第 1 章　数字逻辑基础

内容提要：本章介绍分析数字电路逻辑功能的数学方法。文中首先介绍了数制、编码，然后介绍了逻辑代数的基本公式、常用公式基本定理、逻辑函数及其表示方法，最后介绍如何利用公式、卡诺图等方法来简逻辑函数。

1.1　数制与编码

1.1.1　数制

数制是人们对数量计数的一种统计规则。在日常生活中经常遇到的是十进制。在数字系统中，广泛采用的则是二进制，八进制和十六进制。

一种进位计数包含着两个基本要素。

(1) 基数：基数是计数制中所用到的数码的个数，常用 R 表示。如十进制中，包含 0，1，2，…，9 等 10 个数码。进位规则为“逢 10 进 1”。所以它的基数 $R=10$。

(2) 位权：处在不同数位的数码，代表着不同的数值，每一个数位的数值是由该位数码的值乘以处在这位的一个固定常数。不同数位上的固定常数称为位权值，简称位权。例如，十进制数个位的位权值是 1，十位的位权值是 10^1，百位是 10^2，以此类推。譬如十进制数 1111，各位数码均为 1，由于它们所处的数位不一样，那么它们所表示的数值不一样。犹如军内干部有司令、军长、师长、团长、营长、连长、排长、班长等职务，他们都属军人，但他们所处的地位不同，那么人们给予他们的权力是不一样的。又如杆秤，使用同一个秤砣，它所处的位置不一样，那么所表示的重量也是不一样的。

下面对常用的几种数制一一介绍。

一、十进制(Decimal，计算机中缩写为 D)

基数 $R=10$ 的数制称为十进制。一个十进制数按权展开为

$$(N)_{10}=a_{n-1}a_{n-2}\cdots a_2a_1a_0\cdot a_{-1}\cdots a_{-m}=\sum_{i=-m}^{n-1}a_i10^i \tag{1.1.1}$$

式中，n 为整数位数；m 为小数位数；10 为基数，也称为模；10^i 为第 i 位的位权值。

特点：① 有 0，1，…，9 等 10 个数码(数符)；②“逢 10 进 1”。

二、R 进制

基数为 R 的数制称为 R 进制。进位规则为“逢 R 进 1”。有 0，1，…，$R-1$ 个数码(数符)。按权展开为

$$(N)_R=\sum_{i=-m}^{n-1}a_iR^i \tag{1.1.2}$$

式中，n 为整数位数；m 为小数位数；a_i 为第 i 位数码；R 为基数；R^i 为第 i 位的位权值。

三、二进制(Binary，计算机中缩写为 B)

基数 $R=2$ 的数制为二进制，有 0，1 两个数码，进位规则为“逢 2 进 1”。按权展开为

$$(N)_2 = \sum_{i=-m}^{n-1} a_i 2^i \tag{1.1.3}$$

四、八进制(Octau,计算机中缩写为 O)

基数 $R=8$ 的数制为八进制。有 0,1,…,7 等 8 个数码,进位规则为"逢 8 进 1"。按权展开为

$$(N)_8 = \sum_{i=-m}^{n-1} a_i 8^i \tag{1.1.4}$$

五、十六进制(Hexdelinal,计算机中缩写为 H)

基数 $R=16$ 的数制为十六进制。有 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F 十六个数码,进位规则为"逢 16 进 1"。按权展开为

$$(N)_{16} = \sum_{i=-m}^{n-1} a_i 16^i \tag{1.1.5}$$

1.1.2 数制间的转换

一、各种进制转换成十进制

基数为 R 的 R 进制转换成十进制的方法很简单,只要按式(1.1.2)就可求得。

【例 1.1.1】 一个二进制数为 $(1010.011)_2$ 转换为十进制数。

解:
$$(1010.011)_2 = \sum_{i=-m}^{n-1} a_i 2^i = 1\times 2^3 + 1\times 2^1 + 1\times 2^{-2} + 1\times 2^{-3}$$
$$= (10.375)_{10}$$

二、十进制转换成 R 进制

一个任意的十进制数可以由整数部分和小数部分构成,若设整数部分为 M_1,小数部分为 M_2,则

$$(M)_{10} = (M_1)_{10} + (M_2)_{10}$$

将它转换成 R 进制,根据式(1.1.2)得

$$(M)_{10} = \sum_{i=-m}^{n-1} a_i R^i$$
$$= a_{n-1}R^{n-1} + a_{n-2}R^{n-2} + \cdots + a_2R^2 + a_1R + a_0 +$$
$$a_{-1}R^{-1} + a_{-2}R^{-2} + \cdots + a_{-m}R^{-m} \tag{1.1.6}$$

于是

整数部分为: $(M_1)_{10} = a_{n-1}R^{n-1} + a_{n-2}R^{n-2} + \cdots + a_2R^2 + a_1R + a_0$ (1.1.7)

小数部分为: $(M_2)_{10} = a_{-1}R^{-1} + a_{-2}R^{-2} + \cdots + a_{-m}R^{-m}$ (1.1.8)

现在的问题是如何确定 a_i 的值。

先观察整数部分:

$(M_1)_{10} \div R$,得商为 $a_{n-1}R^{n-2} + a_{n-2}R^{n-3} + \cdots + a_2R + a_1$ …… 余数为 a_0

将上式商再除以 R 得

商为 $a_{n-1}R^{n-3} + a_{n-2}R^{n-4} + \cdots + a_2$ …… 余数为 a_1

依次类推,就可以求得全部的 $a_i(i=0,1,2,\cdots,n-1)$。

电子工业出版社版权所有 盗版必究

我们将这种方法取名为**除以R取余法,逆序排列**。其中 R 为基数。

再观察小数部分:将式(1.1.8)两边同乘以 R 得,整数部分为 a_{-1},小数部分则为

$$a_{-2}R^{-1}+a_{-3}R^{-2}+\cdots+a_{-m}R^{-m+1}$$

然后将小数部分再乘以 R 得,整数部分为 a_{-2},小数部分则为

$$a_{-3}R^{-1}+\cdots+a_{-m}R^{-m+2}$$

依次类推,就可求得全部的 $a_i(i=-1,-2,\cdots,-m)$。

最后一步再乘之后,还可能存在小数部分,不妨设为 e,e 称为剩余误差。其值为

$$e<R^{-m}$$

我们将这种方法取名为**乘以R取整法,顺序排列**。

【例 1.1.2】　将十进制数 10.375 转换成二进制数($R=2$)。

解:将十进制数 10.375 的整数部分和小数部分分别转换。

整数部分转换采用除以 R 取余法(在本例中 $R=2$),即

		余数	对应二进制数码(数符)	
2	10			
2	5	0	a_0	
2	2	1	a_1	
2	1	0	a_2	
	0	1	a_3	于是 $(10)_{10}=(1010)_2$

小数部分采用乘以 R 取整法(在本例中 $R=2$):

	整数部分	对应二进制数码(数符)
$0.375\times2=0.75$	0	a_{-1}
$0.75\times2=1.5$	1	a_{-2}
$0.5\times2=1.0$	1	a_{-3}

剩余误差 $e=0$

于是　　$(0.375)_{10}=(.011)_2+e=(.011)_2$

最后得到　　$(10.375)_{10}=(1010.011)_2$

三、二进制与八进制、十六进制之间的转换

1. 八进制转换为二进制

把八进制数每位数用 3 位二进制数表示即可。

【例 1.1.3】　将八进制数$(312.64)_8$转换成二进制数。

解:3　1　2　.　6　4

011　001　010　.　110　100

于是　　$(312.64)_8=(011001010.110100)_2=(11001010.1101)_2$

2. 二进制转换为八进制

二进制数转换为八进制数时,以小数点为界,分别向左、向右以 3 位为一组,最高位不到 3 位的**用0补齐**,最低位不到 3 位的也**用0补齐**,然后将每 3 位的二进制数用相应的八进制数表示。

【例 1.1.4】　将二进制数$(10110.11)_2$转换成八进制数。

解:二进制数　　010　110. 110

电子工业出版社版权所有 盗版必究

对应的八进制数　2　6　.　6

于是　$(10110.11)_2 = (26.6)_8$

3. 十六进制转换为二进制

将每位十六进制数用相应的4位二进制数表示。

【例 1.1.5】 将十六进制数$(21A.5)_{16}$转换成二进制数。

解：十六进制数　2　1　A　.　5

对应的二进制数　0010　0001　1010　.　0101

于是　$(21A.5)_{16} = (001000011010.0101)_2 = (1000011010.0101)_2$

4. 二进制转换为十六进制

二进制数转换为十六进制数时，以小数点为界，分别向左、向右以4位为一组，最高位不到4位者用0补齐，最低位不到4位者也用0补齐，然后将4位二进制数用相应的十六进制数表示。

【例 1.1.6】 将二进制数$(1100101.101)_2$转换为十六进制数

解：二进制数　0110　0101　.　1010

对应的十六进制数　6　5　.　A

于是　$(1100101.101)_2 = (01100101.1010)_2 = (65.A)_{16}$

5. 八进制与十六进制之间的转换

八进制(或十六进制)转换成十六进制(或八进制)，先将八进制(或十六进制)转换为二进制，然后按二进制转换十六进制(或八进制)的步骤进行转换。

1.1.3 编码

编码就是用二进制码来表示给定的信息符号。这个信息符号可以是十进制数符0,1,2,…,9;字符A、B、C、…;运算符“+”、“−”、“=”等。下面介绍几种常用的编码。

一、带符号的二进制编码

在数字系统中，正、负数的表示方法是：把一个数最高位作为符号位，用“0”表示“+”;用“1”表示“−”。连同符号位一起作为一个数，称为机器数，它的原来的数值形式则称为这个机器数的真值。

例如：真值$X_1 = +0.1101$;$X_2 = -0.1101$

表示成机器数为：$X_1 = 0.1101$;$X_2 = 1.1101$

在数字系统中，表示机器数的方法很多，目前常用的有原码、反码和补码。

1. 原码(True Form)

原码表示法又称符号－数值表示法。正数的符号位用“0”表示;负数的符号位用“1”表示;数值部分保持不变。

例如：真值$X = -1101$

$(X)_{原} = 11101$

2. 反码(One's complement)

反码的符号表示方法与原码相同，正数反码的数值部分保持不变，而负数反码的数值是原码的数值按位求反。

电子工业出版社版权所有 盗版必究

例如：真值 $X_1 = +1101$，则 $(X_1)_{反} = 01101$

真值 $X_2 = -1101$，则 $(X_2)_{反} = 10010$

3. 补码(Two's complement)

补码的符号表示方法和原码相同。正数的补码数值部分也与原码相同。负数的补码是这样得到的：将数值部分按位求反，再在最低位加 1。

例如：真值 $X_1 = +1101$，则 $(X_1)_{补} = (X_1)_{原} = (X_1)_{反} = 01101$

真值 $X_2 = -1101$，则 $(X_1)_{补} = 10011$

小结：正数的原码、反码与补码是一样的，均等于该数的真值；负数的原码、反码与补码的符号均为 1，仅数值部分不相同，原码的数值部分不变，反码的数值部分按位取反，而补码的数值部分仅仅在反码的最后一位加 1 即可。

二、带小数点的数的编码

一个数既有小数部分又有整数部分，在数字系统中是如何表示呢？一般有两种方式：定点表示和浮点表示。

任何数制的数 N，均可以表示为

$$N = R^E \times M \tag{1.1.9}$$

式中，R 为基数；E 为阶码，取值为整数；M 为数 N 的尾数，取值为整数或小数。

例如：$(N)_{10} = (0.25)_{10} = 10^{-2} \times 25$，则数 N 的阶码 $E = -2$，尾数 $M = 25$。

对于二进制数，可表示为

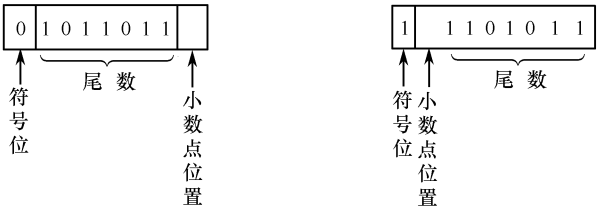
$$N = 2^E \times M \tag{1.1.10}$$

1. 定点表示法

所谓定点表示法就是小数点的位置在数中是固定不变的，这个固定位置是事先约定的，不必用符号表示。

当 $E = 0$，尾数 M 为纯整数时，则认为小数点在尾数 M 最低位右边，为整数定点。

例如：$N = +1011011$，则可表示为图 1.1.1(a) 的形式。



(a) 整数定点　　(b) 小数定点

图 1.1.1　定点表示法示意图

当 $E = 0$，尾数 M 为纯小数时，则认为小数点的位置在 M 最高位的左边，定点数只能表示小数，为定点小数。

例如：$N = -0.1101011$，则表示为图 1.1.1(b) 的形式。

定点表示法，数 N 的范围是有限的，在定点小数时，当用 8 位二进制数表示一个数时，1 位符号位，7 位表示数值，最大值取值为 $(0.1111111)_2 = (127 \div 128)_{10}$，最小数取值为 $(0.0000001)_2 = (1 \div 128)_{10}$。

