数字是十进制，有时也可用 D（Decimal）代替。

所以任意一个十进制数 D 均可展开为

$$D=\sum k_i\times10^i \tag{1-1}$$

式(1-1) 中的 k_i 是第 i 位的系数，它可以是 0～9 这 10 个数码中的任何一个。若整数部分的位数是 n，小数部分的位数为 m，则 i 包含 $n-1$～0 的所有正整数和 -1～$-m$ 的所有负整数。

若用 N 代替式(1-1) 中的 10，就可以得到任意进制（N 进制）数展开式的普遍形式

$$D=\sum k_i\times N^i \tag{1-2}$$

式(1-2) 中 i 的取值同式(1-1) 的规定一样。N 称为计数的基数，k_i 是第 i 位的系数，N_i 称为第 i 位的权。

2. 二进制

二进制是数字电路中应用最广泛的进位计数制。二进制数只有 **0** 和 **1** 两个数码，计数基数是 2。低位与相邻高位之间的进位关系为“逢二进一”，故称为二进制。

根据式(1-2)，一个二进制数可这样描述：

$$B=\sum k_i\times2^i \tag{1-3}$$

例如

$$(\mathbf{1101.11})_2=\mathbf{1}\times2^3+\mathbf{1}\times2^2+\mathbf{0}\times2^1+\mathbf{1}\times2^0+\mathbf{1}\times2^{-1}+\mathbf{1}\times2^{-2}$$

上式中从低位到高位的权分别为 2^{-2}、2^{-1}、2^0、2^1、2^2 和 2^3。

上式中的下角 2 表示括号里的数字是二进制，有时也可用 B（Binary）代替。

3. 十六进制

十六进制数有 16 个数码，分别用 0～9、A(10)、B(11)、C(12)、D(13)、E(14)、F(15)表示，计数基数是 16。低位与相邻高位之间的进位关系为“逢十六进一”，故称为十六进制。

一个十六进制数可这样描述：

$$H=\sum k_i\times16^i \tag{1-4}$$

例如

$$(7B5.7F)_{16}=7\times16^2+11\times16^1+5\times16^0+7\times16^{-1}+15\times16^{-2}$$

上式中的下角 16 有时也可用 H（Hexadecimal）代替。

4. 八进制

八进制数有 8 个数码，分别用 0～7 表示，计数基数是 8。低位与相邻高位之间的进位关系为“逢八进一”，故称为八进制。

一个八进制数可这样描述：

$$O=\sum k_i \times 8^i \tag{1-5}$$

例如

$$(715.41)_8=7\times 8^2+1\times 8^1+5\times 8^0+4\times 8^{-1}+1\times 8^{-2}$$

上式中的下角 8 有时也可用 O(Octal) 代替。

1.2.2　数制转换

1. 二进制数转换成十进制数

称二进制数转换成等值的十进制数为二-十转换。在这种转换中，只要对二进制数采用权展开式展开，然后将所有各项的数值按十进制相加，就可以转换得到等值的十进制数，例如

$$(\mathbf{101.11})_2=\mathbf{1}\times 2^2+\mathbf{0}\times 2^1+\mathbf{1}\times 2^0+\mathbf{1}\times 2^{-1}+\mathbf{1}\times 2^{-2}=(5.75)_{10}$$

2. 十进制数转换成二进制数

十进制数转换成等值的二进制数称为十-二转换。

（1）整数部分的转换　十进制整数转换为二进制整数，采用的是 10 除以 2 取余的方法，即将十进制数反复除以 2，直到商为 0 为止。而各次相除所得余数就是二进制数由低位到高位的各位数字，例如，将 $(185)_{10}$ 化为二进制数可按如下过程进行。

2 | 185	…………	余数=**1**=k_0　(最低位)
2 | 92	…………	余数=**0**=k_1
2 | 46	…………	余数=**0**=k_2
2 | 23	…………	余数=**1**=k_3
2 | 11	…………	余数=**1**=k_4
2 | 5	…………	余数=**1**=k_5
2 | 2	…………	余数=**0**=k_6
2 | 1	…………	余数=**1**=k_7　(最高位)
0		

所以，$(185)_{10}=(\mathbf{10111001})_2$

（2）小数部分的转换　十进制小数转换为二进制小数，采用的是 10 乘以 2 取整的方法，即将十进制小数反复乘以 2，而各次相乘所得的整数就是二进制小数由高位到低位的各位数字。

