第 2 章　数字逻辑基础

2.1　几个基本概念

“逻辑”一词来自于逻辑学，逻辑学是研究逻辑思维与逻辑推理规律的。所谓逻辑，是指事物发生的条件和结果之间的所遵循的一种规律，即因果关系。

在生产实践中存在许多互相对立却又互相依存的两个逻辑状态，如灯的“亮”和“灭”，开关的“通”和“断”，信号的“有”和“无”，事情的“发生”和“不发生”等，这样两个状态在逻辑学中都可以用逻辑“真”和逻辑“假”来表示。当定义其中一个为“真”时，另一个一定为“假”。在数字电路中，通常用“1”来表示“真”，“0”来表示“假”，这种表示方式中不存在中间状态。需要注意的是，这里的“0”和“1”不是表示数量大小，而完全是表示事物的一种逻辑状态。

在这种情况下，我们把条件看作逻辑变量，结果看作逻辑函数，而逻辑变量和逻辑函数的取值只有“0”和“1”两种，这样就把一种逻辑问题转化为一个代数问题，这种用代数的方法去研究逻辑问题的科学称为逻辑代数。逻辑代数是 1849 年英国数学家乔治·布尔（George Bool）最早提出的，因此也称为布尔代数。1938 年克劳得·香农（Claude E. Shannon）将布尔代数理论应用到继电器开关电路的设计中，因此又称为开关代数。逻辑代数是研究数字电路的一个基本的数学工具，因此数字电路也称为逻辑电路。

数字电路中使用高低两个电平表示两种不同的电路状态，如果规定用高电平表示逻辑状态“1”，用低电平表示逻辑状态“0”，称为正逻辑；反之，称为负逻辑。两种逻辑之间是可以相互转变的，如无特殊说明，本书一般采用正逻辑。

2.2　逻辑代数及其运算

2.2.1　逻辑代数中的三种基本逻辑运算

逻辑代数和普通代数一样，用字母代表变量，逻辑代数的逻辑变量称为布尔变量。和普通代数不同的是，逻辑变量只有两种取值，即 0 或 1。并且，常量 0 和 1 没有普通代数中的

0 和 1 的意义，它只表示两种对立的逻辑状态，即命题的“假”和“真”，信号的“无”和“有”等。这种两值的变量称为逻辑变量，通常用字母 A，B，C，…来表示。

在普通代数中，函数这个概念是大家所熟悉的，即随着自变量变化而变化的因变量。与普通代数一样，在逻辑代数中，对于 n 个逻辑变量 A，B，C，…，如果有 $L=f(A, B, C, \cdots)$，则称 L 为逻辑函数。逻辑函数与逻辑变量之间的关系称为逻辑函数表达式，简称逻辑表达式。如果输入逻辑变量 A，B，C，…的取值确定了，逻辑函数的值也就被唯一地确定了。必须注意的是，在逻辑代数中，逻辑函数与逻辑变量一样只有两个取值：0 和 1。同样，这里的 0 和 1 并不表示具体的“数”，而只表示两种不同的逻辑状态。任一逻辑函数和其变量的关系，不管多么复杂，它都是由相应输入变量的与、或、非三种基本运算构成的。也就是说逻辑函数中包含三种基本运算：与、或、非，任何逻辑运算都可以用这三种基本运算来实现。通常把实现与逻辑运算的单元电路叫做与门，把实现或逻辑运算的单元电路叫做或门，把实现非逻辑运算的单元电路叫做非门（也叫做反相器）。

1. 与逻辑和与门

（1）与逻辑

先来看一个简单的例子，图 2－1 中 A、B 为两个开关，F 为灯，F 的亮灭取决于 A、B 的通断状态。

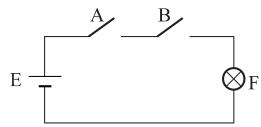


图 2－1　与逻辑电路

如果把开关的闭合和断开作为条件（或导致事物结果的原因），把灯亮作为结果，可以列出输入 A、B 与输出 F 的所有关系如表 2－1 所示。

由表可见：灯 F 亮的条件是开关 A、B 同时闭合，这种 F 与 A、B 的关系称为“与逻辑”关系。所谓与逻辑，是指只有决定事物结果的全部条件同时具备时，结果才会发生。这种因果关系叫做逻辑与，或者叫逻辑乘。在逻辑代数中，逻辑变量之间逻辑与的关系称作与运算，也叫逻辑乘法运算。

若以“1”表示开关 A、B 闭合，以“0”表示开关断开；以“1”表示灯亮，以“0”表示不亮，则可以列出输入变量 A、B 的所有取值组合与输出变量 F 的一一对应关系，这种用表格形式列出的逻辑关系，叫真值表。它是描述逻辑功能的一种重要形式。表 2－2 为与逻辑真值表。

表 2－1　与逻辑关系

A	B	F
断开	断开	灭
断开	闭合	灭
闭合	断开	灭
闭合	闭合	亮

表 2－2　与逻辑真值表

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

与逻辑还可以用输出与输入之间的逻辑关系表达式也即逻辑函数来表示，与逻辑的逻辑函数为：$F = A \cdot B$

式中符号“·”叫逻辑乘号（或逻辑与号），为了书写方便，可以省略不写。

逻辑与的基本运算规则为：

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

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

逻辑与的运算规则可以归纳为：“有0出0，全1出1”。

逻辑与的表达式可以推广到多输入变量的形式为：$F = A \cdot B \cdot C \cdot D \cdots$
或简写成：$F = ABCD \cdots$

（2）与门

能实现与逻辑运算的电路称为“与门”，它是数字电路中最基本的一种逻辑门电路。图2－2（a）是国家标准局规定的与门的标准符号，图2－2（b）为国外技术资料及绘图软件中常用的与门的符号，称为国外符号。

图2－2（a）和图2－2（b）是二输入的与门符号，当输入增加时，符号形状不变，只是输入端增加而已。

2. 或逻辑和或门

（1）或逻辑

或逻辑的例子如图2－3所示。

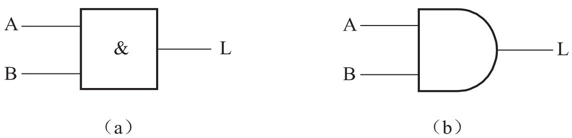


图2－2 “与”门符号

（a）标准符号；（b）国外符号

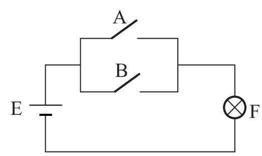


图2－3 或逻辑电路

和与逻辑分析过程类似，可以列出该电路输入A、B与输出F的所有关系组合如表2－3所示。

由此可见：灯F亮的条件是开关A、B只要有一个闭合，这种F与A、B的关系称为“或逻辑”关系。所谓或逻辑，是指只有决定事物结果的全部条件中，只要有一个成立，结果就会发生。这种因果关系叫做逻辑或，或者叫逻辑加。在逻辑代数中，逻辑变量之间逻辑加的关系称作加运算，也叫逻辑加法运算。