电子工业出版社版权所有 盗版必究

2. 浮点表示法

在一个数中小数点的位置不是固定不变的,而是可以变化的,这种表示法称为浮点表示法。

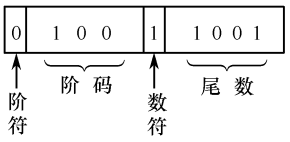


图 1.1.2 浮点表示法示意图

例如:二进制数$(N)_2 = 2^{100} \times (-1001)$可表示为如图1.1.2的形式。

其中:尾数M表示数N的全部有效数字;而阶码E指明了小数点的位置。小数点移动的规则是小数点向左移一位,相当于尾数的数码向右移一位,而阶码加1,如图1.1.2所示。

表1.1.1中列出了常见的几种十进制代码,它们的编码规则各不相同。

表 1.1.1 几种常见的十进制代码

编码种类 / 十进制数	8421码(BCD代码)	余3码	2421码	5211码	余3循环码
0	0000	0011	0000	0000	0010
1	0001	0100	0001	0001	0110
2	0010	0101	0010	0100	0111
3	0011	0110	0011	0101	0101
4	0100	0111	0100	0111	0100
5	0101	1000	1011	1000	1100
6	0110	1001	1100	1001	1101
7	0111	1010	1101	1100	1111
8	1000	1011	1110	1101	1110
9	1001	1100	1111	1111	1010
权	8421		2421	5211	

8421码又称BCD(Binary Coded Decimal)码,是十进制代码中最常用的一种。在这种编码方式中,每一位二值代码的1都代表一个固定数值,将每一位的1代表的十进制数加起来,得到的结果就是它所代表的十进制数码。由于代码中从左到右每一位的1分别表示8、4、2、1,所以将这种代码称为8421码。每一位的1代表的十进制数称为这一位的权。8421码中每一位的权是固定不变的,它属于恒权代码。

余3码的编码规则与8421码不同,如果把每一个余3码看作4位二进制数,则它的数值要比它所表示的十进制数码多3,故而将这种代码称为余3码。

如果将两个余3码相加,所得的和将比十进制数和所对应的二进制数多6。因此,在用余3码进行十进制加法运算时,若两数之和为10,正好等于二进制数的16,于是便从高位自动产生进位信号。

此外,从表1.1.1中还可以看出,0和9、1和8、2和7、3和6、4和5的余3码互为反码,这对于求取对10的补码是很方便的。

余3码不是恒权代码。如果试图将每个代码视为二进制数,并使它等效的十进制数与所表示的代码相等,那么代码中每一位的1所代表的十进制数在各个代码中不能是固定的。

2421码是一种恒权代码,它的0和9、1和8、2和7、3和6、4和5也互为反码,这个特点和余3码相仿。

电子工业出版社版权所有 盗版必究

5211 码是另一种恒权代码。待学了第 6 章中计数器的分频作用后可以发现，如果按 8421 码接成十进制计数器，则连续输入计数脉冲时，4 个触发器输出脉冲对于计数脉冲的分频比从低位到高位依次为 5∶2∶1∶1。可见，5211 码每一位的权正好与 8421 码十进制计数器 4 个触发器输出脉冲的分频比相对应。这种对应关系在构成某些数字系统时很有用。

余 3 循环码是一种变权码，每一位的 1 在不同代码中并不代表固定的数值。它的主要特点是相邻的两个代码之间仅有一位的状态不同。

三、格雷码

格雷码(Gray Code) 又称循环码。从表 1.1.2 的 4 位格雷码编码表中可以看出格雷码的构成方法，这就是每一位的状态变化都按一定的顺序循环。如果从 0000 开始，最右边一位的状态按 0110 顺序循环变化，右边第二位的状态按 00111100/ 顺序循环变化，右边第三位按 0000111111110000/ 顺序循环变化。可见，自右向左，每一位状态循环中连续的 0、1 数目增加一倍。由于 4 位格雷码只有 16 个，所以最左边一位的状态只有半个循环，即 0000000011111111。按照上述原则，我们就很容易得到更多位数的格雷码。

表 1.1.2　4 位格雷码与二进制代码的比较

编码顺序	二进制代码	格雷码
0	0000	0000
1	001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

与普通的二进制代码相比，格雷码的最大优点就在于当它按照表 1.1.2 的编码顺序依次变化时，相邻两个代码之间只有一位发生变化。这样在代码转换的过程中就不会产生过渡“噪声”。而在普通二进制代码的转换过程中，则有时会产生过渡噪声。例如，第四行的二进制代码 0011 转换为第五行的 0100 过程中，如果最右边一位的变化比其他两位的变化慢，就会在一个极短的瞬间出现 0101 状态，这个状态将成为转换过程中出现的噪声。而在第四行的格雷码 0010 向第五行的 0110 转换过程中则不会出现过渡噪声。这种过渡噪声在有些情况下甚至会影响电路的正常工作，这时就必须采取措施加以避免。在后续章节中我们还将进一步讨论这个问题。

十进制代码中的余 3 循环码就是取 4 位格雷码中的十个代码组成的，它仍然具有格雷码的优点，即两个相邻代码之间仅有一位不同。

四、美国信息交换标准代码(ASCII)

美国信息交换标准代码(American Standard Codefor Information Interchange，简称 ASCII 码) 是由美国国家标准化协会(ANSII) 制定的一种信息代码，广泛地用于计算机和通信领域中。ASCII 码已经由国际标准化组织(ISO) 认定为国际通用的标准代码

ASCII 码是一组 7 位二进制代码($b_7b_6b_5b_4b_3b_2b_1$)，共 128 个，其中包括表示 0～9 的十个代码，表示大、小写英文字母的 52 个代码，32 个表示各种符号的代码以及 34 个控制码。表 1.1.3 是 ASCII 码的编码表，每个控制码在计算机操作中的含义列于表 1.1.4 中。

电子工业出版社版权所有 盗版必究

表 1.1.3 美国信息交换标准代码(ASCII 码)

$b_4b_3b_2b_1$	$b_7b_6b_5$							
	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	、	p
0001	SOH	DCl	!	1	A	Q	a	q
0010	STX	DC2	‘‘	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	‘	7	C	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	—	=	M	]	m	}
1110	S0	RS	.	>	N	∧	n	~
1111	SI	US	/	?	O	—	o	DEL

表 1.1.4 ASCII 码中控制码的含义

代码	含义		代码	含义	
NUL	Null	空白,无效	DC1	Device control 1	设备控制 1
SOH	Start of heading	标题开始	DC2	Device control 2	设备控制 2
STX	Start of text	正文开始	DC3	Device control 3	设备控制 3
KTX	End of text	文本结束	DC4	Device control 4	设备控制 4
EOT	End of transmission	传输结束	NAK	Negative acknowledge	否定
ENQ	Enquiry	询问	SYN	Synchronous idle	空转同步
ACK	Acknowledge	承认	ETB	End of transmission block	信息块传输结束
BEI	Bell	报警	CAN	Cancel	作废
BS	Backspace	退格	EM	End of medium	媒体用毕
HT	Horizontal tab	横向制表	SUB	Substitute	代替,置换
LF	Line feed	换行	ESC	Escape	扩展
VT	Vertical tab	垂直制表	FS	File separator	文件分隔
FF	Form feed	换页	CS	Group separator	组分隔
CR	Carriage return	回车	RS	Record separator	记录分隔
SO	Shift out	移出	US	Unit separator	单元分隔
SI	Shift in	移入	SP	Space	空格
DLE	Date Link escape	数据通信换码	DEL	Delete	删除

1.2 逻辑代数

逻辑代数又称布尔代数或开关代数,它是研究开关理论及分析、设计数字逻辑的数学基础。本节讲述逻辑代数的基本概念、基本公式、运算规律。

1.2.1 逻辑变量与逻辑函数概念

在逻辑代数中的变量称为逻辑变量,用字母 A、B、C…… 表示。逻辑变量只能有两种可能的取值:“1” 或 “0”。这里的 “1” 和 “0” 并不表示数量的大小,而是表示完全对立的两种状态。譬如是与非,真与假,有与无,通与断,三极管放大器饱和导通与截止等。“1” 表示条件具备或

电子工业出版社版权所有 盗版必究

事情发生；“0”表示条件不具备或事情不发生。反之亦然。

例如，在图 1.2.1 所示的电路中，指示灯是否点亮取决于开关是否接通。如果定义：F = 1 表示灯亮，F = 0 表示灯灭；A = 1 表示开关接通，A = 0 表示开关断开。

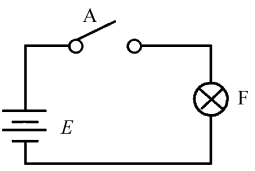


图 1.2.1　指示灯开关电路

那么，F 是 A 的函数，逻辑函数表达式为 F = f(A)。F 和 A 都称为逻辑变量，其中 A 称为输入逻辑变量(简称逻辑变量)，F 称为输出逻辑变量(简称逻辑函数)。如果逻辑函数是多变量的函数，即 F = f(A,B,C,…)，那么逻辑函数的表达式就比较复杂，逻辑函数表达式由逻辑变量 A、B、C…… 和算子“•”(与)、“+”(或)、“—”(非) 及括号、等号等组成。

例如：$F = A \cdot (B + \overline{C})$

其中，$\overline{C}$ 表示逻辑变量 C 的反变量，其他不加上画线为逻辑原变量。

1.2.2　三种基本逻辑及其运算

一、“与”逻辑(又称逻辑乘)

定义：只有当决定某一事件的条件全部具备时，这一事件才会发生。我们称这种因果关系为与逻辑关系。

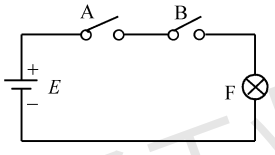


图 1.2.2　“与”逻辑示例

如图 1.2.2 所示。用 A、B 两个串联开关去控制一个电灯 F，两个开关的状态组合共有 4 种，这 4 种不同的组合与灯亮、灯灭之间的关系如表 1.2.1 所示。由表可见，只有当开关 A 与 B 同时闭合时，灯 F 才亮；否则灯灭。

现用“1”来表示开关“闭合”及灯亮；用“0”来表示开关“断开”及灯灭。那么表 1.2.1 所示“与”逻辑关系可表示成表 1.2.2 所示的真值表形式。所谓真值表是指把逻辑变量的所有可能的组合及其对应的结果列成表格形式，此表便称为真值表。

表 1.2.1　“与”电路状态表

开关 A 的状态	开关 B 的状态	灯 F 的状态
断开	断开	不亮
断开	闭合	不亮
闭合	断开	不亮
闭合	闭合	亮

表 1.2.2　“与”电路真值表

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

上述这种“与”逻辑关系可表示成如下逻辑函数式：

$$F = A \cdot B \tag{1.2.1}$$

式中“•”一般可以省略，其中 A、B 为输入逻辑变量(自变量)，F 为输出逻辑变量(因变量)。

由真值表可知“与”逻辑运算的运算规律是

$$0 \cdot 0 = 0$$

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

$$1 \cdot 1 = 1$$

这组运算规律是从逻辑推理中得出的，故是逻辑代数的公理。

还可以将上述公理写成如下一般形式：

自等律　　$A \cdot 1 = A$　　(1.2.2)

电子工业出版社版权所有 盗版必究

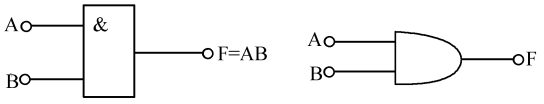
0-1律　　$A \cdot 0 = 0$　　(1.2.3)

重叠律　　$A \cdot A = A$　　(1.2.4)

二输入与门逻辑电路的逻辑符号如图1.2.3所示。

二、“或”逻辑(又称逻辑加)

定义：只要在决定某一事件的各种条件中，有一个或几个条件具备时，这一事件就会发生。我们称这种关系为“或”逻辑关系。“或”逻辑运算简称“或”运算，又称逻辑加。



(a) 国家标准符号　　(b) 国际标准符号

图1.2.3　二输入与门逻辑电路符号

如图1.2.4所示，用A、B两个并联开关去控制一个电灯F，两个开关的状态组合共有4种，这4种不同的组合与灯亮、灯灭之间的关系如表1.2.3所示。也可用真值表1.2.4来描述。由表1.2.3可见，只要开关A和B有一个或一个以上开关闭合时，灯F就会亮。

上述这种“或”逻辑关系可表示成逻辑函数式

$$F = A + B \qquad (1.2.5)$$

由真值表1.2.4可知“或”运算的运算规律为

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

这也是逻辑代数的一组公理。还可以将上述公理写成如下一般形式：

自等律　　$A + 0 = A$　　(1.2.6)

0-1律　　$A + 1 = 1$　　(1.2.7)

重叠律　　$A + A = A$　　(1.2.8)

表1.2.3　“或”逻辑状态表

开关A的状态	开关B的状态	灯F的状态
断开	断开	不亮
断开	闭合	亮
闭合	断开	亮
闭合	闭合	亮

表1.2.4　“或”逻辑真值表

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

“或”逻辑电路的逻辑符号如图1.2.5所示。

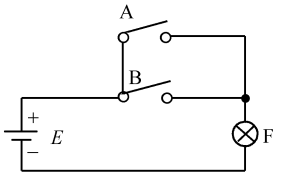
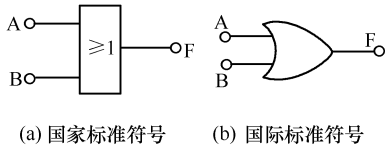


图1.2.4　“或”逻辑示例



(a) 国家标准符号　　(b) 国际标准符号

图1.2.5　二输入“或”门逻辑符号

三、“非”逻辑

“非”逻辑运算简称“非”运算，又称逻辑“非”，或逻辑否定。其意义可用图1.2.6所示的开关电路来描述。当开关A断开时，灯F亮；当开关A闭合时，则灯F被短路而灭。这里灯亮这个事件与开关A闭合这个条件之间的逻辑关系，其真值表如表1.2.5所示。