例如，将 $(0.5625)_{10}$ 转化为二进制小数可按如下过程进行。

$$\begin{array}{r} 0.5625 \\ \times\quad 2 \\ \hline 1.1250 \end{array}\cdots\cdots\text{整数部分}=1=k_{-1}\quad\text{（最高位）}$$

$$\begin{array}{r} 0.1250 \\ \times\quad 2 \\ \hline 0.2500 \end{array}\cdots\cdots\text{整数部分}=0=k_{-2}$$

$$\begin{array}{r} 0.2500 \\ \times \quad 2 \\ \hline 0.5000 \end{array} \cdots\cdots\cdots\cdots\cdots\cdots\text{整数部分}=0=k_{-3}$$

$$\begin{array}{r} 0.5000 \\ \times \quad 2 \\ \hline 1.0000 \end{array} \cdots\cdots\cdots\cdots\cdots\cdots\text{整数部分}=1=k_{-4}\quad\text{（最低位）}$$

所以，$(0.5625)_{10}=(\mathbf{0.1001})_2$

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　二进制数转换成等值的八进制数称为二-八转换。在这种转换中，只需将二进制数的整数部分从右向左每三位分为一组，小数部分从左向右每三位分为一组，每组用一对应的八进制数表示，即可实现二进制向八进制的转换，例如，将 $(\mathbf{1011110.1011001})_2$ 转换为八进制数时可得：

$$\begin{array}{cccccc} (\mathbf{001}, & \mathbf{011}, & \mathbf{110.} & \mathbf{101}, & \mathbf{100}, & \mathbf{100})_2 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ =(\ 1 & 3 & 6. & 5 & 4 & 4)_8 \end{array}$$

（2）八进制数转换成二进制数　八进制数转换成等值的二进制数称为八-二转换。转换时只需将八进制数的每一位用等值的三位二进制数代替即可，例如，将 $(254.73)_8$ 转换为二进制数时可得：

$$\begin{array}{ccccc} (\ 2 & 5 & 4. & 7 & 3\ )_8 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ =(\ \mathbf{010} & \mathbf{101} & \mathbf{100.} & \mathbf{111} & \mathbf{011})_2 \end{array}$$

（3）二进制数转换成十六进制数　二进制数转换成等值的十六进制数称为二-十六转换。转换时只需将二进制数的整数部分从右向左每四位分为一组，小数部分从左向右每四位分为一组，每组用一对应的十六进制数表示，即可实现二进制向十六进制的转换，例如，将 $(\mathbf{1011110.1011001})_2$ 转换为十六进制数时可得：

$$\begin{array}{cccc} (\mathbf{0101}, & \mathbf{1110.} & \mathbf{1011}, & \mathbf{0010})_2 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ =(\ 5 & \mathrm{E}. & \mathrm{B} & 2)_{16} \end{array}$$

（4）十六进制数转换成二进制数　十六进制数转换成等值的二进制数称为十六-二转换。转换时只需将十六进制数的每一位用等值的四位二进制数代替即可，例如，将 $(2\mathrm{DF.E2})_{16}$ 转换为二进制时可得：

$$\begin{array}{ccccc} (\ 2 & \mathrm{D} & \mathrm{F}. & \mathrm{E} & 5\ )_{16} \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ =(\ \mathbf{0010} & \mathbf{1101} & \mathbf{1111.} & \mathbf{1110} & \mathbf{0101})_2 \end{array}$$

1.2.3　码制

用二进制数表示文字、符号等信息的过程称为编码，这些只用来表示不同的事物，而已没有表示数量大小含意的数码称为代码。为了记忆和处理方便，在编制代码时应遵循一定的规则，这些规则就叫做码制。

1. 二-十进制编码

在数字电路中，通常用四位二进制代码来表示一位十进制数的 0～9 这十个状态，常将

这些代码称为二-十进制代码（Binary Coded Decimal，BCD）。换句话说，BCD 码是用四位二进制数来表示一个十进制数的代码。

由于用四位二进制数可以组成 $2^4=16$ 个不同的二进制码组，所以用其中的十个码组表示十进制数 0～9，剩下的六个码组为多余码组，又称为冗余码组。从十六个二进制码组中任意取十个码组的方案有很多种，因此产生了多种 BCD 码，其中比较常见的 BCD 码有 8421 码、2421 码、余 3 码等。它们的编码如表 1-1 所示。

表 1-1　3 种常见的 BCD 代码

十进制	8421 码	2421 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**
权	8421	2421	