同理，若以“1”表示开关A、B闭合，以“0”表示开关断开；以“1”表示灯亮，以“0”表示不亮，则可以列出逻辑或的真值表如表2－4所示。

表 2－3　或逻辑关系

A	B	F
断开	断开	灭
断开	闭合	亮
闭合	断开	亮
闭合	闭合	亮

表 2－4　逻辑或真值表

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

逻辑或的表达式为：$F = A + B$　　逻辑或的运算规则为：

$$0 + 0 = 0 \qquad 0 + 1 = 1 \qquad 1 + 0 = 1 \qquad 1 + 1 = 1$$

$$0 + A = A \qquad 1 + A = 1 \qquad A + A = A$$

逻辑或的运算规则可以归纳为："有 1 出 1，全 0 出 0"。

逻辑或的表达式可以推广到多输入变量的形式为：$F = A + B + C + D\cdots$

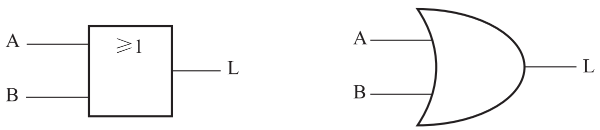
（2）或门

能实现或逻辑运算的电路称为"或门"，图 2－4（a）是或门的标准符号，图 2－4（b）为或门的国外符号。

3. 非逻辑和非门

（1）非逻辑

非逻辑的例子如图 2－5 所示。



（a）　（b）

图 2－4　"或"门符号

（a）标准符号；（b）国外符号

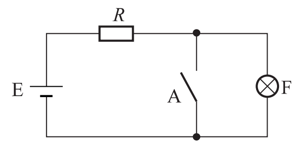


图 2－5　非逻辑电路

该电路输入 A 与输出 F 关系如表 2－5 所示。结果灯 F 的亮、灭与条件开关的闭合、断开呈现一种相反的因果关系，这种关系称为非逻辑，或者叫逻辑反。所谓逻辑非，是指条件具备，结果便不会产生；而条件不具备时，结果一定发生，即结论是对前提条件的否定。

同理，若以"1"表示开关 A 闭合，以"0"表示开关断开；以"1"表示灯亮，以"0"表示不亮，则可以列出逻辑或的真值表如表 2－6 所示。

表 2-5　非逻辑关系

A	F
闭合	灭
断开	亮

表 2-6　非逻辑真值表

A	F
0	1
1	0

在逻辑代数中，逻辑变量之间逻辑非的关系称作非运算，也叫求反运算。逻辑非的表达式为：

$$F = \overline{A}$$

逻辑非的运算规则为：$\overline{0} = 1$
$\overline{1} = 0$

（2）非门

能实现非逻辑运算的电路称为“非门”，图 2-6（a）是非门的标准符号，图 2-6（b）为非门的国外符号。

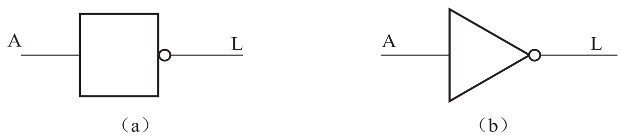


图 2-6　“非”门符号
（a）标准符号；（b）国外符号

2.2.2　逻辑代数中的五种复合逻辑运算

实际的逻辑问题往往比与、或、非复杂得多，不过它们都可以用与、或、非的组合来实现。最常用的复合逻辑运算有与非、或非、与或非、异或、同或等。表 2-7 给出了它们的表达式、真值表和逻辑符号和运算规律。

1. 与非运算

与非运算是将与运算的结果求反得到。其逻辑表达式、真值表和逻辑符号运算规律如表 2-7 所示。

2. 或非运算

或非运算是将或运算的结果求反得到。其逻辑表达式、真值表和逻辑符号和运算规律如表 2-7 所示。

3. 与或非运算

与或非运算是将 A 和 B、C 和 D 分别相与，然后将两者结果求和最后再求反得到。其逻辑表达式、真值表和逻辑符号和运算规律如表 2-7 所示。

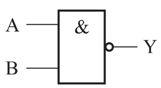
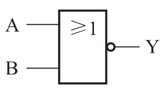
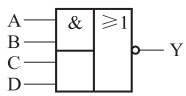
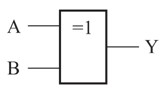
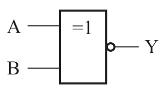
4. 异或运算

异或运算表示的逻辑关系是：当输入变量 A 和 B 的取值不同时，输出变量的值为“1”；当输入变量 A 和 B 的取值相同时，输出变量的值为“0”。其逻辑表达式、真值表和逻辑符号和运算规律如表 2-7 所示。

5. 同或运算

同或运算表示的逻辑关系是：当输入变量 A 和 B 的取值相同时，输出变量的值为“1”；当输入变量 A 和 B 的取值不同时，输出变量的值为“0”。其逻辑表达式、真值表和逻辑符号和运算规律如表 2 －7 所示。

表 2 －7　五种组合逻辑运算

逻辑名称	与非		或非		与或非		异或		同或	
逻辑表达式	$Y=\overline{AB}$		$Y=\overline{A+B}$		$Y=\overline{AB+CD}$		$Y=A\oplus B$		$Y=A\odot B$	
逻辑符号										
真值表	A　B	Y	A　B	Y	A　B　C　D	Y	A　B	Y	A　B	Y
	0　0	1	0　0	1	0　0　0　0	1	0　0	0	0　0	1
	0　1	1	0　1	0	0　0　0　1	1	0　1	1	0　1	0
	1　0	1	1　0	0	…　…　…　…		1　0	1	1　1	0
	1　1	0	1　1	0	1　1　1　1	0	1　1	0	1　1	1
逻辑运算规律	有 0 得 1 全 1 得 0		有 1 得 0 全 0 得 1		与项为 1 结果为 0 其余输出全为 1		不同为 1 相同为 0		不同为 0 相同为 1	

实现本节所述各种逻辑运算的电路称为门电路。常用集成门电路有与门、或门、非门（也称反相器）、与非门、或非门、异或门、同或门、与或非门等，它们的电路组成及工作原理将在后面章节阐述。

2.3　逻辑代数的公式和运算规则

根据逻辑代数中的与、或、非三种基本运算，可以推导出逻辑代数运算的一些基本定律，也可以称为逻辑代数的公理。熟悉这些基本定律以后，可以推出逻辑代数的一些常用公式，这些定律和公式为逻辑函数的化简提供了依据，也是分析和设计数字逻辑电路的理论工具。

2.3.1　逻辑代数中的基本公式

逻辑代数不仅有与普通代数相类似的定律，如交换律、结合律、分配律，还有它本身的一些特殊规律。逻辑代数共有八条基本定律，现将它分成三大类，列在表 2 －8 中。

表 2-8　逻辑代数的八个基本定律

与普通代数相似的定律	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	$A \cdot (B + C) = A \cdot B + AC$	$A + BC = (A + B)(A + C)$
有关变量和常量关系的定律	0，1 律	$A \cdot 1 = A$；$A \cdot 0 = 0$	$A + 1 = 1$；$A + 0 = A$
	互补律	$A \cdot \overline{A} = 0$	$A + \overline{A} = 1$
逻辑代数的特殊规律	重叠律	$A \cdot A = A$	$A + A = A$
	否定律	$\overline{\overline{A}} = A$	$\overline{\overline{A}} = A$
	反演律	$\overline{A \cdot B} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A} \cdot \overline{B}$

表中，交换律、结合律、分配律是与普通代数相似的定律；0、1 律和互补律表明了逻辑运算中常量与变量间的关系；重叠律、否定律和反演律则是逻辑代数特有的规律。

根据基本的逻辑概念及与、或、非的基本运算规律，我们一下子就可以看出下面这些定律是正确的：交换律、结合律、0、1 律、互补律、重叠律、否定律。

还有一些定律如分配律中 $A + BC = (A + B) + (A + C)$以及反演律（也称摩根定律），就不容易一下子就看出是否正确。对于这些不能一下子看出是否正确的定律，可以分别作出等式两边的真值表，再检查其结果是否相同来证明，这个方法式是最方便有效的。