电子工业出版社版权所有
盗版必究

其逻辑表达式为

$$F = \overline{A} \tag{1.2.9}$$

由真值表 1.2.5 可知，“非”运算规律为

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

“非”逻辑也可写成一般形式：

还原律(非非律)　$\overline{\overline{A}} = A$　(1.2.10)

互补律

$$\begin{cases} A + \overline{A} = 1 \\ A \cdot \overline{A} = 0 \end{cases} \tag{1.2.11}$$

“非”门逻辑符号如图 1.2.7 所示。

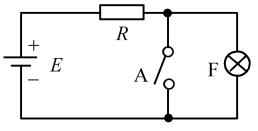
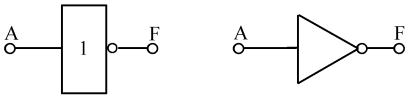


图 1.2.6　“非”逻辑示例

表 1.2.5　“非”逻辑真值表

A	F
0	1
1	0



(a) 国家标准符号　(b) 国际标准符号

图 1.2.7　“非”门逻辑符号

1.2.3　复合逻辑及其运算

实际碰到的逻辑问题往往要比简单的与、或、非复杂得多，但它们可用与、或、非的不同组合来实现。常见的复合逻辑有：与非、或非、与或非、异或及同或等。

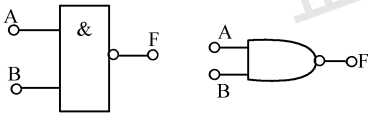
一、“与非”逻辑

“与非”逻辑实际上是由“与”逻辑和“非”逻辑组合而成的。

其函数表达式为

$$F = \overline{A \cdot B} \tag{1.2.12}$$

“与非”门逻辑符号如图 1.2.8 所示。真值表如表 1.2.6 所示。



(a) 国家标准符号　(b) 国际标准符号

图 1.2.8　与非门逻辑符号

表 1.2.6　“与非”逻辑真值表

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

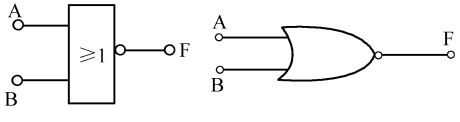
二、“或非”逻辑

“或非”逻辑是由“或”逻辑和“非”逻辑组合而成的。

其函数表达式为

$$F = \overline{A + B} \tag{1.2.13}$$

“或非”门逻辑符号如图 1.2.9 所示。真值表如表 1.2.7 所示。



(a) 国家标准符号　(b) 国际标准符号

图 1.2.9　二输入“或非”门符号

表 1.2.7　“或非”逻辑真值表

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

电子工业出版社版权所有 盗版必究

三、“与或非”逻辑

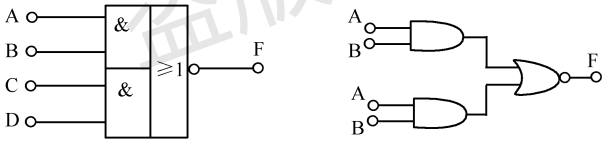
“与或非”逻辑是由“与”逻辑、“或”逻辑和“非”逻辑组合而成的。其函数表达式为

$$F = \overline{AB + CD} \tag{1.2.14}$$

真值表如表 1.2.8 所示。逻辑符号如图 1.2.10 所示。

表 1.2.8 “与或非”逻辑真值表

A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0



(a) 国家标准符号　　(b) 国际标准符号

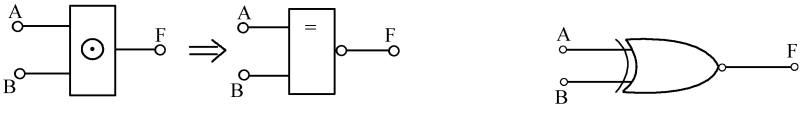
图 1.2.10 “与或非”门符号

四、“同或”逻辑

“同或”逻辑定义：若两个输入逻辑变量相同时，其输出为 1；相异时，输出为 0，则输出与输入的逻辑关系为“同或”逻辑关系。其函数表达式为：

$$F = A \odot B = AB + \overline{A}\,\overline{B} \tag{1.2.15}$$

真值表如表 1.2.9 所示，逻辑符号如图 1.2.11 所示。



(a) 国家标准符号　　(b) 国际标准符号

图 1.2.11 “同或”门逻辑符号

电子工业出版社版权所有 盗版必究

由真值表 1.2.9 可知“同或”运算的规律为

$$0\odot 0=1$$
$$0\odot 1=1\odot 0=0$$
$$1\odot 1=1$$

表 1.2.9　“同或”逻辑真值表

A	B	F = A⊙B
0	0	1
0	1	0
1	0	0
1	1	1

据此不难证明“同或”逻辑的下列公式和性质：

(1) $A\odot 0=\overline{A}$　　(1.2.16)

(2) $A\odot 1=A$　　(1.2.17)

(3) 在“同或”运算中，等式一边或两边的变量位置可以互换。设 $A\odot B=C$，则 $B\odot A=C$，或 $B\odot C=A$，或 $C\odot A=B$ 等。

(4) $$A\odot(B\odot C)=(A\odot B)\odot C \tag{1.2.18}$$

(5) $$A+(B\odot C)=(A+B)\odot(A+C) \tag{1.2.19}$$

(6) $A\odot A=1$，$A\odot\overline{A}=0$，据此可以推得

$$\underbrace{A\odot A\odot\cdots\odot A}_{\text{偶数个A}}=1,\underbrace{A\odot A\odot\cdots\odot A}_{\text{奇数个A}}=A \tag{1.2.20}$$

由此可知，同或运算的输出结果与变量值为 1 的个数无关；若变量值为 0 的个数为偶数，则输出为 1，若变量值为 0 的个数为奇数，则输出为 0。如 $1\odot 1\odot 0\odot 0\odot 0=0$，$1\odot 1\odot 1\odot 1\odot 0\odot 0=1$。

五、“异或”逻辑

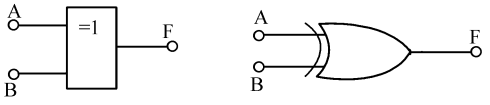
“异或”逻辑定义：若两个输入变量相异时，其输出为 1；相同时，输出为 0，则输出与输入的逻辑关系为“异或”逻辑关系。其函数表达式为

$$F=A\oplus B=A\overline{B}+\overline{A}B \tag{1.2.21}$$

真值表如表 1.2.10 所示。逻辑符号如图 1.2.12 所示。

表 1.2.10　“异或”逻辑真值表

A	B	F = A⊕B
0	0	0
0	1	1
1	0	1
1	1	0



(a) 国家标准符号　　(b) 国际标准符号

图 1.2.12　“异或”门逻辑符号

由真值表 1.2.10 可知“异或”运算的运算规律为

$$0\oplus 0=0$$
$$0\oplus 1=1\oplus 0=1$$
$$1\oplus 1=0$$

据此不难证明“异或”逻辑的下列公式和性质：

(1) $A\oplus 0=A$　　(1.2.22)

(2) $A\oplus 1=\overline{A}$　　(1.2.23)

(3) 在“异或”运算中，等式一边或两边的变量位置可互换。设 $A\oplus B=C$，则 $B\oplus A=C$ 或 $B\oplus C=A$，$C\oplus A=B$ 等。

(4) $A\oplus(B\oplus C)=(A\oplus B)\oplus C$　　(1.2.24)

电子工业出版社版权所有 盗版必究

(5) $A\cdot(B\oplus C)=(AB)\oplus(AC)$ (1.2.25)

(6) $A\oplus A=0$, $A\oplus\overline{A}=1$,据此可以推出

$$\underbrace{A\oplus A\oplus\cdots\oplus A}_{\text{偶数个A}}=0,\quad \underbrace{A\oplus A\oplus\cdots\oplus A}_{\text{奇数个A}}=A \tag{1.2.26}$$

由此可知,“异或”运算的输出结果与变量值为0的个数无关;若变量值为1的个数为奇数,则输出为1,若变量值为1的个数为偶数,则输出为0。如$1\oplus1\oplus1\oplus0\oplus0=1$,$1\oplus1\oplus1\oplus1\oplus0\oplus0=0$。

对“异”运算与“同”运算进行比较可以发现

$$A\odot B=\overline{A\oplus B}=A\oplus\overline{B}=\overline{A}\oplus B \tag{1.2.27}$$

$$A\oplus B=\overline{A\odot B}=A\odot\overline{B}=\overline{A}\odot B \tag{1.2.28}$$

1.2.4 逻辑函数的描述

描述逻辑函数的方法有:逻辑函数表达式、真值表、逻辑图、波形图和卡诺图等。

一、逻辑函数表达式

用与、或、非等逻辑运算表示逻辑变量之间关系的代数式,称逻辑函数表达式。

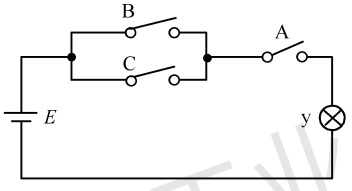


图1.2.13 举重裁判电路

任何一件具体的因果关系都可以用一个逻辑函数来描述。例如,图1.2.13是一个举重裁判电路,可以用一个逻辑函数表达式来描述它的逻辑功能。

比赛规则规定,在一名主裁判和两名副裁判中,必须有两人以上(而且必须包括主裁判)认定运动员的动作合格,试举才算成功。比赛时主裁判掌握着开关A,两名副裁判掌握着开关B和C。当运动员举起杠铃时,裁判认为动作合格了就合上开关,否则不合。显然,指示灯y的状态(亮与暗)是开关A、B、C状态(合上与断开)的函数,其逻辑函数表达式为

$$y=A\cdot(B+C) \tag{1.2.29}$$

二、真值表

将输入变量所有的取值下对应的输出值找出来,列成表格,即可得到真值表。

仍以图1.2.13的举重裁判电路为例,根据电路的工作原理不难看出,只有$A=1$,同时B、C至少有一个为1时y才等于1,于是可列出图1.2.13的真值表,如表1.2.11所示。

表1.2.11 举重裁判电路的真值表

输入			输出
A	B	C	y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

三、逻辑图

将逻辑函数中各变量之间的与、或、非等逻辑关系用图形符号表示出来,就是表示逻辑关系的逻辑图。

为了画出表示图1.2.13电路功能的逻辑图,只要用逻辑运算的符号代替式(1.2.29)中的代数运算符号便可以得到图1.2.14所示的逻辑图。

四、波形图

用波形图来描述输出与输入变量之间的逻辑关系也是行之有效的方法。仍以举重裁判电

电子工业出版社版权所有 盗版必究

路为例，其波形图如图 1.2.15 所示。

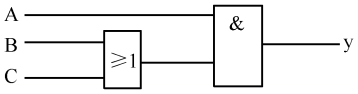


图 1.2.14　举重裁判电路的逻辑图

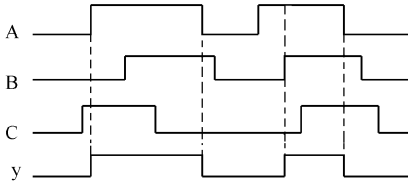


图 1.2.15　举重裁判电路的波形图

五、卡诺图

卡诺图是图形化的真值表。如果把各种输入变量取值组合下的输出函数值填入一种特殊的方格图中，即可得到逻辑函数的卡诺图，用卡诺图表示的函数的方法将在后面专门介绍。

1.2.5　逻辑代数的定律、规则及常用公式

一、逻辑函数相等的定义

假设 $F(A_1,A_2,\cdots,A_n)$ 为变量 $A_1,A_2,\cdots,A_n$ 的逻辑函数，$G(A_1,A_2,\cdots,A_n)$ 也为变量 $A_1,A_2,\cdots,A_n$ 的逻辑函数，如果对应于 $A_1,A_2,\cdots,A_n$ 的任一组状态组合，F 和 G 的值都相同，则称 F 和 G 是等值的，也就是说，F 和 G 是相等的，记作 F = G。

二、基本定律

表 1.2.12 中 0-1 律、自等律、重叠律、互补律已在前几节中证明过。交换律、结合律、分配律比较明显，与初等代数中三定律相对应。当然也可以根据逻辑函数相等的定义，利用真值表很容易证明这此定律。

表 1.2.12　基 本 定 律

定律名称	公　　式	
0-1 律	$A\cdot 0=0$	$A+1=1$
自等律	$A\cdot 1=A$	$A+0=A$
重叠律	$A\cdot A=A$	$A+A=A$
互补律	$A\cdot\overline{A}=0$	$A+\overline{A}=1$
交换律	$A\cdot B=B\cdot A$	$A+B=B+A$
结合律	$A\cdot(B\cdot C)=(A\cdot B)\cdot C$	$A+(B+C)=(A+B)+C$
分配律	$A(B+C)=AB+AC$	$A+BC=(A+B)\cdot(A+C)$
还原律	$\overline{\overline{A}}=A$	$(A^*)^*=A$
反演律	$\overline{AB}=\overline{A}+\overline{B}$	$\overline{A+B}=\overline{A}\cdot\overline{B}$
吸收律(一)	$AB+A\overline{B}=A$	$(A+B)(A+\overline{B})=A$
吸收律(二)	$A+AB=A$	$A\cdot(A+B)=A$
吸收律(三)	$A+\overline{A}B=A+B$	$A\cdot(\overline{A}+B)=AB$
吸收律(四)	$AB+\overline{A}C+BC=AB+\overline{A}C$	$(A+B)(\overline{A}+C)(B+C)=(A+B)(\overline{A}+C)$

现在用两逻辑函数相等的定义，列出它们的真值表很容易证明反演律的正确性。即$\overline{AB}=\overline{A}+\overline{B}$，$\overline{A+B}=\overline{A}\cdot\overline{B}$。