（1）8421 BCD 码　8421 码是一种最常见的 BCD 码。在这种编码方式中，每一位二进制码的 **1** 都代表一个固定的数值，把每一位 **1** 代表的十进制数加起来，得到的结果就是所代表的十进制数。8421 码是一种有权码，每一位的 **1** 代表的十进制数即为这一位的权，而每一位的权又是固定不变的，所以它属于恒权代码。由于它的四位二进制码各位的权值从左至右分别为 8、4、2、1，所以将其称做 8421 码。

（2）2421 BCD 码　2421 BCD 码也是一种有权码，它的四位二进制码各位的权从左到右分别为 2、4、2、1，其中，0 和 9、1 和 8、2 和 7、3 和 6、4 和 5 互为反码。应该指出的是，2421 BCD 码的编码方案不是唯一的。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 **1101**，也可编成 **0111**。

（3）余 3 BCD 码　余 3 BCD 码与 8421 码不同，它所表示的十进制数 0～9 的代码数值等于该代码各位的加权数之和再加常数 3，例如十进制数 7 的 8421 码是 0111，则余 3 码是 1010。

余 3 码是一种特殊的有权码，将二进制码为 1 的各码位加权之和，与它所代表的十进制数相比，始终相差一个固定的常数，所以又称为余权码。

2. 循环码

循环码的各位没有确定的权值，是无权码。循环码是一种常用的单位距离码，它的编码特点是：任何一个相邻码组（包括首尾两个码组）之间仅有一个码位不同。用循环码来表示十进制数时，为使 0 和 9 的码组只有一个码位不同，常采用余 3 循环码，它是从四位循环码的十六个码组中去掉首尾各三个码组而构成的，如表 1-2 所示。

3. ASCII 码

在数字电路设备，特别是计算机中，除了需要传送数字，常常还需要传送如字母、字符以及控制信号等这样的信息，因此，就需要采用一种符号，即数字编码。目前最普遍采用的是美国标准信息交换码，即 ASCII 码（American Standard Code for Information Interchange），如表 1-3 所示。

表 1-2 循环码

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

表 1-3 ASCII 字符编码表

ASCII 值	字符	ASCII 值	字符	ASCII 值	字符	ASCII 值	字符
0	NUL	32	(space)	64	@	96	‘
1	SOH	33	!	65	A	97	a
2	STX	34	"	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEL	39	’	71	G	103	g
8	BS	40	(	72	H	104	h
9	HT	41	)	73	I	105	i
10	LF	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF	44	,	76	L	108	l
13	CR	45	—	77	M	109	m
14	SO	46	.	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DC1	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	r
19	DC3	51	3	83	S	115	s
20	DC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u
22	SYN	54	6	86	V	118	v
23	ETB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	:	90	Z	122	z
27	ESC	59	;	91	[	123	{
28	FS	60	<	92	\	124	|
29	GS	61	=	93	]	125	}
30	RS	62	>	94	∧	126	~
31	US	63	?	95	_	127	DEL

【思考题】

1-2-1　在数字电路中为什么采用二进制？它有何优点？

1-2-2　比较下列各数，找出最大数和最小数：

(1) $(105)_{10}$；(2) $(F8)_{16}$；(3) $(1001001)_2$。

1-2-3　六位二进制数的最大值对应的十进制数是多少？

1-2-4　循环码的特点是什么？为什么说它是可靠性代码？

1.3　逻辑代数

逻辑代数是反映和处理客观事物间逻辑关系的数学方法。逻辑代数中也用字母表示变量，这种变量称为逻辑变量，简称变量。在二值逻辑中，每个逻辑变量的取值只有 **0** 和 **1** 两种可能，代表两种不同的逻辑状态。在逻辑代数中，有些公式和定理与普通代数并无区别，有些则完全不同。

1.3.1　基本逻辑运算

当两个二进制数码表示不同的逻辑状态时，它们之间可以按照制定的某种因果关系进行所谓的逻辑运算。这种逻辑运算与算术运算有着本质上的不同。

逻辑代数中，基本的逻辑关系有**与**逻辑、**或**逻辑和**非**逻辑三种，与之相对应的逻辑运算有**与**运算、**或**运算和**非**运算。

1. 与逻辑

只有决定某个事件的全部条件同时具备时，这件事才会发生，这种逻辑关系称为**与**逻辑，或者称为逻辑相乘。

在如图 1-1 所示电路中，只有开关 A 与开关 B 都闭合时，灯 Y 才会亮，所以对于灯 Y 亮这件事来说，开关 A、开关 B 闭合是与的逻辑关系。若以 A、B 表示开关的状态，且以 **1** 表示开关闭合，**0** 表示开关断开；以 Y 表示灯的状态，且以 **1** 表示灯亮，**0** 表示灯灭，则可以列出以 **1**、**0** 表示的**与**逻辑关系的图表，如表 1-4 所示。这种图表叫做逻辑真值表，简称为真值表。

