

第3章

布尔代数基础



课程思政



知识导学



习题答案

布尔代数是一种用于描述客观事物逻辑关系的数学方法,由英国数学家乔治·布尔(George Boole)于1847年提出,他成功地将形式逻辑归结为一种代数演算。布尔代数也称“逻辑代数”。布尔代数有一套完整的运算规则,包括公理、定理和定律。它被广泛地应用于开关电路和数字逻辑电路的变换、分析、化简和设计上,因此也被称为开关代数。随着数字技术的发展,布尔代数已经成为分析和设计逻辑电路的基本工具和理论基础。

本章将从实用的角度介绍布尔代数中的逻辑运算、基本定律和基本规则,并在此基础上,介绍逻辑函数的表示及化简方法。

3.1 逻辑函数与逻辑运算

3.1.1 逻辑函数的基本概念

用来描述任何一种具体事物因果关系的公式称为逻辑函数,逻辑函数的一般表达式为

$$Y = F(A, B, C, D, \dots)$$

式中, Y 为输出变量; $A, B, C, D, \dots$ 为输入变量; F 为输出变量与输入变量的逻辑关系。

逻辑函数中的变量称为逻辑变量。逻辑变量的取值仅为0和1,0和1并不表示数量的大小,而是表示两种不同的逻辑状态,例如,用1和0表示是和、非、真和假、高电平和低电平、开和关等。任何一个逻辑函数均可由“与”、“或”、“非”三种基本逻辑运算组合而成。

3.1.2 基本逻辑运算

布尔代数中的基本逻辑运算只有“与”“或”“非”三种。

1. “与”运算

在逻辑问题的描述中,只有当决定事物结果的所有条件全都具备时,结果才会发生,则称这种因果关系为与逻辑。在布尔代数中,与逻辑关系用与运算描述。与运算又称为逻辑乘,其运算符号为“ $\cdot$ ”。两变量与运算关系可表示为

$$Y = A \cdot B$$

读作“ Y 等于 A 与 B ”。其中, A, B 是参加运算的两个逻辑变量, Y 表示运算结果。意思是:若 A, B 均为1,则 Y 为1;否则, Y 为0。该逻辑关系可用表3.1来描述。

表 3.1 与运算的真值表

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

当然,与运算的条件可以有多个。在不引起混淆的前提下,“ $\cdot$ ”常被省略。

例如,在图 3.1 所示电路中,两个开关串联控制同一个灯。显然,仅当两个开关均闭合时,灯才能亮;否则,灯灭。假定开关闭合状态用 1 表示,开关断开状态用 0 表示,灯亮用 1 表示,灯灭用 0 表示,则电路中灯 Y 和开关 A、B 之间的关系即为表 3.1 所示与运算的真值表。

由表 3.1 可得出与运算的运算法则为

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

即,与逻辑的运算规律为“有 0 得 0,全 1 得 1”。

在数字系统中,实现与运算关系的逻辑电路称为与门。与逻辑和与门的逻辑符号如图 3.2 所示。其中,“与”的条件可以有多个。

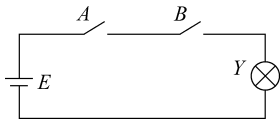


图 3.1 串联开关电路

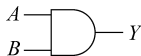


图 3.2 与逻辑和与门的逻辑符号

2. “或”运算

在逻辑问题中,如果决定事物结果的几个条件中,只要有一个或一个以上条件得到满足,结果就会发生,则称这种因果关系为或逻辑。在布尔代数中,或逻辑关系用或运算描述。或运算又称为逻辑加,其运算符号为“ $+$ ”。两变量或运算的关系可表示为

$$Y = A + B$$

读作“Y 等于 A 或 B”。意思是: A、B 中只要有一个为 1,则 Y 为 1;仅当 A、B 均为 0 时,Y 才为 0。该逻辑关系可用表 3.2 来描述。

表 3.2 或运算的真值表

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

例如,在图 3.3 所示电路中,开关 A 和 B 并联控制灯 Y。可以看出,当开关 A、B 中只要有一个闭合时,灯 Y 亮;只有当开关全部断开时,灯 Y 才灭。因此,灯 Y 与开关 A、B 之间的关系是或逻辑关系。假定开关断开用 0 表示,开关闭合用 1 表示,灯灭用 0 表示,灯亮用 1 表示,则灯 Y 与开关 A、B 的关系即为表 3.2 所示或运算的真值表。