电子工业出版社版权所有 盗版必究

表 1.2.13 真 值 表

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

证明：列真值表如表 1.2.13 所示。

根据逻辑函数相等的定义得

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

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

下面用逻辑代数的公式证明几个吸收律的正确性。

证明：

吸收律(一)

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

$$(A+B)(A+\overline{B})=A(A+\overline{B})+B(A+\overline{B})=A+BA=A(1+B)=A$$

吸收律(二)

$$A+AB=A(1+B)=A$$

$$A\cdot(A+B)=A\cdot A+A\cdot B=A+AB=A(1+B)=A$$

吸收律(三)

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

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

吸收律(四)

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

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

三、逻辑代数的三个规则

1. 代入规则

在任意逻辑代数等式中，如果等式两边所有出现某一个变量(如 A)的位置都代以一个逻辑函数，则等式仍然成立。

因为任何一个逻辑函数，它和一个逻辑变量一样，只有 0 和 1 两种取值，所以代入规则是正确的。

代入规则可以用来扩展定理的应用范围，因为将已知等式某一个变量用任意一个函数代替后，就得到一个新的等式。

例如：反演律$\overline{AE}=\overline{A}+\overline{E}$，若令 E = BC，则有$\overline{ABC}=\overline{A}+\overline{BC}=\overline{A}+\overline{B}+\overline{C}$。

反复使用此规则有

推论 1：$$\overline{A\cdot B\cdot C\cdot D\cdot\cdots\cdot K}=\overline{A}+\overline{B}+\overline{C}+\overline{D}+\cdots+\overline{K}$$

推论 2：$$\overline{A+B+C+D+\cdots+K}=\overline{A}\cdot\overline{B}\cdot\overline{C}\cdot\overline{D}\cdots\overline{K}$$

2. 反演规则

对于任何一个逻辑式 F，若将其中所有的“·”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，则得到的结果就是$\overline{F}$。这个规则叫作反演规则。

反演规则为求取已知逻辑式的反逻辑式提供了方便。但是在使用反演规则时应注意遵守以下两个规则：

(1) 仍须遵守“先括号，然后乘，最后加”的运算优先次序；

(2) 不属于单个变量上的反号应保留不变。

电子工业出版社版权所有 盗版必究

【例 1.2.1】　已知 $F=\overline{A}\,\overline{B}+CD$，求 $\overline{F}$。

解：$\overline{F}=(A+B)(\overline{C}+\overline{D})$

【例 1.2.2】　已知　$F=A+\overline{B+\overline{C}\cdot\overline{D+\overline{E}}}$，求 $\overline{F}$。

解：∵　$F=A+\overline{B+\overline{C}\cdot\overline{D+\overline{E}}}=A+\overline{B+\overline{C}\cdot\overline{D}\cdot E}$

∴　$\overline{F}=\overline{A}\cdot(B+\overline{C}\,\overline{D}E)=\overline{A}\,B+\overline{A}\,\overline{C}\,\overline{D}E$

3. 对偶规则

对于任何一个逻辑表达式 F，如果将式中所有的“·”换成“+”，“+”换成“·”，“0”换成“1”，“1”换成“0”，而变量保持不变，原表达式中的运算优先顺序不变。那么就可以得到一个新的表达式，这个新的表达式称为 F 的对偶式 F^*。

【例 1.2.3】　已知 $F=\overline{A}\,\overline{B}+CD$，求 F^*。

解：$F^*=(\overline{A}+\overline{B})(C+D)$

【例 1.2.4】　已知 $F=A+\overline{B+\overline{C}\,\overline{D+\overline{E}}}$，求 F^*。

解：$F^*=A\cdot\overline{B[\overline{C}+D\,\overline{E}]}=A(\overline{B}+CD\,\overline{E})=A\,\overline{B}+ACD\,\overline{E}$

注意：①F 的对偶式 F^* 和 F 的反演式 $\overline{F}$ 是不同的，如例 1.2.3 和例 1.2.1；

② 在求 F^* 时，不需要将原变量与反变量互换。

对偶式有两个重要的性质：

性质 1：若 $F(A,B,\cdots)=G(A,B,\cdots)$，则

$$F^*=G^*$$

性质 2：$(F^*)^*=F$

利用对偶式两个重要性质，很容易证明表 1.2.12 所列基本定律，已知左边的定律，证明右边定律的正确性。反之亦然。

【例 1.2.5】　证明 $(A+B)(\overline{A}+C)(B+C)=(A+B)(\overline{A}+C)$

证明：∵　$AB+\overline{A}\,C+BC=AB+\overline{A}\,C$　　[吸收律(四)]

对上式两边取对偶得

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

1.3　逻辑函数化简

逻辑函数有三种化简方法：

(1) 公式化简法：利用逻辑代数的基本公式和规则来化简逻辑函数。

(2) 图解化简法：又称卡诺图(Karnaugh Map) 化简法。

(3) 表格法：又称 Q-M(Quine-Mc Cluskey) 化简法。这部分内容已删去，感兴趣的读者请参考相关的书籍。

1.3.1　逻辑函数的最简形式

同一个逻辑函数可以写成各种形式，但是，表达式越简单，它所表示的逻辑关系越明显，同时可用最小的电子器件来实现这个逻辑函数。因此需要通过化简的方法找出逻辑函数的最简形式。例如，下面为同一逻辑函数的两个不同的表达式：

电子工业出版社版权所有 盗版必究

$$F_1 = \overline{A}B + B + A\overline{B}, \qquad F_2 = A + B$$

显然,F_2 要比 F_1 简单得多。

在各种逻辑函数表达式中,最常用的是与或表达式。因为由它可以推出其他形式表达式,所以这里着重讨论最简与或表达式。判别与或表达式是否最简的条件是:① 乘积项(与项)最少;② 每个乘积项中变量最少。

1.3.2 逻辑函数的代数化简法

代数化简法就是利用逻辑代数的公理、定律、定理、基本公式等进行化简的方法。常用的方法有吸收法和配项法。

一、吸收法

利用吸收四定律进行化简

吸收律(一):$AB + A\overline{B} = A$, $(A+B)(A+\overline{B}) = A$

吸收律(二):$A + AB = A$, $A \cdot (A+B) = A$

吸收律(三):$A + \overline{A}B = A + B$, $A \cdot (\overline{A} + B) = AB$

吸收律(四):$AB + \overline{A}C + BC = AB + \overline{A}C$, $(A+B)(\overline{A}+C)(B+C) = (A+B)(\overline{A}+C)$

【例 1.3.1】 化简 $F = A(BC + \overline{B}\,\overline{C}) + A(B\,\overline{C} + \overline{B}C)$

解法 1:

$$\begin{aligned} F &= A(BC + \overline{B}\,\overline{C}) + A(B\,\overline{C} + \overline{B}C) \\ &= A(B \odot C) + A(B \oplus C) = A(B \odot C) + A(\overline{B \odot C}) && [\text{同或异或关系}] \\ &= A && [\text{利用吸收律(一)}] \end{aligned}$$

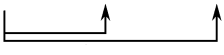
解法 2:

$$\begin{aligned} F &= A(BC + \overline{B}\,\overline{C}) + A(B\,\overline{C} + \overline{B}C) \\ &= A(BC + \overline{B}\,\overline{C} + B\,\overline{C} + \overline{B}C) && [\text{分配律}] \\ &= A[(BC + B\,\overline{C}) + (\overline{B}\,\overline{C} + \overline{B}C)] && [\text{结合律}] \\ &= A(B + \overline{B}) = A && [\text{利用吸收律(一)}] \end{aligned}$$

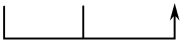
【例 1.3.2】 化简 $F = AC + A\overline{B}CD + ABC + \overline{C}D + ABD$

解:

$$F = AC + A\overline{B}CD + ABC + \overline{C}D + ABD \qquad [\text{分配律}]$$



$$= AC + \overline{C}D + ABD \qquad [\text{利用吸收律(二)}]$$



$$= AC + \overline{C}D \qquad [\text{利用吸收律(四)}]$$

【例 1.3.3】 化简 $F = A + \overline{\overline{A}\,\overline{BC}}(\overline{A} + \overline{\overline{B}\,\overline{C} + D}) + BC$

解:

$$\begin{aligned} F &= A + \overline{\overline{A}\,\overline{BC}}(\overline{A} + \overline{\overline{B}\,\overline{C} + D}) + BC \\ &= (A + BC) + (A + BC)(\overline{A} + \overline{\overline{B}\,\overline{C} + D}) && [\text{利用反演律}] \\ &= A + BC && [\text{利用吸收律(二)}] \end{aligned}$$

【例 1.3.4】 化简 $F = AB + \overline{A}C + \overline{B}C$

解:

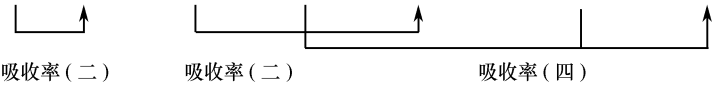
$$\begin{aligned} F &= AB + \overline{A}C + \overline{B}C = AB + (\overline{A} + \overline{B})C \\ &= AB + \overline{AB}\,C && [\text{利用反演律}] \end{aligned}$$

电子工业出版社版权所有 侵权必究

$= AB + C$　　[利用吸收律(三)]

【例 1.3.5】　化简 $F = A(A+B)(\overline{A}+C)(B+D)(\overline{A}+C+E+F)(\overline{B}+F)(D+E+F)$

解法 1： $F = A(A+B)(\overline{A}+C)(B+D)(\overline{A}+C+E+F)(\overline{B}+F)(D+E+F)$



$= A(\overline{A}+C)(B+D)(\overline{B}+F)$　　[利用吸收律(二)、(四)]

$= AC(B+D)(\overline{B}+F)$　　[利用吸收律(三)]

解法 2： 解题思路：利用公式 $F = (F^*)^*$ 即

将或与式 $\xrightarrow{\text{利用对偶规则}}$ 与或式 $\xrightarrow{\text{利用公式}}$ 化简 $\xrightarrow{\text{利用对偶规则}}$ 或与式

$F^* = A + AB + \overline{A}C + BD + \overline{A}CEF + \overline{B}F + DEF$

$= (A + AB) + (\overline{A}C + \overline{A}CEF) + (BD + \overline{B}F + DEF)$

$= A + \overline{A}C + BD + \overline{B}F$　　[利用吸收律(二)、(四)]

$= A + C + BD + \overline{B}F$　　[利用吸收律(三)]

故　　$F = (F^*)^* = AC(B+D)(\overline{B}+F)$

二、配项法

配项法步骤如下：

① 利用公式　$A + \overline{A} = 1$ 配项，将一项展为二项

② 利用公式　$AB + \overline{A}C = AB + \overline{A}C + BC$ 增加一项　} 再与其他项合并

③ 利用公式　$A = A + AB$ 或 $A = A + A$ 增加一项　}

【例 1.3.6】　化简 $F = A\overline{B} + B\overline{C} + \overline{B}C + \overline{A}B$

解法 1： $F = A\overline{B} + B\overline{C} + \overline{B}C + \overline{A}B$

$= A\overline{B} + B\overline{C} + \overline{B}C(A + \overline{A}) + \overline{A}B(C + \overline{C})$　　利用公式 $A + \overline{A} = 1$

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

$= A\overline{B} + B\overline{C} + \overline{A}C$　　[利用吸收律(一)、(二)]

解法 2： $F = A\overline{B} + B\overline{C} + \overline{B}C + \overline{A}B$

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

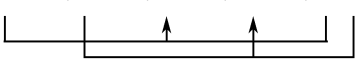
[利用公式 $A + \overline{A} = 1$，将一项展为二项]

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

$= \overline{B}C + \overline{A}B + A\overline{C}$　　[利用吸收律(一)、(二)]

解法 3： $F = A\overline{B} + B\overline{C} + \overline{B}C + \overline{A}B$

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

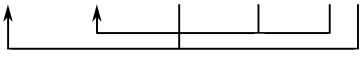


[利用公式 $\overline{B}C + \overline{A}B = \overline{B}C + \overline{A}B + \overline{A}C$ 增加一项]

$= A\overline{B} + B\overline{C} + \overline{A}C$　　[利用吸收律(四)]

解法 4： $F = A\overline{B} + B\overline{C} + \overline{B}C + \overline{A}B$

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



电子工业出版社版权所有 翻版必究

$$= \overline{B}C + \overline{A}B + A\overline{C}$$ [利用公式 $A\overline{B}+B\overline{C}=A\overline{B}+B\overline{C}+A\overline{C}$ 增加一项] [利用吸收律(四)]

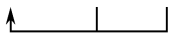
【例 1.3.7】 化简 $F = \overline{A}B\overline{C} + \overline{A}BC + ABC$

解：
$$F = \overline{A}B\overline{C} + \overline{A}BC + ABC$$
$$= (\overline{A}B\overline{C} + \overline{A}BC) + (\overline{A}BC + ABC)$$ [利用公式 $A = A + A$ 增加一项]
$$= \overline{A}B + BC$$ [利用吸收律(一)]

公式化简法，尚没有一套完整的方法，能否快速准确地化简取决于运用公式的熟练程度，下面再举几个综合例子加以说明。

【例 1.3.8】 化简式 $F = AC + \overline{B}C + B\overline{D} + C\overline{D} + A(B+\overline{C}) + \overline{A}BC\overline{D} + A\overline{B}DE$