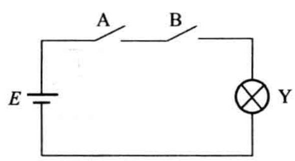


图 1-1　**与**逻辑关系电路

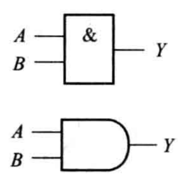


图 1-2　**与**门符号

表 1-4　与逻辑真值表

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

由该表可看出逻辑变量 A、B 与函数 Y 之间的关系满足**与**运算规律（逻辑相乘），可表示为

$$Y = A \cdot B \tag{1-6}$$

为了简化书写，“·”也可以省略。

实现**与**运算的电路称为与门，其逻辑符号如图 1-2 所示。

2. 或逻辑

决定某个事件的各个条件中，只要有一个具备，这件事就会发生，这种逻辑关系称为**或**逻辑，或者称为逻辑相加。

在如图 1-3 所示电路中，只要开关 A 和开关 B 中有一个闭合时，灯 Y 就会亮，所以对于灯 Y 亮这件事来说，开关 A、开关 B 闭合是**或**的逻辑关系。**或**逻辑真值表如表 1-5 所示。

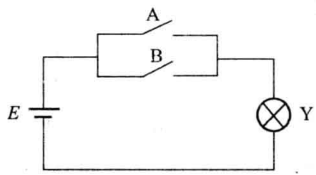


图 1-3　**或逻辑关系电路**

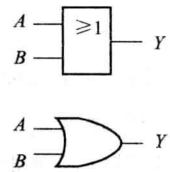


图 1-4　**或门符号**

表 1-5　或逻辑真值表

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

由该表可看出逻辑变量 A、B 与函数 Y 之间的关系满足**或**运算规律（逻辑加），可表示为

$$Y=A+B \tag{1-7}$$

实现**或**运算的电路称为**或**门，其逻辑符号如图 1-4 所示。

3. 非逻辑

只要条件具备了，事件就不会发生；而条件不具备时事件一定发生，这种互相否定的因果关系称为**非**逻辑，或者称为逻辑求反。

在如图 1-5 所示电路中，当开关 A 闭合时，灯 Y 灭；当开关 A 断开时，灯 Y 亮，所以对于灯 Y 亮这件事来说，与开关 A 闭合是一种**非**的逻辑关系。**非**逻辑真值表如表 1-6 所示。

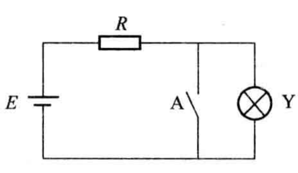


图 1-5　**非逻辑关系电路**

表 1-6　非逻辑真值表

A	Y
0	1
1	0

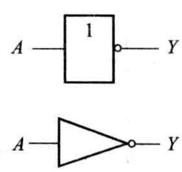


图 1-6　**非门符号**

由该表可看出逻辑变量 A 与函数 Y 之间的关系满足**非**运算规律，可表示为

$$Y=\overline{A} \tag{1-8}$$

实现**非**运算的电路称为**非**门，其逻辑符号如图 1-6 所示。

4. 复合逻辑运算

在逻辑代数中，除了**与**、**或**、**非**三种基本逻辑运算外，经常用到的还有**与非**、**或非**、**与或非**、**异或**、**同或**等复合逻辑运算。这些复合逻辑运算的真值表如表 1-7～表 1-11 所示，它们的逻辑符号如图 1-7 所示。

表 1-7　与非逻辑真值表

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

表 1-8　或非逻辑真值表

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

表 1-9　与或非逻辑真值表

A	B	C	D	Y	A	B	C	D	Y
0	0	0	0	1	1	0	0	0	1
0	0	0	1	1	1	0	0	1	1
0	0	1	0	1	1	0	1	0	1
0	0	1	1	0	1	0	1	1	0
0	1	0	0	1	1	1	0	0	0
0	1	0	1	1	1	1	0	1	0
0	1	1	0	1	1	1	1	0	0
0	1	1	1	0	1	1	1	1	0