例 2-1　用真值表证明 $A + BC = (A + B) + (A + C)$和$\overline{A \cdot B} = \overline{A} + \overline{B}$ 的正确性。

解：将 A、B、C 的各种取值代入上面等式的两边，得到的结果填入表中，可得真值表分别为表 2-9 和表 2-10。

表 2-9　例 2-1 真值表 1

A	B	C	BC	A + BC	A + B	A + C	(A + B)(A + C)
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

表 2－10 例 2－1 真值表 2

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

分析两表，可知两个公式的正确性，其他公式也可用相同的方法证明。

2.3.2 逻辑代数的常用公式

利用表 2－8 的基本定律，可以推演出一些在逻辑函数的化简中经常用到的公式。

公式 1 $AB+A\overline{B}=A$

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

该公式的意义在于：在一个与或表达式中，若两个乘积项分别含有同一因子的原变量和反变量，而其他因子相同，则这个乘积项可以合并，并可消去互为相反的因子，这也是卡诺图化简的理论基础。

公式 2 $A+AB=A$

证明： $A+AB=A(1+B)=A\cdot 1=A$

该公式的意义在于：在一个与或表达式中，若一个乘积项是另一个乘积项的因子，则另一个乘积项是多余的，可以消去。

公式 3 $A+\overline{A}B=A+B$

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

该公式说明：在一个与或表达式中，若一个乘积项或一个因子的反是另一个乘积项的一部分，则另一个乘积项中该乘积项或因子是多余的，可以消去。

公式 4 $AB+\overline{A}C+BC=AB+\overline{A}C$

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

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

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

$=AB+\overline{A}C$

该公式的意义在于：如果在一个与或表达式中，两个乘积项分别含有同一因子的原变量和反变量，而这两项的剩余因子正好组成第三项，则第三项是多余的，可以消去。

以上各公式也可以用真值表来证明。

2.3.3 逻辑代数的三个重要运算规则

1. 代入规则

任何一个含有某变量的等式，如果等式中所有出现此变量的位置均代之以一个逻辑函数式，则此等式依然成立。这一规则称为代入规则。

利用代入规则很容易把表 2 – 8 中的基本公式和常用公式推广为多变量的形式。用代入规则证明反演律也适用于多变量的情况。

例 2 – 2 证明：$\overline{A+B+C}=\overline{A}\cdot\overline{B}\cdot\overline{C}$

解：二变量反演律为：$\overline{A+B}=\overline{A}\cdot\overline{B}$，$\overline{A\cdot B}=\overline{A}+\overline{B}$，现以（B + C）代入$\overline{A+B}=\overline{A}\cdot\overline{B}$等式中 B 的位置，于是得到$\overline{A+B+C}=\overline{A}\cdot\overline{B+C}=\overline{A}\cdot(\overline{B}\cdot\overline{C})=\overline{A}\cdot\overline{B}\cdot\overline{C}$

同理，可以得到：$\overline{A\cdot B\cdot C}=\overline{A}+\overline{B}+\overline{C}$

2. 反演规则

对于任何一个逻辑函数 L，若同时将式中所有的“·”和“+”互换、“0”和“1”互换、“原变量”和“反变量”互换，则得到的逻辑函数就是原函数 L 的反函数。若两个函数式相等，则它们的反函数必然相等，称这一规则为反演规则。运用反演规则时必须注意运算符号的先后顺序，必须按照先括号，然后按再与、后或的顺序变换。

3. 对偶规则

在一个逻辑表达式 L 中，若将式中所有的“·”和“+”互换、“0”和“1”互换，则新得到的函数表达式 L′称为 L 的对偶函数或对偶式。若两个函数式相等，则它们的对偶函数必然相等，称这一规则为对偶规则。运用反演规则时同样必须注意运算符号的先后顺序。

利用对偶规则，表 2 – 8 八个基本公式的左边和右边可以互相推导。

2.4 逻辑函数及其表示方法

2.4.1 逻辑函数

逻辑函数是用以描述数字逻辑系统输出与输入变量之间逻辑关系的表达式。在实际的数字系统中，任何逻辑问题都可以用逻辑函数来描述。现在举一个简单例子来说明。在二层楼房装了一盏楼梯灯 L，并在一楼和二楼各装一个单刀双掷开关 A 和 B，如图 2 – 7 所示。如果用 A = 1 和 B = 1 代表开关在向上位置 a 和 a'，A = 0 和 B = 0 代表开关在向下的位置 b 和 b'；以 L = 1 代表灯

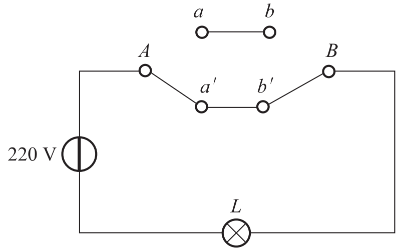


图 2 – 7 楼梯灯控制电路

亮，以 L = 0 代表灯灭，显然 A、B 的状态决定了 L 状态，则可将 A、B 的状态和 L 的状态表达为逻辑函数 L = F(A,B)。

2.4.2　逻辑函数的表示方法

在分析和处理实际的逻辑问题时，根据逻辑函数的不同特点，可以采用不同方法来表示逻辑函数。但无论采用何种表示方法，都应将其逻辑功能完全准确地表达出来。逻辑函数常用的表示方法有真值表、逻辑函数式、逻辑图和卡诺图等。这些方法以不同形式表示了同一个逻辑函数，因此，各种表示方法之间可以互相转换。下面分别加以介绍。

1. 真值表

真值表是将逻辑函数的输入变量取值的所有组合和输出取值对应关系以表格的形式列出。以图 2 – 7 楼梯灯控电路为例，可将开关 A 和 B 的四种组合和灯 L 关系列成表 2 – 11，即为表示输入与输出间逻辑关系的真值表。

表 2 – 11　灯控电路真值表

A	B	L
0	0	1
0	1	0
1	0	0
1	1	1

真值表的优点是能直观、明了地反映变量取值和函数间的关系，缺点是不能进行运算，当变量较多时，真值表就会变得比较复杂。一个确定的逻辑函数，只有一个真值表，真值表具有唯一性。

2. 逻辑表达式

逻辑表达式是用与、或、非三种基本运算组合成表示各种输入与输出之间逻辑关系的一种数学表示形式。

描述图电路中逻辑关系的函数式是：$L = \overline{A}\,\overline{B} + AB$

由此可见，前面介绍的基本逻辑式 $L = A \cdot B$、$L = A + B$、$L = \overline{A}$ 以及复合逻辑式 $L = \overline{A \cdot B}$、$L = \overline{A} + \overline{B}$ 等，都是逻辑函数式。

逻辑表达式的优点是书写方便、形式简洁，便于运算和演变，也便于用相应的逻辑图来实现，缺点是不直观。

3. 逻辑图

逻辑图是用逻辑符号表示逻辑关系的表示方法。在数字电路中，逻辑运算符号就是实现相应运算的门电路的符号。因此，逻辑图实际上就是用相应的门电路通过各种连接来实现逻辑函数。