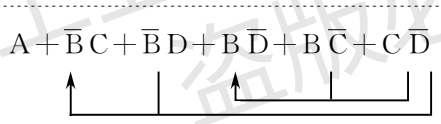
解：
$$F = AC + \overline{B}C + B\overline{D} + C\overline{D} + A(B+\overline{C}) + \overline{A}BC\overline{D} + A\overline{B}DE$$
$$= A(C + B + \overline{C} + \overline{B}DE) + (C\overline{D} + \overline{A}BC\overline{D}) + \overline{B}C + B\overline{D}$$
$$= A + C\overline{D} + \overline{B}C + B\overline{D}$$



$$= A + \overline{B}C + B\overline{D}$$

【例 1.3.9】 化简表达式 $F = AB + A\overline{C} + \overline{B}C + \overline{B}D + B\overline{D} + B\overline{C} + ADE(F+G)$

解法 1：
$$F = AB + A\overline{C} + \overline{B}C + \overline{B}D + B\overline{D} + B\overline{C} + ADE(F+G)$$
$$= A(B+\overline{C}) + \overline{B}C + \overline{B}D + B\overline{D} + B\overline{C} + ADE(F+G)$$
$$= A\overline{\overline{B}\,\overline{C}} + \overline{B}C + \overline{B}D + B\overline{D} + B\overline{C} + ADE(F+G)$$ [利用反演律]
$$= A + \overline{B}C + \overline{B}D + B\overline{D} + B\overline{C} + ADE(F+G)$$ [利用吸收律(三)]
$$= A + \overline{B}C + \overline{B}D + B\overline{D} + B\overline{C}$$ [利用吸收律(二)]



$$= A + \overline{B}C + \overline{B}D + B\overline{D} + B\overline{C} + C\overline{D}$$

[利用公式 $\overline{B}C + B\overline{D} = \overline{B}C + B\overline{D} + C\overline{D}$ 增加一项] [利用吸收律(四)]
$$= A + \overline{B}D + B\overline{C} + C\overline{D}$$

解法 2： 前几步化简同解法 1(虚线之前)
$$F = AB + A\overline{C} + \overline{B}C + \overline{B}D + B\overline{D} + B\overline{C} + ADE(F+G)$$
$$= A + \overline{B}C + \overline{B}D + B\overline{D} + B\overline{C}$$
$$= A + \overline{B}C + \overline{B}D + B\overline{D} + B\overline{C} + \overline{C}D$$

[利用公式 $\overline{B}D + B\overline{C} = \overline{B}D + B\overline{C} + \overline{C}D$ 增加一项] [利用吸收定律(四)]
$$= A + \overline{B}C + B\overline{D} + \overline{C}D$$

由例 1.3.6 和例 1.3.9 可知，逻辑函数化简的方法可以是多种多样，且最后结果不一定唯一。

1.3.3 图解化简法(卡诺图化简法)

图解化简法简称图解法。这种方法是美国工程师卡诺提出来的。所以又称卡诺图法。本节先介绍几个重要概念，然后介绍卡诺图的构成，用卡诺图表示逻辑函数，最后介绍利用卡

电子工业出版社版权所有 盗版必究

诺图化简逻辑函数。

一、几个重要概念

1. 最小项

设有 n 个变量，由它们所组成的具有 n 个变量的“与”项中，每个变量或以原变量或以反变量的形式出现一次，且仅出现一次，这个乘积项称为该组变量的最小项。

n 个变量有 2^n 个最小项。例如，4 变量 A、B、C、D，有如下 16 个最小项：$\overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$，$\overline{A}\,\overline{B}\,\overline{C}D$，$\overline{A}\,\overline{B}C\,\overline{D}$，…，ABCD。为了书写方便，把最小项记作 m_i。i 是由如下方法确定的：把乘积项中的原变量记作“1”，反变量记作“0”，把每个乘积项表示成一个二进制数，这个二进制数所对应的十进制数就是 i 的值。如 $\overline{A}\,\overline{B}CD$ 为 0011，即为 3，则可记 $\overline{A}\,\overline{B}CD = m_3$。4 变量的最小项为：

$$\overline{A}\,\overline{B}\,\overline{C}\,\overline{D} = m_0 \qquad A\,\overline{B}\,\overline{C}\,\overline{D} = m_8$$
$$\overline{A}\,\overline{B}\,\overline{C}D = m_1 \qquad A\,\overline{B}\,\overline{C}D = m_9$$
$$\overline{A}\,\overline{B}C\,\overline{D} = m_2 \qquad A\,\overline{B}C\,\overline{D} = m_{10}$$
$$\overline{A}\,\overline{B}CD = m_3 \qquad A\,\overline{B}CD = m_{11}$$
$$\overline{A}B\,\overline{C}\,\overline{D} = m_4 \qquad AB\,\overline{C}\,\overline{D} = m_{12}$$
$$\overline{A}B\,\overline{C}D = m_5 \qquad AB\,\overline{C}D = m_{13}$$
$$\overline{A}BC\,\overline{D} = m_6 \qquad ABC\,\overline{D} = m_{14}$$
$$\overline{A}BCD = m_7 \qquad ABCD = m_{15}$$

任何一个逻辑函数 F 都可以用最小项之和（亦称“积之和”形式）来表示。n 个变量应有 2^n 个最小项，它们不是包含在 F 的“与或”表达式中，便是包含在 $\overline{F}$ 的“与或”表达式中。

例如：一个三变量逻辑函数表达式为

$$F = \overline{A}B\,\overline{C} + AB\,\overline{C} + \overline{A}BC + \overline{A}\,\overline{B}\,\overline{C}$$

可变成：

$$F = m_2 + m_6 + m_3 + m_0 = \sum m^3(0,2,3,6)$$

这里“$\sum$”表示逻辑“或运算”，m^3 表示三个变量的最小项。而 $\overline{F}$ 则应包含除 m_0、m_2、m_3、m_6 之外的其余最小项：

$$\overline{F} = m_1 + m_4 + m_5 + m_7 = \sum m^3(1,4,5,7)$$

若函数不是以最小项之和的形式给出的，则可以利用公式 $A + \overline{A} = 1$ 将每个乘积项中缺少的因子补全，把它展开成最小项之和的形式。

例如，一个四变量函数 $F = ABC + \overline{A}B\,\overline{D}$，展开为最小项之和的形式：

$$\begin{aligned} F &= ABC + \overline{A}B\,\overline{D} = ABC(D + \overline{D}) + \overline{A}B\,\overline{D}(C + \overline{C}) \\ &= ABCD + ABC\,\overline{D} + \overline{A}BC\,\overline{D} + \overline{A}B\,\overline{C}\,\overline{D} = m_{15} + m_{14} + m_6 + m_4 \\ &= \sum m^4(4,6,14,15) \end{aligned}$$

一个逻辑函数最小项之和的形式是唯一的。

下面讨论最小项的性质：

① 在输入变量的任何取值下必有一个最小项，且仅有一个最小项的值为 1。

② 任意两个最小项 m_i 和 $m_j(i \neq j)$，其逻辑“与”为“0”。

例如：$m_i m_j = m_2 m_5 = (\overline{A}B\,\overline{C})(A\,\overline{B}C) = 0\ (i = 2, j = 5)$

电子工业出版社版权所有 盗版必究

③ n 个变量的全部最小项之逻辑“或”为“1”。

$$\sum_{i=0}^{2^n-1} m_i = 1$$

④ 某一个最小项不是包含在函数 F 中,就是包含在反变量 $\overline{F}$ 中。

⑤ 具有相邻性的两个最小项之和可以合并成一项,并消去一对因子。

若两个最小项只有一个因子不同,则称这两个最小项具有相邻性。

例如:$\overline{A}\,B\,\overline{C}$ 和 $AB\,\overline{C}$ 仅有 A 因子不同,则

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

这个性质是卡诺图化简逻辑函数的依据。

2. 最大项

设有 n 个变量,由它们所组成的具有 n 个变量的“或”项中,每个变量以原变量或反变量的形式出现一次,且仅出现一次,这个“或”项称为最大项。例如,两个变量的 4 个最大项为 $\overline{A}+\overline{B}$,$\overline{A}+B$,$A+\overline{B}$,$A+B$。最大项常用 M_i 来表示。i 是这样确定的:把或项中的原变量记作“0”,反变量记作“1”,形成一个二进制数,此二进制数所对应的十进制数就是 i 的值。如 4 个变量的最大项为:

$$M_0 = A+B+C+D \qquad M_8 = \overline{A}+B+C+D$$
$$M_1 = A+B+C+\overline{D} \qquad M_9 = \overline{A}+B+C+\overline{D}$$
$$M_2 = A+B+\overline{C}+D \qquad M_{10} = \overline{A}+B+\overline{C}+D$$
$$M_3 = A+B+\overline{C}+\overline{D} \qquad M_{11} = \overline{A}+B+\overline{C}+\overline{D}$$
$$M_4 = A+\overline{B}+C+D \qquad M_{12} = \overline{A}+\overline{B}+C+D$$
$$M_5 = A+\overline{B}+C+\overline{D} \qquad M_{13} = \overline{A}+\overline{B}+C+\overline{D}$$
$$M_6 = A+\overline{B}+\overline{C}+D \qquad M_{14} = \overline{A}+\overline{B}+\overline{C}+D$$
$$M_7 = A+\overline{B}+\overline{C}+\overline{D} \qquad M_{15} = \overline{A}+\overline{B}+\overline{C}+\overline{D}$$

任何一个逻辑函数 F 都可以用最大项之积(亦称“和之积”形式)来表示。下面通过实例来加以说明。

【例 1.3.10】 将 $F = \overline{A}\,B\,\overline{C} + AB\,\overline{C} + \overline{A}\,BC + \overline{A}\,\overline{B}\,\overline{C} = \sum m^3(0,2,3,6)$ 用最大项之积来表示。

解:对 F 两次求反,并利用基本公式得

$$\overline{\overline{F}} = \overline{\overline{A}\,B\,\overline{C} + AB\,\overline{C} + \overline{A}\,BC + \overline{A}\,\overline{B}\,\overline{C}} = \overline{\sum m^3(1,4,5,7)}$$
$$= \overline{\overline{A}\,\overline{B}C + A\,\overline{B}\,\overline{C} + A\,\overline{B}\,C + ABC}$$
$$= (A+B+\overline{C})\cdot(\overline{A}+B+C)\cdot(\overline{A}+B+\overline{C})\cdot(\overline{A}+\overline{B}+\overline{C})$$
$$= M_1 \cdot M_4 \cdot M_5 \cdot M_7 = \prod M^3(1,4,5,7)$$

这里“$\prod$”表示逻辑“与”运算,M^3 表示三变量的最大项。由该例可知,一个以最小项表示的逻辑函数 F 转换成以最大项表示的方法如下:先将 $\overline{F}$ 用最小项的形式表示,然后取与最小项有相同下标的最大项进行逻辑“与”,即可得 F 的最大项表示形式。n 个变量的 2^n 个最大项中的某一个,不是包含在函数 F 的“或与”表达式中,便是包含在 $\overline{F}$ 的“或与”表达式中。

电子工业出版社版权所有 盗版必究

【例 1.3.11】　已知 $F=\prod M^3(1,2,6,7)$ 求以最大项表示的 $\overline{F}$。

解：$\because F=\prod M^3(1,2,6,7)=\sum m^3(0,3,4,5)$

$\therefore \overline{F}=\overline{\sum m^3(0,3,4,5)}=\sum m^3(1,2,6,7)=\prod M^3(0,3,4,5)$

一个逻辑函数以最大项之积表示的形式是唯一的。

下面分析最大项的性质：

① 在输入变量的任何取值下必有一个最大项，且只有一个最大项的值为 0。

② 任意两个最大项 M_i 和 M_j，其逻辑“或”为“1”。例如：

已知 $M_2=A+\overline{B}+C$，$M_5=\overline{A}+B+\overline{C}$，则 $M_2+M_5=A+\overline{B}+C+\overline{A}+B+\overline{C}=1$。

③ n 个变量的全部最大项之逻辑“与”为 0。

$$\prod_{i=0}^{2^n-1} M_i=0$$

④ 某一个最大项不是包括在函数 F 中，就是包括在反函数 $\overline{F}$ 中。

⑤ 具有相邻性的两个最大项之积可以合并成一个或项，并消去一对因子。例如：

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

若两个最大项只有一个因子不同，则称这两个最大项具有相邻性。这个性质也是卡诺图化简逻辑函数的依据。

3. 最小项与最大项的关系

① 相同 i 的最小项和最大项为互补关系。即 $m_i=\overline{M_i}$。

② $\sum m_i$ 和 $\prod M_i$ 互为互补式。即 $\overline{(\sum m_i)}=\prod M_i$，或者 $\overline{(\prod M_i)}=\sum m_i$。

4. 逻辑函数的标准形式

(1) 逻辑函数的最小项之和形式。

利用基本公式　$A+\overline{A}=1$ 可以将任何一个逻辑函数化为最小项之和的标准形式。这种标准形式在逻辑函数图形化简及计算机辅助分析和设计中得到了广泛的应用。

例如：给定逻辑函数为

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

则可以化为

$$\begin{aligned}F&=AB\,\overline{C}+BC=AB\,\overline{C}+(A+\overline{A})BC=AB\,\overline{C}+ABC+\overline{A}\,BC\\&=\sum m^3(3,6,7)\end{aligned}$$