表 1-10　异或逻辑真值表

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

表 1-11　同或逻辑真值表

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

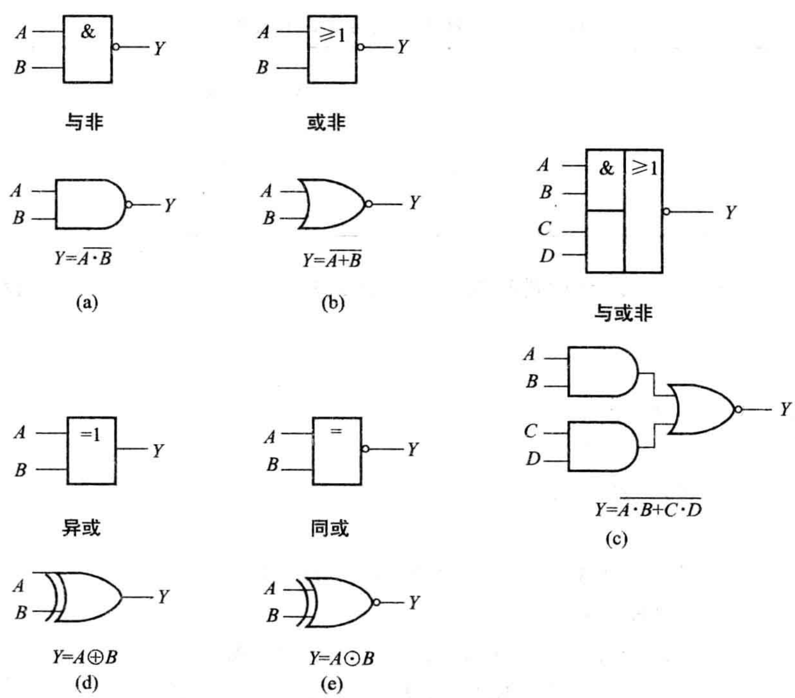


图 1-7　复合逻辑符号

1.3.2 逻辑代数的基本公式和定理

1. 逻辑代数的基本公式

（1）逻辑常量运算公式　逻辑常量只有**0**和**1**，而最基本的逻辑运算只有**与**、**或**、**非**三种，所以常量之间的运算关系见表1-12。

表1-12　逻辑常量运算公式

与运算	或运算	非运算
$0\cdot 0=0$	$0+0=0$	$\overline{1}=0$
$0\cdot 1=0$	$0+1=1$	
$1\cdot 0=0$	$1+0=1$	$\overline{0}=1$
$1\cdot 1=1$	$1+1=1$	

（2）逻辑变量与常量运算公式　表1-13给出了逻辑变量与常量之间的运算公式，其中A为逻辑变量。

表1-13　逻辑变量与常量运算公式

与运算	或运算	非运算
$A\cdot 0=0$	$A+0=A$	$\overline{\overline{A}}=A$
$A\cdot 1=A$	$A+1=1$	
$A\cdot A=A$	$A+A=A$	
$A\cdot\overline{A}=0$	$A+\overline{A}=1$	

2. 逻辑代数的基本定律

（1）与普通代数相似的定律　表1-14给出了与普通代数相似的定律：交换律、结合律、分配律。

表1-14　交换律、结合律、分配律

交换律	$A+B=B+A$ $A\cdot B=B\cdot A$
结合律	$(A+B)+C=A+(B+C)$ $(A\cdot B)\cdot C=A\cdot(B\cdot C)$
分配律	$A\cdot(B+C)=A\cdot B+A\cdot C$ $A+B\cdot C=(A+B)\cdot(A+C)$

（2）吸收律　吸收律可以利用逻辑代数基本公式推导得到，它们是逻辑函数化简中常用的基本定律。

① $AB+A\overline{B}=A$　(1-9)

证明：$AB+A\overline{B}=A(B+\overline{B})=A$

② $A+AB=A$　(1-10)

证明：$A+AB=A(1+B)=A$

③ $A+\overline{A}B=A+B$　(1-11)

证明：$A+\overline{A}B=(A+\overline{A})(A+B)=1\cdot(A+B)=A+B$

④ $AB+\overline{A}C+BC=AB+\overline{A}C$　(1-12)

证明：$AB+\overline{A}C+BC=AB+\overline{A}C+BC(A+\overline{A})=AB+\overline{A}C+ABC+\overline{A}BC$