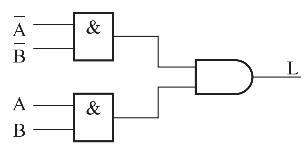


图 2 – 8　灯控电路逻辑图

根据图 2 – 7 灯控电路的逻辑功能，用逻辑图表示，如图 2 – 8 所示。

逻辑图的优点是与器件有明显的对应关系，便于制成实际的电路，缺点是不能直接进行计算。

4. 卡诺图（Karnagh Map）

卡诺图的表示方法将在 2.6 节详细介绍。

5. 各种表示方法间的互相转换

从以上的分析可知，同样一个逻辑问题，可以用多种不同的方法表示其功能，可见，它们之间是可以互相转换的。

在数字电路的分析中，经常用到的变换是：逻辑图→函数表达式→真值表；在数字电路的设计中，经常用到的变换是：真值表→卡诺图→函数表达式→逻辑图。下面介绍几种主要表示方法间的转换。

（1）真值表转换成逻辑表达式

步骤如下：

找出真值表中函数值 L 为“1”的变量取值组合，如表 2-11 中的第二行和第五行；

将这些变量取值组合分别写成乘积项，且在每个乘积项中凡变量取值为“1”的写成原变量、为“0”的写成反变量。如表中第二行 $A=0$、$B=0$，均写成反变量组成一个乘积项 $\overline{A}\,\overline{B}$，第五行 $A=1$、$B=1$，均写成原变量组成另一个乘积项 AB；将所有乘积项相加，即可得到该函数的逻辑表达式 $L=\overline{A}\,\overline{B}+AB$。

（2）逻辑表达式转换成真值表

给定逻辑函数表达式，只要将每个输入变量可能出现的取值组合与经过相应运算得到的函数值顺序列表即可。

（3）逻辑图转换成逻辑表达式

根据逻辑图逐级递推即可得到最终的逻辑表达式。详见组合逻辑电路的分析。

（4）逻辑表达式转换成逻辑图

详见组合逻辑电路的设计。

2.5 逻辑函数的公式化简法

2.5.1 化简的意义与标准

同一个逻辑问题，可以用不同形式的逻辑函数来表示。例如逻辑函数：

$$
\begin{aligned}
L &= AB+\overline{A}C && \text{与-或表达式}\\
&= \overline{\overline{AB}\cdot\overline{\overline{A}C}} && \text{与非-与非表达式}\\
&= (A+C)(\overline{A}+B) && \text{或-与表达式}\\
&= \overline{\overline{A+C}+\overline{\overline{A}+B}} && \text{与非-或非表达式}\\
&= \overline{\overline{A}\,\overline{C}+A\overline{B}} && \text{与或非表达式}
\end{aligned}
$$

五个形式不同的逻辑函数是等价的，可以对应五个形式不同的电路，但都对应同一个真

值表，可以实现同一个逻辑问题。一般来说，逻辑函数越简单，对应的电路也越简单。电路越简单，意味着电路成本越低，并且系统可靠性越高。因此，数字电路中，逻辑函数通常都要进行化简。而其中与－或表达式是逻辑函数的最基本表示形式，其对应的门电路也最常用，所以这里以最常用的“与－或型”表达式为例来介绍“最简”的标准。“最简与－或表达式”的标准是：一是与项最少；二是每个与项中的变量个数最少。与项最少，可以使电路实现时所需的逻辑门的个数最少；每个与项中的变量数最少，可以使电路实现时所需逻辑门的输入端个数最少。这样就可以保证电路最简、成本最低、可靠性最高。

常用的化简逻辑函数的方法有代数法和卡诺图法两种。同一类型的逻辑函数表达式有时候可能会有简单程度相同的多个最简与－或表达式，这与化简时所使用的方法有关，不影响逻辑问题的最简实现。

2.5.2　逻辑函数的公式化简法

1. 公式化简的方法

公式化简法的原理就是反复使用逻辑代数的八个基本定律和四个常用公式消去函数中多余的乘积项和多余的因子，以求得函数式的最简形式。公式化简法没有固定的步骤，现将经常的使用方法归纳如下。

（1）并项法

利用常用公式 1：$AB + A\bar{B} = A$，将两项合并，保留相同因子，消去互为相反的因子。

例 2－3　化简函数 $F = AB + AC + A\bar{B}\,\bar{C}$

解：$F = AB + AC + A\bar{B}\,\bar{C}$

$= A(B + C) + A\bar{B}\,\bar{C}$　　（分配律）

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

$= A$　　（$B + C$、$\overline{B + C}$相反，并项法）

（2）吸收法

利用常用公式 2：$A + AB = A$，常用公式 4：$AB + \bar{A}C + BC + AB + \bar{A}C$，吸收多余的乘积项。

例 2－4　化简函数 $F = A\bar{C} + A\overline{BC} + BC$

解：$F = A\bar{C} + A\bar{B}\,\bar{C} + BC = A\bar{C} + BC$　　（吸收$A\bar{B}\,\bar{C}$）

（3）消去法

利用常用公式 3：$A + \bar{A}B = A + B$，消去多余的因子。

例 2－5　化简函数 $F = AB + \bar{A}C + \bar{B}C$

解：$F = AB + \bar{A}C + \bar{B}C = AB + (\bar{A} + \bar{B})C$　　（分配律）

$= AB + \overline{AB}C$　　（反演律）

$= AB + C$　　（消去$\overline{AB}$）

（4）配项法

利用 $A = A(B + \bar{B})$ 增加必要的乘积项，然后再用公式进行化简。

例 2-6　化简函数 $Y = A\bar{B} + \bar{A}\bar{C} + \bar{B}\bar{C}$

解：
$$
\begin{aligned}
Y &= A\bar{B} + \bar{A}\bar{C} + \bar{B}\bar{C} \\
&= A\bar{B} + \bar{A}\bar{C} + \bar{B}\bar{C}(A + \bar{A}) = A\bar{B} + \bar{A}\bar{C} + A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} && \text{（配项展开）} \\
&= (A\bar{B} + A\bar{B}\bar{C}) + (\bar{A}\bar{C} + \bar{A}\bar{B}\bar{C}) && \text{（交换律）} \\
&= A\bar{B} + \bar{A}\bar{C} && \text{（消去}A\bar{B}\bar{C}\text{、}\bar{A}\bar{B}\bar{C}\text{）}
\end{aligned}
$$

利用配项法化简是需要较强的技巧和经验，若使用不当，可能会导致转了一圈却没有任何进展，甚至越配越繁。所以配项法化简只能一步一步试探性使用。

2. 综合化简

实际化简时，往往需要综合运用上述几种方法进行化简，才能得到最简的结果。

例 2-7　用公式化简法将下列逻辑函数化简成最简与或表达式。

解：
$$
\begin{aligned}
F &= \overline{A\bar{C}}B + \overline{A\bar{C} + B} + BC \\
&= \overline{A\bar{C}}B + \overline{A\bar{C}} \cdot \bar{B} + BC && \text{（反演律）} \\
&= \overline{A\bar{C}} + BC && \text{（并项法）} \\
&= \bar{A} + C + BC && \text{（反演律）} \\
&= \bar{A} + C && \text{（吸收法）}
\end{aligned}
$$