有时也写成 $\sum\limits_i m_i(i=3,6,7)$，或 $\sum m(3,6,7)$，或 $\sum(3,6,7)$。

(2) 逻辑函数的最大项之积形式。

可以证明，任何一个逻辑函数都可以化成最大项之积的标准形式。

前面已证明，任何一个逻辑函数均可以化为最小项之和的形式。同时，从最小项的性质可知全部最小项之和为 1。由此可知，若给定逻辑函数为 $F=\sum m_i$，则 $\sum m_i$ 以外的那些最小项之和必为 $\overline{F}$，即

$$\overline{F}=\sum_{k\neq i} m_k \tag{1.3.1}$$

利用反演定律可将式(1.3.1) 变换为最大项之积的形式

电子工业出版社版权所有 盗版必究

$$F=\prod_{k\neq i}\overline{m_k}=\prod_{k\neq i}M_k \tag{1.3.2}$$

这就是说,若已知逻辑函数为 $F=\sum m_i$ 时,定能将 F 化成编号为 i 以外的那些最大项的乘积。

【例 1.3.12】 试将逻辑函数 $F=AB\overline{C}+BC$ 化成最大项之积的标准形式。

解:前面已经得到了它的最小项之和的形式为

$$F=AB\overline{C}+BC=\sum m^3(3,6,7)$$

根据式(1.3.2)可得

$$\begin{aligned}F&=\prod_{k\neq i}M_k=M_0\cdot M_1\cdot M_2\cdot M_4\cdot M_5\\&=(A+B+C)(A+B+\overline{C})(A+\overline{B}+C)(\overline{A}+B+C)(\overline{A}+B+\overline{C})\end{aligned}$$

二、卡诺图的构成

用一个大方块表示 1,如图 1.3.1(a)所示。然后将大方块分为上下两部分,上下两部分没有公共部分。如果上半部分表示 $\overline{A}$,那么下半部分就表示 A。如图 1.3.1(b)所示。如果将大方块分为左右两部分,如果右半部分表示 B,那么左半部分就表示 $\overline{B}$,如图 1.3.1(c)所示。

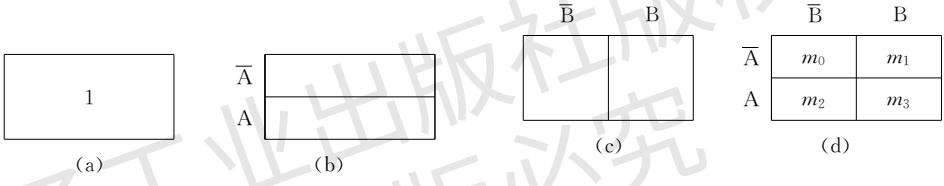


图 1.3.1 二变量卡诺图

基于上述,将大方块分成 4 个小方块,那么左上角小方块既划归于 $\overline{A}$,又划归于 $\overline{B}$,它属于 $\overline{A}$ 和 $\overline{B}$ 的公共部分,即表示为 $\overline{A}\cdot\overline{B}$;同理,右上角表示为 $\overline{A}B$,左下角表示为 $A\overline{B}$,右下角表示为 AB,如图 1.3.1(d)所示。

其实,这 4 个小方块也就表示为二变量的 4 个最小项:m_0、m_1、m_2、m_3,如图 1.3.1(d)所示。把这 4 个小方块叠加起来仍是大方块,即

$$\overline{A}\,\overline{B}+\overline{A}B+A\overline{B}+AB=1$$

我们把图 1.3.1(d)所示的图称为二变量卡诺图。

同理,不难画出三变量,四变量及五变量卡诺图,分别如图 1.3.2(a)、(b)、(c)所示。

卡诺图是由美国工程师卡诺首先提出的一种描述函数的特殊方格图,在这个方格图中,每个小方块代表逻辑函数的最小项,而且几何相邻的小方块具有逻辑相邻性,即相邻小方块所代表的最小项只有一个变量取值不同。

从图 1.3.2 可知,卡诺图有如下几个特点:

① 卡诺图中的小方块数等于最小项总数,即等于 2^n(n 为变量数)。

② 变量取值不能按二进制数的顺序排列,必须按循环码排列,这样保障了相邻最小项只有一个变量是相反的,而其余变量是相同的。两个相邻最小项相加(或),可以消去一项,且消除一个因子。这个特点是卡诺图简化逻辑函数的依据。如四变量卡诺图(如图 1.3.2(b))中的 m_5 和 m_7 属相邻项,其中 $m_5=\overline{A}B\overline{C}D$,$m_7=\overline{A}BCD$,则

$$m_5+m_7=\overline{A}B\overline{C}D+\overline{A}BCD=\overline{A}BD$$

电子工业出版社版权所有 侵权必究

③ 卡诺图是一个上下、左右闭合的图形，即不但紧挨着的方块是相邻的，而且上下、左右相对应的方块也是相邻的。

A \ BC	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

(a) 三变量卡诺图

AB \ CD	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

(b) 四变量卡诺图

AB \ CDE	000	001	011	010	110	111	101	100
00	m_0	m_1	m_3	m_2	m_6	m_7	m_5	m_4
01	m_8	m_9	m_{11}	m_{10}	m_{14}	m_{15}	m_{13}	m_{12}
11	m_{24}	m_{25}	m_{27}	m_{26}	m_{30}	m_{31}	m_{29}	m_{28}
10	m_{16}	m_{17}	m_{19}	m_{18}	m_{22}	m_{23}	m_{21}	m_{20}

(c) 五变量卡诺图

图 1.3.2　三变量、四变量、五变量卡诺图

三、用卡诺图表示逻辑函数

既然任何一个逻辑函数都能表示为若干最小项之和的形式，那么自然也就可以设法用卡诺图来表示任意一个逻辑函数。具体的方法是首先把逻辑函数化为最小项之和的形式，然后在卡诺图上与这些最小项对应的位置上填入 1，在其余的位置上填入 0，就得到了表示逻辑函数的卡诺图。也就是说，任何一个逻辑函数都等于它的卡诺图中填入 1 的那些最小项之和。

【例 1.3.13】　用卡诺图表示如下的逻辑函数。

$$F=\overline{A}\,\overline{B}\,\overline{C}D+\overline{A}\,B\,\overline{D}+ACD+A\,\overline{B}$$

解：首先将 F 化为最小项之和的形式

$$\begin{aligned}F&=\overline{A}\,\overline{B}\,\overline{C}D+\overline{A}\,B(C+\overline{C})\overline{D}+A(B+\overline{B})CD+A\,\overline{B}(C+\overline{C})(D+\overline{D})\\&=\overline{A}\,\overline{B}\,\overline{C}\,D+\overline{A}\,BC\,\overline{D}+\overline{A}\,B\,\overline{C}\,\overline{D}+A\,\overline{B}C\,D+A\,\overline{B}C\,\overline{D}+A\,\overline{B}\,\overline{C}\,D+A\,\overline{B}\,\overline{C}\,\overline{D}+ABCD\\&=m_1+m_4+m_6+m_8+m_9+m_{10}+m_{11}+m_{15}\\&=\sum m^4(1,4,6,8,9,10,11,15)\end{aligned}$$

画出四变量最小项的卡诺图，在对应于函数式中各最小项的位置上填入 1，其余位置上填入 0，就得到如图 1.3.3 所示的 F 的卡诺图。

用卡诺图表示逻辑函数的另一种方法：根据逻辑函数 F 的真值表直接填写。

【例 1.3.14】　已知逻辑函数 F 的真值表如表 1.3.1 所示。试画出 F 的卡诺图。

AB \ CD	00	01	11	10
00	0	1	0	0
01	1	0	0	1
11	0	0	1	0
10	1	1	1	1

图 1.3.3　例 1.3.13 的卡诺图

表 1.3.1　F 的真值表

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

电子工业出版社版权所有

解：画出三个变量的卡诺图并直接填写。如图1.3.4所示。

A \ BC	00	01	11	10
0	0	0	1	0
1	0	1	1	1

图1.3.4　例1.3.14的卡诺图

四、逻辑函数的卡诺图化简

利用卡诺图化简逻辑函数的方法称为卡诺图化简法或图形化简法。化简的依据是具有相邻性的最小项可以合并，并消去不同的因子。由于在卡诺图上几何位置相邻与逻辑上的相邻性是一致的，因而从卡诺图上能直观地找出那些具有相邻性的最小项并将其合并化简。

1. 合并最小项的规则

若卡诺图中有2^i个相邻项，则2^i个相邻项可合并成一项，并且消去i个变量(其中$i=1,2,\cdots$)，只剩下公共因子。

例如，在图1.3.4中m_5和m_7为相邻项，其中$m_5=A\overline{B}C$，$m_7=ABC$，$m_5+m_7=A\overline{B}C+ABC=AC$。

2. 化简步骤

用卡诺图化简逻辑函数时可按如下步骤进行：

① 将函数化简为最小项之和的形式(或列出逻辑函数真值表)；

② 画出表示该逻辑函数的卡诺图；

③ 找出可以合并的最小项(画圈)；

④ 写出最简“与或”逻辑函数表达式。

【例1.3.15】 用图形化简法对逻辑函数$F=\sum m^4(1,2,4,9,10,11,13,15)$进行化简。

解：根据化简步骤，因逻辑函数已表示成最小项之和的形式，可以省去步骤①。

② 画出逻辑函数F的卡诺图，如图1.3.5所示。

③ 画圈，将相邻“1”格圈起来，先圈单个“1”格，再圈2个“1”格，4个“1”格……。

合并最小项：

$$\sum m^4(4)=\overline{A}B\overline{C}\overline{D}$$

$$\sum m^4(1,9)=\overline{A}\overline{B}\overline{C}D+A\overline{B}\overline{C}D=\overline{B}\overline{C}D$$

$$\sum m^4(2,10)=\overline{A}\overline{B}C\overline{D}+A\overline{B}C\overline{D}=\overline{B}C\overline{D}$$

$$\sum m^4(9,11,13,15)=A\overline{B}\overline{C}D+A\overline{B}CD+AB\overline{C}D+ABCD=AD$$

④ 写出最简“与或”逻辑函数表达式

$$F=\sum m^4(4)+\sum m^4(1,9)+\sum m^4(2,10)+\sum m^4(9,11,13,15)$$
$$=\overline{A}B\overline{C}\overline{D}+\overline{B}\overline{C}D+\overline{B}C\overline{D}+AD$$

【例1.3.16】 用卡诺图化简逻辑函数

$$F(A,B,C,D)=\overline{A}\overline{B}\overline{C}+\overline{A}C\overline{D}+A\overline{B}C\overline{D}+A\overline{B}\overline{C}$$

解：① 将逻辑函数F化为最小项之和的形式，即

$$F(A,B,C,D)=\overline{A}\overline{B}\overline{C}(D+\overline{D})+\overline{A}(B+\overline{B})C\overline{D}+A\overline{B}C\overline{D}+A\overline{B}\overline{C}(D+\overline{D})$$
$$=\overline{A}\overline{B}\overline{C}D+\overline{A}\overline{B}\overline{C}\overline{D}+\overline{A}BC\overline{D}+\overline{A}\overline{B}C\overline{D}+A\overline{B}C\overline{D}+A\overline{B}\overline{C}D+A\overline{B}\overline{C}\overline{D}$$

电子工业出版社版权所有 盗版必究

$$= m_1 + m_0 + m_6 + m_2 + m_{10} + m_9 + m_8$$

$$= \sum m^4(0,1,2,6,8,9,10)$$

② 画出表示该逻辑函数的卡诺图，如图 1.3.6 所示。

③ 画圈，合并最小项。

$$\sum m^4(2,6) = \overline{A}\,\overline{B}\,C\,\overline{D} + \overline{A}\,BC\,\overline{D} = \overline{A}\,C\,\overline{D}$$

$$\sum m^4(0,1,8,9) = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}\,\overline{C}\,D + A\,\overline{B}\,\overline{C}\,\overline{D} + A\,\overline{B}\,\overline{C}\,D = \overline{B}\,\overline{C}$$

$$\sum m^4(0,2,8,10) = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}\,C\,\overline{D} + A\,\overline{B}\,\overline{C}\,\overline{D} + A\,\overline{B}\,C\,\overline{D} = \overline{B}\,\overline{D}$$

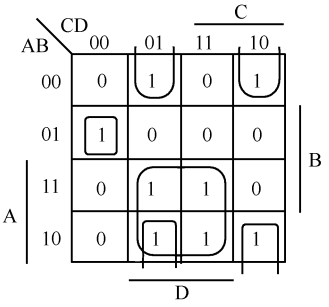


图 1.3.5　【例 1.3.15】的卡诺图

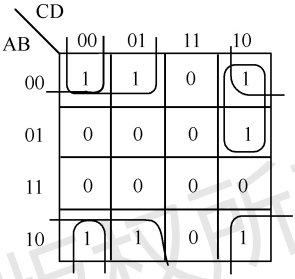


图 1.3.6　【例 1.3.16】的卡诺图

④ 写出最简“与或”逻辑函数表达式

$$F(A,B,C,D) = \overline{A}\,C\,\overline{D} + \overline{B}\,\overline{C} + \overline{B}\,\overline{D}$$

3. 画圈应注意的几个问题

用卡诺图化简，最关键是画圈这一步。画圈应注意如下几个问题：

① “1”格允许被一个以上的圈所包围，这是因为 $A + A = A$。

② “1”格不能漏画，否则简化后的逻辑表达式与原式不相等。

③ 圈的个数要尽量少，因为一个圈与一个“与”项相对应，圈数越少，表达式中的“与”项就越少。

④ 圈的面积越大越好，但必为 2^i 个方块。因为圈面越大，消去的变量就越多。

⑤ 每个圈至少包含一个新的“1”格，否则这个圈是多余的。

为了便于记忆，用一句话概括，即“可以重画，不能漏画，圈数要少，圈面要大，每圈必有一个新‘1’格”。

图 1.3.7 给出了几种不正确的画圈法与正确画圈法的比较。

1.3.4　具有无关项的逻辑函数及其化简

一、约束项、任意项和逻辑函数式中的无关项

在分析某些具体的逻辑函数时，经常会遇到这样一种情况，即输入变量的取值不是任意的。对输入变量取值所加的限制称为约束。同时，把这一组变量称为具有约束的一组变量。

例如，有三个逻辑变量 A、B、C，它们分别表示一台电动机的正转、反转和停止的命令，A＝1 表示正转，B＝1 表示反转，C＝1 表示停止。因为电动机任何时候只能执行其中的一个命令，所以不允许两个以上的变量同时为 1。ABC 的取值只可能是 001、010、100 当中的某一种，而不能是 000、011、101、110、111 中的任何一种。因此，A、B、C 是一组具有约束的变量。