$=(AB+ABC)+(\overline{A}C+\overline{A}BC)=AB+\overline{A}C$

推论　$AB+\overline{A}C+BCD=AB+\overline{A}C$　(1-13)

(3) 摩根定律　摩根定律又称为反演律，它有两种形式：

$$\overline{A\cdot B}=\overline{A}+\overline{B} \tag{1-14}$$

$$\overline{A+B}=\overline{A}\cdot\overline{B} \tag{1-15}$$

摩根定律可利用真值表来证明，如表 1-15 和表 1-16 所示。

表 1-15　$\overline{A\cdot B}=\overline{A}+\overline{B}$ 的证明

A	B	$\overline{A\cdot B}$	$\overline{A}+\overline{B}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

表 1-16　$\overline{A+B}=\overline{A}\cdot\overline{B}$ 的证明

A	B	$\overline{A+B}$	$\overline{A}\cdot\overline{B}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

3. 逻辑代数的三个基本定理

(1) 代入定理　任意一个包含变量 A 的逻辑等式都可以将等式两边所有的变量 A 用另外一个逻辑函数表达式替代，等式依然成立，这就是所谓的代入定理。利用代入定理可以把基本公式推广为多变量的形式，从而扩大了逻辑等式的应用范围。

【例 1-1】 用代入定理推导摩根定律多变量的情况。

解： 已知二变量摩根定律为

$$\overline{A\cdot B}=\overline{A}+\overline{B},\ \overline{A+B}=\overline{A}\cdot\overline{B}$$

若分别用 $Y=BC$，$Y=B+C$ 代替两等式中的 B，则

$$\overline{A\cdot(BC)}=\overline{A}+\overline{BC}=\overline{A}+\overline{B}+\overline{C}$$

$$\overline{A+(B+C)}=\overline{A}\cdot\overline{B+C}=\overline{A}\cdot\overline{B}\cdot\overline{C}$$

所以摩根定律可以推广到多个变量，其逻辑表达式如下：

$$\begin{cases}\overline{A\cdot B\cdot C\cdot\cdots}=\overline{A}+\overline{B}+\overline{C}+\cdots\\ \overline{A+B+C+\cdots}=\overline{A}\cdot\overline{B}\cdot\overline{C}\cdot\cdots\end{cases} \tag{1-16}$$

(2) 反演定理　对于任意一个逻辑函数表达式 Y，如果将其中的"·"换成"+"，"+"换成"·"；**0** 换成 **1**，**1** 换成 **0**；原变量换成反变量，反变量换成原变量，那么得到的逻辑函数表达式就是 Y 的反函数 $\overline{Y}$。这个规则称为反演定理。

反演定理为求已知逻辑函数的反函数提供了方便。在使用反演定理时必须注意以下两点。

① 注意运算符号的优先顺序。先括号，再乘积，最后加。

② 反变量换成原变量只对单个变量有效。不属于单个变量上的反号应保留不变。

【例 1-2】 已知逻辑函数 $Y=\overline{\overline{A\overline{B}C}+\overline{C}D}(AC+BD)$，求 $\overline{Y}$。

解： 根据反演定理，得

$$\begin{aligned}\overline{Y}&=\overline{\overline{(\overline{A}+B+\overline{C})}\cdot(C+\overline{D})}+(\overline{A}+\overline{C})(\overline{B}+\overline{D})=\overline{\overline{\overline{A}+B+\overline{C}}}+\overline{C+\overline{D}}+\overline{A}\,\overline{B}+\overline{A}\,\overline{D}+\overline{B}\,\overline{C}+\overline{C}\,\overline{D}\\&=(\overline{A}+B)C+\overline{C}D+\overline{A}\,\overline{B}+\overline{A}\,\overline{D}+\overline{B}\,\overline{C}+\overline{C}\,\overline{D}=\overline{A}C+BC+\overline{C}D+\overline{A}\,\overline{B}+\overline{A}\,\overline{D}+\overline{B}\,\overline{C}+\overline{C}\,\overline{D}\\&=\overline{A}+B+\overline{C}\end{aligned}$$

(3) 对偶定理　对于任意一个逻辑函数表达式 Y，如果将其中的"·"换成"+"，"+"换成"·"；**0** 换成 **1**，**1** 换成 **0**，那么就会得到一个新的逻辑函数表达式 Y'，Y' 称为 Y 的对偶式。

例如：$Y=A+\overline{A}B=A+B$，　则 $Y'=A\cdot(\overline{A}+B)=AB$