由以上的化简过程可以看出，用公式法化简需要在熟悉逻辑代数公式的基础上具有一定的经验和技巧才能完成，在化简一些较为复杂的逻辑函数时有时很难判定化简结果是否最简。

因此下一节详细介绍逻辑函数化简的另一种常用的图形方式即卡诺图化简法。

2.6　逻辑函数的卡诺图化简法

2.6.1　逻辑函数的最小项表达式

1. 最小项的定义

在 n 个输入变量的逻辑函数中，如果一个乘积项包含 n 个变量，而且每个变量以原变量或反变量的形式出现且仅出现一次，那么该乘积项称为该函数的一个最小项。

例如：在两变量逻辑函数 $Y = F(A, B)$ 中，根据最小项的定义，它们组成的四个乘积项：AB、$A\bar{B}$、$\bar{A}B$、$\bar{A}\bar{B}$ 是最小项。而根据定义 $A + B$、$A\bar{A}$、$AB\bar{B}$等不是最小项。

对 n 个输入变量的逻辑函数来说，共有 2^n 个最小项。如上例中对于 $n = 3$ 的三变量函数来说，最小项共有 4 个。

2. 最小项的编号

2^n 个最小项通常需要进行编号。一般最小项用 m_i 表示，下标 i 即最小项编号，用十进

制数表示。编号的方法是：先将最小项的原变量用“1”、反变量用“0”表示，构成二进制数；将此二进制数转换成相应的十进制数就是该最小项的编号。三个变量的最小项编号如表 2－12 所示。

表 2－12　三变量的最小项编号

最小项	变量取值			最小项编号
	A	B	C	
$\overline{A}\,\overline{B}\,\overline{C}$	0	0	0	m_0
$\overline{A}\,\overline{B}C$	0	0	1	m_1
$\overline{A}B\overline{C}$	0	1	0	m_2
$\overline{A}BC$	0	1	1	m_3
$A\overline{B}\,\overline{C}$	1	0	0	m_4
$A\overline{B}C$	1	0	1	m_5
$AB\overline{C}$	1	1	0	m_6
ABC	1	1	1	m_7

3. 最小项的性质

由上表 2－12 不难看出，对于一个 n 输入变量的函数，其最小项具有如下一些性质：

① 对于任意一个最小项，只有变量的一组取值使得它的值为 1，而取其他值时，这个最小项的值都是“0”。

例如：对于最小项 $\overline{A}B\overline{C}$，只有当 ABC＝010 时，该最小项才为 1，其他七种取值 $\overline{A}B\overline{C}$ 均为“0”。

② 对于任意一组变量的取值，只有一个最小项的值为 1，而其他最小项的值为 0。

例如：当 ABC＝010 时，只有最小项 $\overline{A}B\overline{C}$ 的值为 1，而其他七个最小项的值为 0。

③ 任何两个最小项的积恒为“0”。即 $m_i \cdot m_j = 0 \quad (i \neq j)$。

④ 对于任意一种取值全体最小项之和为 1。即 $\sum m_i = 1\ (i = 0 \sim 2^n - 1)$

由②可以直接推出③和④，③和④是②的进一步表述。

⑤ 若两个最小项之间只有一个变量不同，其余各变量均相同，则称这两个最小项满足逻辑相邻。每个最小项有 n 个最小项与之相邻。

2.6.2 逻辑函数的卡诺图表示法

1. 卡诺图

卡诺图是按相邻原则排列的最小项的方格图。它是由美国工程师卡诺（Karnaugh）首先提出来的，所以称为卡诺图（Karnaugh Map）。

n 个变量的逻辑函数，由 2^n 个最小项组成，将这些最小项用几何相邻来反映逻辑相邻排列成方格图的形式，就成为卡诺图。

下面具体介绍一变量～五变量的卡诺图：

一变量卡诺图：如图 2－9（a）。它有 $2^1=2$ 个最小项，分别为 A、$\overline{A}$，因此有二个方格。卡诺图上面的 0 表示反变量，1 表示原变量。左上方标注变量 A，每个小方格对应着一个最小项。

二变量卡诺图：如图 2－9（b）所示。它有 $2^2=4$ 个最小项，因此有四个方格。卡诺图上面和左面的 0 表示反变量，1 表示原变量。左上方标注变量，斜线下面为 A，上面为 B，A 和 B 可以交换。每个小方格对应着一个最小项，这四个最小项按相邻原则排列，如 m_0 和 m_1，m_1 和 m_2，m_2 和 m_3 均相邻。这里要注意的是：一行的两头 m_0 和 m_2 也相邻，因为 $\overline{A}\,\overline{B}$ 和 $A\overline{B}$ 只有一个变量不同。

三变量卡诺图：如图 2－9（c）所示。它有 $2^3=8$ 个最小项，因此有八个方格。每个小方格对应着一种变量的取值，即对应着一个最小项。这八个最小项按相邻原则排列。其中 m_0 和 m_2，m_4 和 m_6 也相邻。右边为它的简化形式，每个小方格的右下方标注所对应的最小项的下标。

四变量卡诺图：如图 2－9（d）所示。它有 $2^4=16$ 个最小项，因此有十六个方格。每个小方格对应着一种变量的取值，即对应着一个最小项。这十六个最小项按相邻原则排列。这里要注意：m_0 和 m_2，m_4 和 m_6，m_{12} 和 m_{14}，m_8 和 m_{10} 均分别左右相邻；m_0 和 m_8，m_1 和 m_9，m_3 和 m_{11}，m_2 和 m_{10} 均上下相邻。右边为它的简化形式，每个小方格的右下方标注所对应的最小项的下标。

五变量卡诺图：如图 2－9（e）所示。它有 $2^5=32$ 个最小项，因此有 32 个方格。每个小方格对应着一种变量的取值，即对应着一个最小项。这 32 个最小项按相邻原则排列。在五变量卡诺图上，除了以上的“左右相邻”和“上下相邻”，还有所谓“对称相邻”。如 m_3 和 m_7，m_1 和 m_5，m_9 和 m_{13}，m_{11} 和 m_{15}，m_{17} 和 m_{21}，m_{19} 和 m_{23}，m_{25} 和 m_{29}，m_{27} 和 m_{31}。右边为它的简化形式，每个小方格的右下方标注所对应的最小项的下标。

A	0	1
	$m_0=\bar{A}$	$m_1=A$

A	0	1
	0	1

（a）

AB	00	01	11	10
	$m_0=\bar{A}\bar{B}$	$m_1=\bar{A}B$	$m_3=AB$	$m_2=A\bar{B}$

AB	00	01	11	10
	0	1	3	2

（b）

A / BC	00	01	11	10
0	$m_0=\bar{A}\bar{B}\bar{C}$	$m_1=\bar{A}\bar{B}C$	$m_3=\bar{A}BC$	$m_2=\bar{A}B\bar{C}$
1	$m_4=\bar{A}BC$	$m_5=A\bar{B}C$	$m_7=ABC$	$m_6=AB\bar{C}$

A / BC	00	01	11	10
0	0	1	3	2
1	4	5	7	6

（c）

CD / AB	00	01	11	10
00	$m_0=\bar{A}\bar{B}\bar{C}\bar{D}$	$m_1=\bar{A}\bar{B}\bar{C}D$	$m_3=\bar{A}\bar{B}CD$	$m_2=\bar{A}\bar{B}\bar{C}D$
01	$m_4=\bar{A}B\bar{C}\bar{D}$	$m_5=\bar{A}B\bar{C}D$	$m_7=\bar{A}BCD$	$m_6=\bar{A}BC\bar{D}$
11	$m_{12}=AB\bar{C}\bar{D}$	$m_{13}=AB\bar{C}D$	$m_{15}=ABCD$	$m_{14}=ABC\bar{D}$
10	$m_8=A\bar{B}\bar{C}\bar{D}$	$m_9=A\bar{B}\bar{C}D$	$m_{11}=A\bar{B}CD$	$m_{10}=A\bar{B}C\bar{D}$