电子工业出版社版权所有 盗版必究

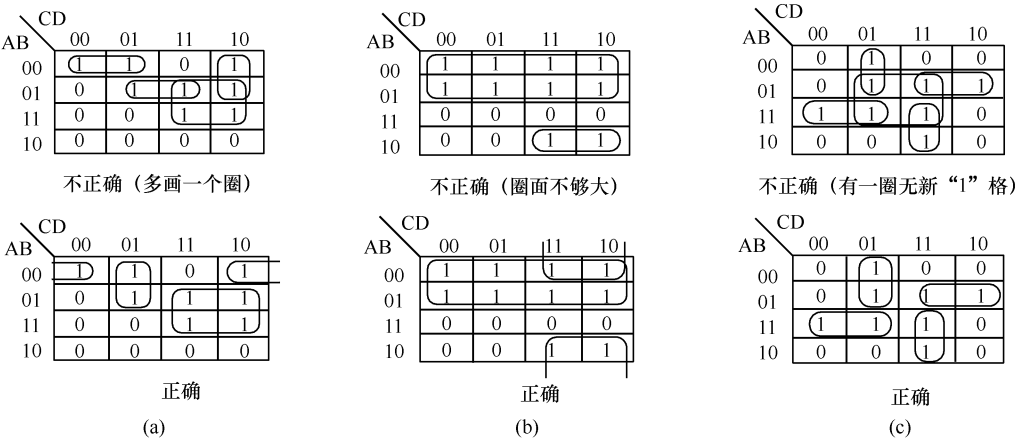


图 1.3.7　几种不正确的画圈法与正确画圈法的比较

通常用约束条件来描述约束的具体内容。显然，用上面的这样一段文字叙述约束条件是很不方便的，最好能用简单、明了的逻辑语言表述约束条件。

由于每一组输入变量的取值都使一个、而且仅有一个最小项的值为1，所以当限制某些输入变量的取值不能出现时，可以用它们对应的最小项恒等于0来表示。这样，上面例子中的约束条件可以表示为

$$\begin{cases} \overline{A}\,\overline{B}\,\overline{C}=0 \\ \overline{A}\,B\,C=0 \\ A\,\overline{B}\,C=0 \\ A\,B\,\overline{C}=0 \\ A\,B\,C=0 \end{cases}$$

或写成

$$\overline{A}\,\overline{B}\,\overline{C}+\overline{A}\,BC+A\,\overline{B}\,C+AB\,\overline{C}+ABC=0$$

同时，把这些恒等于0的最小项叫作约束项。

有时还会遇到另外一种情况，就是在输入变量的某些取值下函数值是1还是0皆可，并不影响电路的功能。在这些变量取值下，其值等于1的那些最小项称为任意项。

在存在约束项的情况下，由于约束项的值始终等于0，所以既可以把约束项写进逻辑函数式中，也可以把约束项从函数式中删掉，而不影响函数值。同样，既可以把任意项写入函数式中，也可以不写进去，因为输入变量的取值使这些任意项为1时，函数值是1还是0无所谓。

因此，又把约束项和任意项统称为逻辑函数式中的无关项。这里所说的无关是指是否把这些最小项写入逻辑函数式无关紧要，可以写入也可以删除。

1.3.3节中曾经讲到，在用卡诺图表示逻辑函数时，首先将函数化为最小项之和的形式，然后在卡诺图中将这些最小项对应的位置上填入1，其他位置上填入0。既然可以认为无关项包含于函数式中，也可以认为不包含在函数式中，那么在卡诺图中对应的位置上就可以填入1，也可以填入0。为此，在卡诺图中用×(或Ø)表示无关项。在化简逻辑函数时既可以认为它是1，也可以认为它是0。

二、无关项在逻辑函数化简中的应用

化简具有无关项的逻辑函数时，如果能合理利用这些无关项，一般都可得到更加简单的化简结果。

电子工业出版社版权所有 盗版必究

为达到此目的，加入的无关项应与函数式中尽可能多的最小项（包括原有的最小项和已写入的无关项）具有逻辑相邻性。

合并最小项时，究竟把卡诺图上的“×”作为 1（即认为函数式中包含了这个最小项）还是作为 0（即认为函数式中不包含这个最小项）对待，应以得到的相邻最小项矩形组合最大、而且矩形组合数目最少为原则。

【例 1.3.17】 化简如下具有约束的逻辑函数

$$Y=\overline{A}\,\overline{B}\,\overline{C}\,D+\overline{A}\,BCD+A\,\overline{B}\,\overline{C}\,\overline{D}$$

给定约束条件为

$$\overline{A}\,\overline{B}\,CD+\overline{A}\,B\,\overline{C}\,D+AB\,\overline{C}\,\overline{D}+A\,\overline{B}\,\overline{C}\,D+ABCD+ABC\,\overline{D}+A\,\overline{B}\,C\,\overline{D}=0$$

解：如果不利用约束项，则 Y 已无可化简。但适当地加进一些约束项以后，可以得到

$$\begin{aligned}Y&=(\overline{A}\,\overline{B}\,\overline{C}D+\underbrace{\overline{A}\,\overline{B}\,CD}_{\text{约束项}})+(\overline{A}\,BCD+\underbrace{\overline{A}\,B\,\overline{C}\,D}_{\text{约束项}})+\\&\quad(A\,\overline{B}\,\overline{C}\,\overline{D}+\underbrace{AB\,\overline{C}\,\overline{D}}_{\text{约束项}})+(\underbrace{ABC\,\overline{D}}_{\text{约束项}}+\underbrace{A\,\overline{B}\,C\,\overline{D}}_{\text{约束项}})\\&=(\overline{A}\,\overline{B}\,D+\overline{A}\,BD)+(A\,\overline{C}\,\overline{D}+AC\,\overline{D})\\&=\overline{A}\,D+A\,\overline{D}=A\oplus D\end{aligned}$$

可见，利用了约束项以后，使逻辑函数得以进一步化简。但是，在确定该写哪些约束项时尚不够直观。

如果改用卡诺图化简法，则只要将表示 Y 的卡诺图画出，就能从图上直观判断对这些约束项应如何取舍。

图 1.3.8 是例 1.3.17 题的逻辑函数的卡诺图。从图上不难看出，为了得到最大的相邻最小项的矩形组合，应取约束项 m_3、m_5 为 1，与 m_1、m_7 组成一个矩形组。同时取约束项 m_{10}、m_{12}、m_{14} 为 1，与 m_8 组成一个矩形组。将两组相邻的最小项合并后得到的化简结果与上面推演的结果相同。卡诺图中没有被圈进去的约束项（m_9 和 m_{15}）是当做 0 对待的。

【例 1.3.18】 试化简逻辑函数

$$Y=\overline{A}\,C\,\overline{D}+\overline{A}\,B\,\overline{C}\,\overline{D}+A\,\overline{B}\,\overline{C}\,\overline{D}$$

已知约束条件为

$$A\,\overline{B}\,C\,\overline{D}+A\,\overline{B}\,C\,D+AB\,\overline{C}\,\overline{D}+AB\,\overline{C}\,D+ABC\,\overline{D}+ABCD=0$$

解：画出函数 Y 的卡诺图，如图 1.3.9 所示。

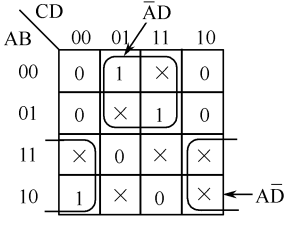


图 1.3.8　例 1.3.17 的卡诺图

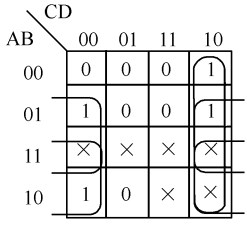


图 1.3.9　例 1.3.18 的卡诺图

由图 1.3.9 可见，若认为其中的约束项 m_{10}、m_{12}、m_{14} 为 1，而约束项 m_{11}、m_{13}、m_{15} 为 0，则可将 m_4、m_6、m_{12} 和 m_{14} 合并为 $B\,\overline{D}$，将 m_8、m_{10}、m_{12} 和 m_{14} 合并为 $A\,\overline{D}$，将 m_2、m_6、m_{10} 和 m_{14} 合并为 $C\,\overline{D}$，于是得到

$$Y=B\,\overline{D}+A\,\overline{D}+C\,\overline{D}$$

电子工业出版社版权所有 盗版必究

本章小结

本章所讲的主要内容是数制与编码,逻辑代数的公式、定律和规则,逻辑函数的表示方法和逻辑函数化简方法这几部分。

(1) 数制是人们对数量计数的一种统计规则。任何一种进位计数包含基数和位权两个基本因素。

基数为R的数制为R进制,进位规则:"逢R进1"。有0,1,…,$R-1$个数码(数符)。按权展开为$(N)_R=\sum_{i=-m}^{n-1}a_iR^i$($R$取$\geqslant 2$的正整数)。

R进制转为十进制只要按权展开($(N)_R=\sum_{i=-m}^{n-1}a_iR^i$)就很容易得到。

十进制转为R进制可分为整数和小数部分分别考虑,整数部分按除以R取余法,逆序排列;小数部分按乘以R取整法,顺序排列。

(2) 编码就是用二进制码来表示给定的信息符号。信息符号可以是十进制数符、字符、运算符等。

在数字系统中,目前常用的有原码、反码和补码。带小数点的数的编码有定点表示法和浮点表示法两种。

十进制的二进制编码可分为两大类:一类是有权码,常用的有8421BCD、2421BCD、5121BCD、631-1BCD等,这些码可以按权展开为所表示的十进制数;另一类是无权码,如格雷码。这种编码是一种可靠性编码。

(3) "与"、"或"、"非"是属于基本逻辑,由三个基本逻辑可以组合成"与非"、"或非"、"与或非"、"同或"和"异或"逻辑。

(4) 逻辑函数反映的是实际逻辑问题中输入逻辑变量与输出逻辑变量之间的因果关系。可用逻辑函数表达式、真值表、逻辑图、卡诺图、波形图、文字描述和高级语言描述等。

(5) 表1.2.12列出了逻辑代数的基本定律。这些定律完全可以根据函数相等的定义和真值表加以证明。同时还介绍了逻辑代数三个规则。

(6) 逻辑函数的化简是本章的重点。本章介绍两种化简方法。公式化简归纳为两种方法,即吸收法和配项法。吸收法是利用4个吸收定律(含8个公式)进行化简逻辑函数;配项法是利用公式$A+\overline{A}=1$,或$AB+\overline{A}C=AB+\overline{A}C+BC$,或者$A=A+AB$增加多余项,然后再与其他项合并(配项化简)。

卡诺图化简方法是简单、直观、而且有一定化简步骤可循。初学者容易掌握,而且化简过程中也易于避免差错。然而在逻辑变量超过5个以上时,将失去简单、直观的优点,因而也就没有太大的实用意义了。

习 题 一

1.1 填空题:

(1) 数制是人们对数量计数的一种统计规则。任何一种进位计数包含_____和_____两个基本因素。

(2) 十进制数转换为R进制数可分为整数和小数部分分别考虑,整数部分按_____,小数部分按_____。

电子工业出版社版权所有 盗版必究

(3) $(0011)_{631\text{-}1BCD}=(\quad\quad)_{10}$

(4) 编码就是用二进制码来表示给定的_____。

(5) A⊕1=_____。

(6) 已知 $F=A+\overline{B+\overline{C}\cdot\overline{D+\overline{E}}}$,则 $\overline{F}$=_____。

(7) 已知 $F=A+\overline{B+\overline{C}\cdot\overline{D+\overline{E}}}$,则 F^* =_____。

(8) n 个变量有_____个最小项。

1.2　将下列十六进制数转换为等值的二进制数和等值的十进制数。

(1) $(8C)_{16}$　　(2) $(3D.BE)_{16}$　　(3) $(8F.FF)_{16}$　　(4) $(10.00)_{16}$

1.3　将下列十进制数转换成等效的二进制数和等值的十六进制数。要求二进制数保留小数点以后 4 位有效数字。

(1) $(17)_{10}$　　(2) $(127)_{10}$　　(3) $(0.39)_{10}$　　(4) $(25.7)_{10}$

1.4　写出下列二进制数的原码和补码。

(1) $(+1011)_2$　　(2) $(+00110)_2$　　(3) $(-1101)_2$　　(4) $(-00101)_2$

1.5　试总结并解释

(1) 从真值表写逻辑函数式的方法;

(2) 从函数式列真值表的方法;

(3) 从逻辑图写逻辑函数式的方法;

(4) 从逻辑函数式画逻辑图的方法。

1.6　已知逻辑函数的真值表如表 P1.1(a)、(b)所示,试写出对应的逻辑函数式。

表 P1.1　(a)

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

表 P1.1　(b)

M	N	P	Q	Z
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

1.7　试用列真值表的方法证明下列异或运算公式。

(1) A⊕0=A

(2) $A\oplus 1=\overline{A}$

(3) A⊕A=0

(4) $A\oplus\overline{A}=1$

(5) (A⊕B)⊕C=A⊕(B⊕C)