CD / AB	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

（d）

CDE / AB	000	001	011	010	110	111	101	100
00	0	1	3	2	6	7	5	4
01	8	9	11	10	14	15	13	12
11	24	25	27	26	30	31	29	28
10	16	17	19	18	22	23	21	20

（e）

图 2 – 9　卡诺图

（a）一变量卡诺图；（b）二变量卡诺图；（c）三变量卡诺图；

（d）四变量卡诺图；（e）五变量卡诺图

五变量以上的比较复杂，一般不用，在这里不作介绍。

上面的卡诺图没有用来表示逻辑函数，称为“空卡诺图”。

2. 最小项表达式

卡诺图中的每一个小方格都对应一个最小项，如果一个逻辑函数表示成若干个最小项之和的形式，那么就可以用卡诺图来表示该逻辑函数。任何一个逻辑函数都可以表示成若干个最小项之和，这样的逻辑表达式称为最小项表达式。

例 2 – 8 将逻辑函数 $Y=\overline{(AB+\bar{A}\,\bar{B}+\bar{C})\,\overline{AB}}$ 展开成最小项表达式。

解：

$$
\begin{aligned}
Y &= \overline{(AB+\bar{A}\,\bar{B}+\bar{C})\,\overline{AB}} \\
&= \overline{AB+\bar{A}\,\bar{B}+\bar{C}}+\overline{\overline{AB}} && (\text{反演律}) \\
&= \overline{AB}\cdot\overline{\bar{A}\,\bar{B}}\cdot\bar{\bar{C}}+AB && (\text{反演律}) \\
&= (\bar{A}+\bar{B})(A+B)C+AB && (\text{反演律}) \\
&= \bar{A}BC+A\bar{B}C+AB && (\text{分配律}) \\
&= \bar{A}BC+A\bar{B}C+AB(C+\bar{C}) && (\text{互补律}) \\
&= \bar{A}BC+A\bar{B}C+ABC+AB\bar{C} && (\text{最小项表达式}) \\
&= m_3+m_5+m_6+m_7 && (m\ \text{型}) \\
&= \sum(3,5,6,7) && (\textstyle\sum m\ \text{型})
\end{aligned}
$$

一般来说，将一个逻辑函数表示成最小项表达式可以通过下列三个步骤进行：

① 利用反演律去掉反变量以外的“非”号。

② 利用分配律去掉所有的括号，直到得到与或表达式。

③ 如果表达式中某与项缺少变量，则利用互补律添上所缺变量，然后再用分配律展开，最后得到最小项表达式。

3. 用卡诺图表示逻辑函数

（1）给出逻辑函数最小项之和式

由逻辑函数的最小项表达式，可以得到该逻辑函数对应的卡诺图。具体做法是在“空卡诺图”上，对表达式中出现的最小项，其对应的小方格内填上“1”；对表达式中没有出现的最小项，在其对应的小方格内填上“0”或者什么都不填。

例 2 – 9 画出逻辑函数 $F(A,B,C,D)=\sum m(0,1,2,5,7,8,10,11,14,15)$ 的卡诺图。

解：画四变量卡诺图的一般形式，然后在该图中对应于最小项的编号为 0，1，2，5，7，8，10，11，14，15 的位置填入“1”，其余位置填“0”或空着，即可得到该逻辑函数的卡诺图，如图 2 – 10 所示。

（2）给出逻辑函数一般与或式

由逻辑函数的一般与或式，可以将其化为最小项表达式，然后再得到对应的卡诺图。也可以确定使每个与项为“1”的所有输入变量取值，并在卡诺图上对应方格填“1”，而其余的方格填“0”或不填。

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

或

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

图 2－10　例 2－9 卡诺图

例 2－10　用卡诺图描述逻辑函数 $Y = A + B\overline{C}$。

解：对于第一项 A：当 ABC＝1××（×表示可以为 0，也可以为 1）时该与项为 1，在卡诺图上对应四个方格 m_4，m_5，m_6，m_7 处填 1。对于第二项$B\overline{C}$：当 ABC＝×10 时该与项为 1，在卡诺图上对应两个方格 m_2，m_6 处填 1。最小项 m_6 重复，只需填一次即可。（因为 A＋A＝A）得到的卡诺图如图 2－11 所示。

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

图 2－11　例 2－10 卡诺图

2.6.3　用卡诺图化简逻辑函数

用卡诺图表示逻辑函数的根本目的是为了利用卡诺图这种图形方式更直观地来化简逻辑函数，这种函数的化简方法称为逻辑函数的卡诺图化简法。

（1）化简依据

利用公式 $AB + A\overline{B} = A$，可以将两个最小项合并消去互非的变量。卡诺图的逻辑相邻性保证了在卡诺图中相邻两方格所代表的最小项只有一个变量不同。因此，在卡诺图中，凡是几何位置相邻的最小项均可以合并成一个“与”项，“与”项中保留相同的变量，消去互非的变量，简称“留同去异”。

（2）化简规律（合并规律）

卡诺图中合并的具体做法是：将几何相邻的“1”方格画一个包围圈，这个包围圈也称为合并圈或卡诺圈。一个卡诺圈对应一个“与”项，“与”项中保留相同的变量，消去取值不同的变量。

① 两个几何相邻的“1”方格可以合并为一个“与”项，“与”项中保留相同的变量，消去一个取值不同的变量。例如图 2－12。

在图 2－12（a）中，m_1 和 m_5 为两个相邻“1”格，$m_1 + m_5 = \overline{B}C$；在图（b）中，$m_4$ 和 m_6 为两个相邻“1”格，$m_4 + m_6 = A\overline{C}$。图中还有其他的一些例子，请读者自行分析。

② 四个相邻的“1”方格可合并为一个“与”项，“与”项中保留相同变量，消去两个取值不同的变量。例如图 2－13。

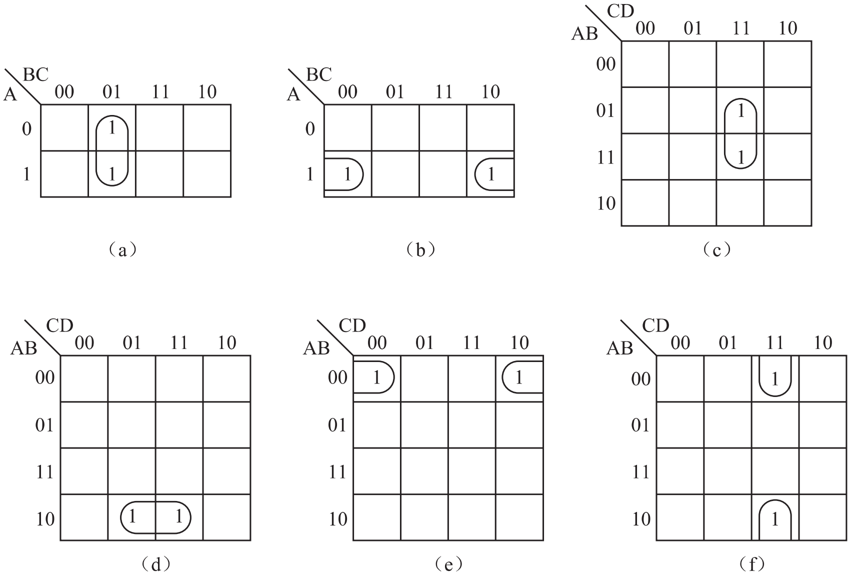


图 2－12　化简规律

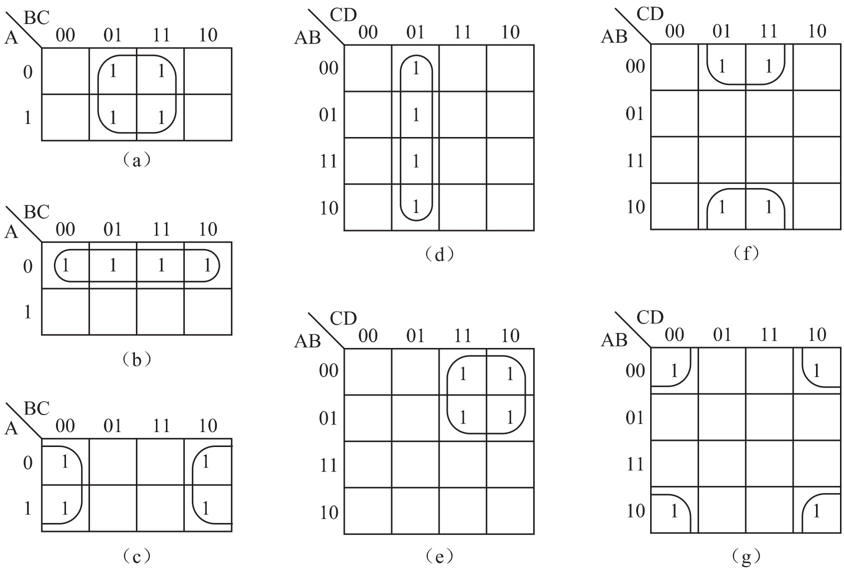


图 2－13　化简规律

图 2－13（a）图中，m_1、m_3、m_5、m_7 为 4 个相邻“1”格，把它们圈在一起加以合并可消去两个变量，$m_1 + m_3 + m_5 + m_7 = C$；图 2－13 中还有其他一些 4 个“1”格合并后消去两个变量的例子，请读者自行分析。

③ 八个相邻的“1”方格可合并为一个“与”项，“与”项中保留相同变量，消去三个取值不同的变量。例如图 2－14。

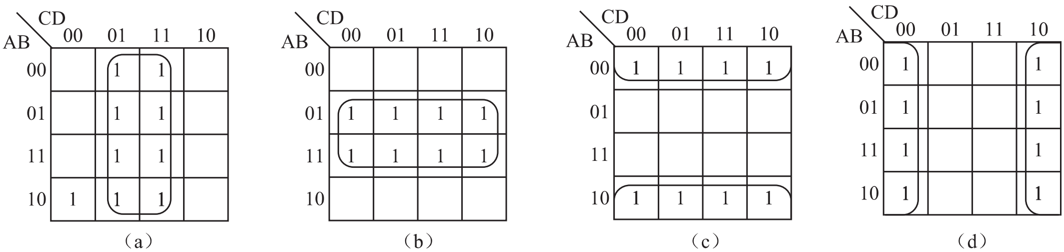


图 2－14　化简规律

图 2－14（a）、（b）中有八个“1”方格相邻，可合并为一项；（c）中 m_0、m_1、m_2、m_3 与 m_8、m_9、m_{10}、m_{11} 在几何位置上对称，在逻辑上同样相邻（m_0、m_1、m_2、m_3 合并为 $\overline{A}\,\overline{B}$，$m_8$、$m_9$、$m_{10}$、$m_{11}$ 合并为 $A\overline{B}$，$A\overline{B}$ 和 $\overline{A}\,\overline{B}$ 逻辑相邻），所以可以合并为一项 $\overline{B}$；（d）中的情况，请读者自行分析。

④ 16 个相邻的“1”方格可合并为一个“与”项，“与”项中保留相同变量，消去四个取值不同的变量。如果是四变量的卡诺图，当出现 16 个“1”方格相邻时，说明所有的最小项相加，相加的结果是恒为 1。

由以上的分析可以推出以下结论：

① 任何一个卡诺圈所含的“1”方格个数为 2^n 个，即 2、4、8、16 个。

② 几何相邻包括三种情况：一是同一行或同一列紧挨着的方格相邻；二是同一行或同一列的两头、两边、四角相邻，如图 2－12（b）、（e）、（f）和图 2－13（g）；三是以对称轴为中心对称的位置相邻，如图 2－13（c）、（f）、（g）和 2－14（d）、（e）。

③ 2^n 个“1”方格合并成一个卡诺圈，对应一个“与”项，这个“与”项的因子保留“1”方格中相同的变量，消去 n 个不同变量。

（3）化简步骤

① 用卡诺图表示逻辑函数，逻辑函数可以是最小项表达式或非最小项表达式。

② 画卡诺圈：按合并规律，将 2^n 个相邻的“1”方格圈起来合并，直到所有的“1”方格都被圈。

③ 得出化简结果：一个卡诺圈对应一个“与”项，“与”项中保留相同的变量，消去不同的变量。再将各个卡诺圈所得的“与”项相或，即得到化简后的逻辑表达式。

例 2 - 11 用卡诺图化简法求逻辑函数 F(A,B,C) = ∑(1,2,3,6,7)最简与或表达式。

解：首先，画出该函数的卡诺图。对于函数 F 的标准与或表达式中出现的那些最小项，在该卡诺图的对应小方格中填上 1，其余方格不填，结果如图 2 - 15 所示。其次，合并最小项。把图中相邻且能够合并在一起的 1 格圈在一个大圈中，如图 2 - 15 所示。最后，写出最简与或表达式。对卡诺图中所画每一个圈进行合并，保留相同的变量，去掉互反的变量。例如 $m_1 = 001$ 和 $m_3 = 011$ 合并时，保留 $\overline{A}C$，去掉互反的变量 B 和 $\overline{B}$，得到其相应的与项为 $\overline{A}C$；$m_2 = 010$、$m_3 = 011$、$m_6 = 110$ 和 $m_7 = 111$ 合并时，保留相同的变量 B，得到其相应的与项为 B。将这两个与项相或，便得到最简与或表达式：$F = \overline{A}C + B$。

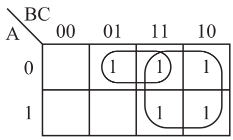


图 2 - 15 例 2 - 10 卡诺图

（4）画圈原则

为了得到最简与或表达式，画圈时必须注意如下：

① 卡诺圈应按 2^n 方格来圈，卡诺圈越大越好，卡诺圈越少越好。卡诺圈越大，“与”项中的变量越少；卡诺圈越少，“与”项越少。这样才能保证得到的是最简与或表达式。画卡诺圈时，一般从大到小画。

② 为了得到最少最大的卡诺圈，卡诺图中的“1”方格可以重复使用。“1”方格重复使用，表示同一个最小项在逻辑函数中被“加”了多次，而 A + A = A，因此不会改变原来的逻辑函数。