(6) A(B⊕C)=AB⊕AC

(7) $A\oplus\overline{B}=\overline{A\oplus B}=A\oplus B\oplus 1$

1.8　用逻辑代数的基本公式和常用公式将下列逻辑函数化简为最简与或形式。

(1) $Y=A\,\overline{B}+B+\overline{A}\,B$

(2) $Y=A\,\overline{B}\,C+\overline{A}+B+\overline{C}$

(5) $Y=A\,\overline{B}(\overline{A}\,CD+\overline{AD+\overline{B}\,C})(\overline{A}+B)$

(3) $Y=\overline{\overline{A}\,BC+\overline{A}\,\overline{B}}$

(4) $Y=A\,\overline{B}\,CD+ABD+A\,\overline{C}\,D$

电子工业出版社版权所有
盗版必究

(6) $Y=AC(\overline{C}D+\overline{A}B)+BC(\overline{\overline{B}+AD}+CE)$

(7) $Y=A\overline{C}+ABC+AC\overline{D}+CD$

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

(9) $Y=B\overline{C}+AB\overline{C}E+\overline{B}(\overline{\overline{A}\overline{D}+AD})+B(A\overline{D}+\overline{A}D)$

(10) $Y=AC+A\overline{C}D+A\overline{B}\overline{E}F+B(D\oplus E)+B\overline{C}D\overline{E}+B\overline{C}\overline{D}E+AB\overline{E}F$

1.9 写出图 P1.1 中各逻辑图的逻辑函数式,并化简为最简与或形式。

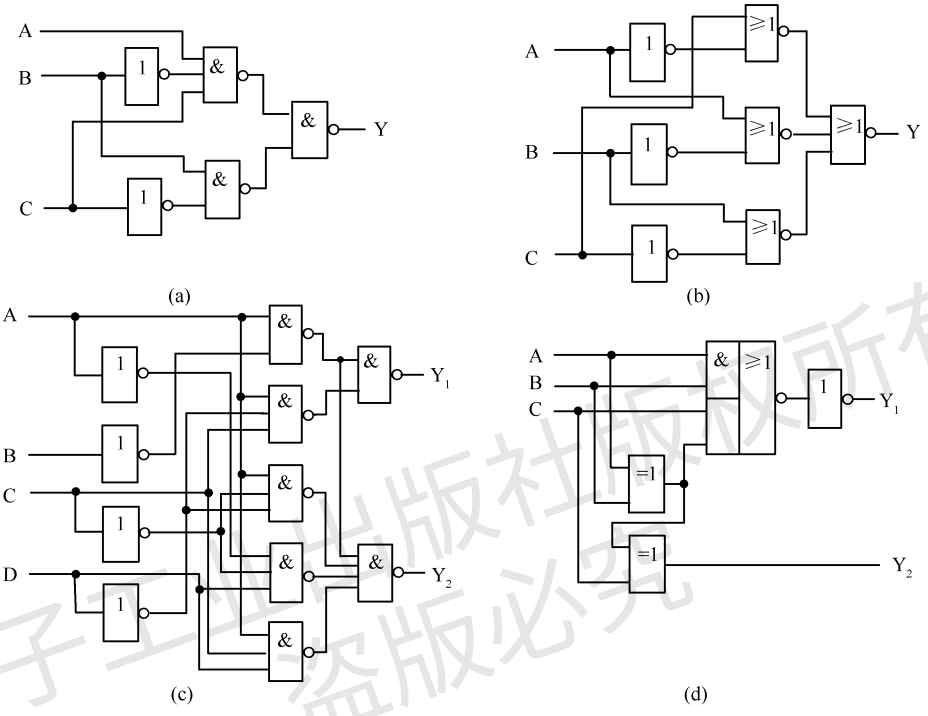


图 P1.1

1.10 求下列函数的反函数并化简为最简与或形式。

(1) $Y=AB+C$

(2) $Y=(A+BC)\overline{C}D$

(3) $Y=\overline{(A+\overline{B})(\overline{A}+C)}AC+BC$

(4) $Y=\overline{\overline{A}\overline{B}C+\overline{C}D}(AC+BD)$

(5) $Y=A\overline{D}+\overline{A}\overline{C}+\overline{B}\overline{C}D+C$

(6) $Y=\overline{E}\overline{F}\overline{G}+\overline{E}\overline{F}G+\overline{E}F\overline{G}+\overline{E}FG+E\overline{F}\overline{G}+E\overline{F}G+EF\overline{G}+EFG$

1.11 将下列各函数式化为最小项之和的形式。

(1) $Y=\overline{A}BC+AC+\overline{B}C$

(2) $Y=A\overline{B}\overline{C}D+BCD+\overline{A}D$

(3) $Y=A+B+CD$

(4) $Y=AB+\overline{B\overline{C}(\overline{C}+\overline{D})}$

(5) $Y=L\overline{M}+M\overline{N}+N\overline{L}$

1.12 将下列各式化为最大项之积的形式。

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

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

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

(4) $Y=BC\overline{D}+C+\overline{A}D$

(5) $Y(A,B,C)=\sum(m_1,m_2,m_4,m_6,m_7)$

1.13 用卡诺图化简法将下列函数化简为最简与或形式。

(1) $Y=ABC+ABD+\overline{C}\overline{D}+A\overline{B}C+\overline{A}C\overline{D}+A\overline{C}D$

(2) $Y=A\overline{B}+\overline{A}C+BC+\overline{C}D$

(3) $Y=\overline{A}\overline{B}+B\overline{C}+\overline{A}+\overline{B}+ABC$

(4) $Y=\overline{A}\overline{B}+AC+\overline{B}C$

(5) $Y=A\overline{B}\overline{C}+\overline{A}\overline{B}+\overline{A}D+C+BD$

电子工业出版社版权所有 盗版必究

(6) $Y(A,B,C)=\sum(m_0,m_1,m_2,m_5,m_6,m_7)$

(7) $Y(A,B,C)=\sum(m_1,m_3,m_5,m_7)$

(8) $Y(A,B,C,D)=\sum(m_0,m_1,m_2,m_3,m_4,m_6,m_8,m_9,m_{10},m_{11},m_{14})$

(9) $Y(A,B,C,D)=\sum(m_0,m_1,m_2,m_5,m_8,m_9,m_{10},m_{12},m_{14})$

(10) $Y(A,B,C)=\sum(m_1,m_4,m_7)$

1.14　化简下列逻辑函数(方法不限)。

(1) $Y=A\overline{B}+\overline{A}C+\overline{C}\,\overline{D}+D$

(2) $Y=\overline{A}(C\,\overline{D}+\overline{C}D)+B\,\overline{C}D+A\,\overline{C}D+\overline{A}C\,\overline{D}$

(3) $Y=\overline{(\overline{A}+\overline{B})D}+(\overline{A}\,\overline{B}+BD)\overline{C}+\overline{A}\,\overline{C}BD+\overline{D}$

(4) $Y=A\,\overline{B}D+\overline{A}\,\overline{B}\,\overline{C}D+\overline{B}CD+\overline{(A\,\overline{B}+C)}(B+D)$

(5) $Y=\overline{A\,\overline{B}\,\overline{C}D+A\,\overline{C}DE+\overline{B}D\,\overline{E}+A\,\overline{C}\,\overline{D}E}$

1.15　证明下列逻辑恒等式(方法不限)。

(1) $A\,\overline{B}+B+\overline{A}B=A+B$

(2) $(A+\overline{C})(B+D)(B+\overline{D})=AB+B\,\overline{C}$

(3) $\overline{\overline{(A+B+\overline{C})}\,\overline{C}D}+(B+\overline{C})(A\,\overline{B}D+\overline{B}\,\overline{C})=1$

(4) $\overline{A}\,\overline{B}\,\overline{C}\,\overline{D}+\overline{A}B\,\overline{C}D+ABCD+A\,\overline{B}C\,\overline{D}=\overline{A\,\overline{C}+\overline{A}C+B\,\overline{D}+\overline{B}D}$

(5) $\overline{A}(C\oplus D)+B\,\overline{C}D+AC\,\overline{D}+A\,\overline{B}\,\overline{C}D=C\oplus D$

1.16　试画出用与非门和反相器实现下列函数的逻辑图。

(1) $Y=AB+BC+AC$　　(2) $Y=(\overline{A}+B)(A+\overline{B})C+\overline{BC}$

(3) $Y=\overline{AB\,\overline{C}+A\,\overline{B}C+\overline{A}BC}$　　(4) $Y=A\,\overline{BC}+\overline{(A\,\overline{B}+\overline{A}\,\overline{B}+BC)}$

1.17　试画出用或非门和反相器实现下列函数的逻辑图。

(1) $Y=A\,\overline{B}C+B\,\overline{C}$　　(2) $Y=(A+C)(\overline{A}+B+\overline{C})(\overline{A}+\overline{B}+C)$

(3) $Y=\overline{(AB\,\overline{C}+\overline{B}\,\overline{C})}\overline{D}+\overline{A}\,\overline{B}D$　　(4) $Y=\overline{\overline{C}\,\overline{D}\,\overline{BC}\,\overline{ABC}\,\overline{D}}$

1.18　什么叫约束项？什么叫任意项？什么叫逻辑函数式中的无关项？

1.19　对于互相排斥的一组变量 A、B、C、D、E(即任何情况下 A、B、C、D、E 不可能有两个或两个以上同时为 1)，试证明 $A\,\overline{B}\,\overline{C}\,\overline{D}\,\overline{E}=A$，$\overline{A}B\,\overline{C}\,\overline{D}\,\overline{E}=B$，$\overline{A}\,\overline{B}C\,\overline{D}\,\overline{E}=C$，$\overline{A}\,\overline{B}\,\overline{C}D\,\overline{E}=D$，$\overline{A}\,\overline{B}\,\overline{C}\,\overline{D}E=E$.

1.20　用卡诺图法将下列函数化简为最简与或函数式。

(1) $Y=\overline{A+C+D}+\overline{A}\,\overline{B}C\,\overline{D}+A\,\overline{B}\,\overline{C}D$，给定约束条件为

$$A\,\overline{B}C\,\overline{D}+A\,\overline{B}CD+AB\,\overline{C}\,\overline{D}+AB\,\overline{C}D+ABC\,\overline{D}+ABCD=0;$$

(2) $Y=C\,\overline{D}(A\oplus B)+\overline{A}B\,\overline{C}+\overline{A}\,\overline{C}D$，给定约束条件为

$$AB+CD=0;$$

(3) $Y=(A\,\overline{B}+B)C\,\overline{D}+\overline{(A+B)(\overline{B}+C)}$，给定约束条件为

$$ABC+ABD+ACD+BCD=0;$$

(4) $Y(A,B,C,D)=\sum(m_3,m_5,m_6,m_7,m_{10})$，给定约束条件为 $m_0+m_1+m_2+m_4+m_8=0$;

(5) $Y(A,B,C)=\sum(m_0,m_1,m_2,m_4)$，给定约束条件为 $m_3+m_5+m_6+m_7=0$;

(6) $Y(A,B,C,D)=\sum(m_2,m_3,m_7,m_8,m_{11},m_{14})$，给定约束条件为 $m_0+m_5+m_{10}+m_{15}=0$。

1.21　试证明两个逻辑函数间的与、或、异或运算可以通过将它们的卡诺图中对应的最小项进行与、或、异或运算来实现，如图 P1.2 所示。

电子工业出版社版权所有　盗版必究

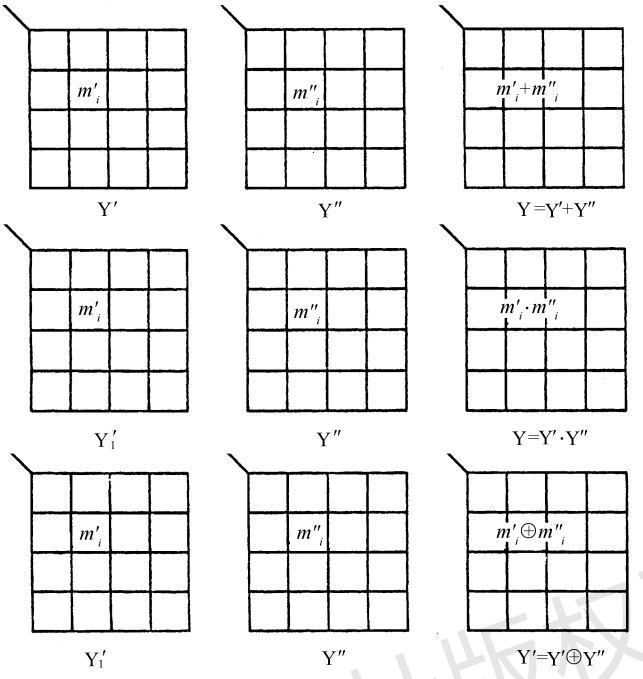


图 P1.2

1.22 利用卡诺图之间的运算(参见上题)将下列逻辑函数化简为最简与或形式。

(1) $Y=(AB+\overline{A}C+\overline{B}D)(A\overline{B}\,\overline{C}D+\overline{A}CD+BCD+\overline{B}C)$

(2) $Y=(\overline{A}\,\overline{B}C+\overline{A}B\overline{C}+AC)(A\overline{B}\,\overline{C}D+\overline{A}BC+CD)$

(3) $Y=(\overline{A}\,\overline{D}+\overline{C}D+C\overline{D})\oplus(A\overline{C}\,\overline{D}+ABC+\overline{A}D+CD)$

(4) $Y=(\overline{A}\,\overline{C}\,\overline{D}+\overline{B}\,\overline{D}+B\overline{D})\oplus(\overline{A}B\overline{D}+\overline{B}D+BC\overline{D})$

电子工业出版社版权所有 盗版必究