③ 如果某个卡诺圈中所有的“1”方格都被别的卡诺圈圈过，说明这个卡诺圈是多余的，应去掉，否则最后得到的化简结果中多了一个与项，不是最简与或表达式。

例 2 - 12 用卡诺图化简法求逻辑函数 F(A,B,C,D) = ∑(1,5,6,7,11,12,13,15)最简与或表达式。首先根据逻辑函数画出对应的四变量卡诺图，如图 2 - 16 所示。

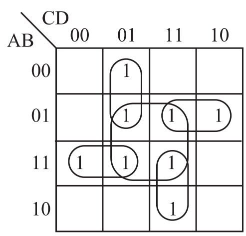


图 2 - 16 例 2 - 12 卡诺图

然后画卡诺圈。根据卡诺圈由大到小的原则，首先是将 m_5、m_7、m_{13}、m_{15} 四个最小项合并成一项为 BD，但是当将剩下的 m_1、m_6、m_{11}、m_{12} 再分别圈完卡诺圈后发现，m_5、m_7、m_{13}、m_{15} 这四个“1”构成的卡诺圈是多余的。因为它们都已经被包含在其他四个卡诺圈里。

④ 不能遗漏任何一个“1”方格。遗漏某个“1”方格，说明遗漏了某个最小项，则改变了原来的逻辑函数。

2.6.4　具有无关项的逻辑函数的化简

1. 无关项的含义

在实际的逻辑问题中，有些变量的取值是不允许、不可能、不应该出现的，或者对应输出函数值没有确定值，即函数值可以为“1”，也可以为“0”，这些取值对应的最小项称为约束项，又称为禁止项、无关项、任意项。在卡诺图或真值表中用 × 或 Φ 来表示。

含有无关项的逻辑函数，由于在无关项的相应取值下，函数值随意取成“0”或“1”都不改变原有的逻辑函数，因此对于含有约束项的逻辑函数的化简，可以利用无关项来扩大卡诺圈，即如果它对函数化简有利，则认为它是“1”；反之，则认为它是“0”。

2. 具有约束项的函数化简

具有约束项的逻辑函数时，在逻辑函数表达式中用 d（…）表示约束项。例如 $\sum d(2,4,5)$，表示最小项 m_2、m_4、m_5 为约束项。约束项在真值表或卡诺图中用 × 表示。

具有约束项的逻辑函数的卡诺图化简法实际应用中经常会遇到，例如，某逻辑电路的输入为 8421BCD 码，显然信息中有 6 个变量组合（1010～1111）是不使用的。这些变量取值所对应的最小项称为约束项。如果该电路正常工作，这些约束项决不会出现，那么与这些约束项对应的输出是什么，也就无所谓了，可以假定为“1”，也可以假定为“0”。约束项的意义在于，它的值可以取“0”，也可以取“1”，具体取什么值，可以根据使函数尽量简化这个原则而定。

例 2－13　已知 $Y(A,B,C,D)=\sum m(0,2,7,8,13,15)+\sum d(1,5,6,9,10,11,12)$ 求最简的函数表达式。

解：① $\sum d(1,5,6,9,10,11,12)$（约束条件），首先根据最小项表达式画卡诺图如图 2－17 所示。

② 画卡诺圈，将 m_1、m_2、m_5、m_9、m_{10}、m_{11} 当成“1”可以扩大卡诺圈，简化函数，所以当成“1”来使用；而将 m_6、m_{12} 当成“1”会增加“与”项数，因此得当成“0”来使用。

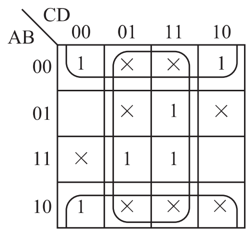


图 2－17

③ 最后得到函数的最简表达式为：$Y(A,B,C,D)=\overline{B}\,\overline{D}+D$。

本 章 小 结

1. 逻辑函数是分析和设计数字逻辑电路的重要工具。逻辑变量是一种二值变量，其取值只能是“0”和“1”，而不能有第三种取值，它仅用于表示对立的两种不同的状态。

2. 逻辑代数中有三种基本逻辑运算——与、或、非，由它们可以组合成几种基本的复合逻辑运算——与非、或非、异或、同或和与或非等。

3. 在逻辑代数的常用定律和公式中，除常量之间及常量与变量之间的逻辑运算外，还

有互补律、重叠律、交换律、结合律、分配律、吸收律、摩根定律等，其中交换律和结合律以及分配律的第一种形式和普通代数中的有关定律一样，而其他定律则完全不同，在使用时应当注意这一点。

4. 逻辑函数有四种常用的表示方法，它们是真值表、逻辑函数式、卡诺图和逻辑图，它们之间可以相互转换，在逻辑电路的分析和设计中常用到这些方法。

5. 逻辑函数的化简是分析和设计数字电路的重要环节。实现同样的功能，电路越简单，成本越低且工作越可靠。逻辑函数化简的方法主要有公式化简法和卡诺图化简法两种。公式化简法具有运算、演变直接等优点，适于各种情况的逻辑函数，但它需要对基本公式和常用公式有一定灵活运用的能力，有时难以判断化简结果的准确性。卡诺图化简法利用卡诺图中的相邻项在几何位置上也相邻的特点对相邻项进行合并，从而达到化简的目的。卡诺图化简法直观方便，便于检查化简结果的准确性，但不适于对输入变量超过 4 个的逻辑函数的化简。

6. 约束项和任意项都是无关项。它可以取“0”，也可以取“1”，根据化简的需要，应合理利用它，以得到最简与—或表达式。应当指出，无关项是为化简 1 方格（或 0 方格）服务的。当化简 1 方格需用无关项时，则无关项作 1 处理。对于没有被利用的无关项，则不能画包围圈进行化简。

习题和思考题

2-1　用真值表证明：正逻辑的“与”等于负逻辑的“或”，正逻辑的“或”等于负逻辑的“与”。

2-2　逻辑函数有哪几种表示方法？它们各自有什么意义？它们之间是如何转换的？

2-3　什么叫最小项？如何由一个任意形式的逻辑函数化成一个标准的最小项表达式？

2-4　什么叫卡诺图？卡诺图化简的依据是什么？什么叫无关项？无关项在逻辑函数的化简中有什么作用？

2-5　用代数法将下列逻辑函数化简成最简形式。

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

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

（3）$F=AC+A\overline{B}CD+ABC+\overline{C}D+ABD$

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

（5）$F=(AD+\overline{A}\,\overline{D})C+ABC+(A\overline{D}+\overline{A}D)B+BCD$

2-6　用卡诺图将下列函数化简成最简与或表达式。

（1）$F(A,B,C)=\sum m(3,5,6,7)$

（2）$F(A,B,C,D)=\sum m(0,3,5,7,8,9,11,13,15)$

（3） $F(A,B,C,D)=\sum m(0,1,2,6,8,10,11,12)$

（4） $F(A,B,C)=\sum m(0,2,4,5)$

2-7　用卡诺图化简下列逻辑函数。

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

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

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

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

2-8　用卡诺图化简下列具有无关项的逻辑函数。

（1） $F(A,B,C,D)=\sum m(0,2,6,8)+\sum d(9,15)$

（2） $F(A,B,C)=\sum m(0,1,3,7)+\sum d(2,5)$

（3） $F(A,B,C,D)=\sum m(3,5,6,9,12)+\sum d(0,1,7)$

（4） $F(A,B,C,D)=\sum m(0,1,4,7,9,10,15)+\sum d(2,5,8,12,15)$