由表 3.2 可得出或运算的运算法则为

$$0 + 0 = 0, \quad 0 + 1 = 1, \quad 1 + 0 = 1, \quad 1 + 1 = 1。$$

即,或逻辑的运算规律为“有 1 得 1,全 0 得 0”。

在数字系统中,实现或运算关系的逻辑电路称为或门。或逻辑和或门的逻辑符号如图 3.4 所示。其中,“或”的条件可以有多个。

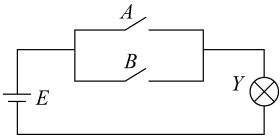


图 3.3 并联开关电路

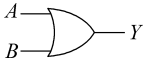


图 3.4 或逻辑和或门的逻辑符号

3. “非”运算

在逻辑问题中,如果某一事件的发生取决于条件的否定,即事件与事件发生的条件之间构成矛盾,则称这种因果关系为非逻辑。在布尔代数中,非逻辑用非运算描述。非运算也叫求反运算,其运算符号为“ $\neg$ ”。非运算的逻辑关系可表示为

$$Y = \overline{A}$$

读作“Y 等于 A 非”。意思是:若 A 为 0,则 Y 为 1;若 A 为 1,则 Y 为 0。该逻辑关系可用表 3.3 描述。

表 3.3 非运算的真值表

A	Y
0	1
1	0

例如,在图 3.5 所示电路中,开关与灯并联。显然,仅当开关断开时,灯亮;一旦开关闭合,则灯灭。令开关断开用 0 表示,开关闭合用 1 表示,灯亮用 1 表示,灯灭用 0 表示,则电路中灯 Y 与开关 A 的关系即为表 3.3 所示非运算的真值表。

由表 3.3 可得出非运算的运算法则为

$$\overline{0} = 1, \quad \overline{1} = 0.$$

在数字系统中,实现非运算功能的逻辑电路称为非门,有时又称为反相器。非逻辑和非门的逻辑符号如图 3.6 所示。其中,“非”的条件只能有一个。

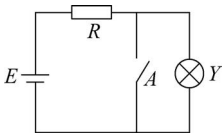


图 3.5 开关与灯并联电路

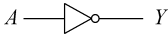


图 3.6 非逻辑和非门的逻辑符号

3.1.3 复合逻辑运算

布尔代数所定义的基本逻辑运算只有“与”“或”“非”三种,实际的逻辑问题往往是很复杂的。不过,复杂的逻辑问题都可以利用这三种基本逻辑运算组合成的复合逻辑运算来实现。常用的复合逻辑运算有“与非”“或非”“异或”“同或”等。

1. “与非”运算

“与”和“非”运算的复合运算称为与非运算,A、B 与非运算的逻辑表达式为

$$Y = \overline{AB}$$

与非运算的真值表如表 3.4 所示。

表 3.4 与非运算的真值表

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

从表 3.4 中可以看出,只有输入全为 1 时,输出才为 0; 否则输出为 1。其运算规律可归纳为“有 0 则 1,全 1 才 0”。

实现与非运算的电路称为与非门,与非逻辑和与非门的逻辑符号如图 3.7 所示。

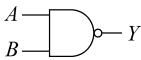


图 3.7 与非逻辑和与非门的逻辑符号

2. “或非”运算

“或”和“非”运算的复合运算称为或非运算, A 、 B 或非运算的逻辑表达式为

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

或非运算的真值表如表 3.5 所示。

表 3.5 或非运算的真值表

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

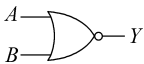


图 3.8 或非逻辑的逻辑符号

从表 3.5 中可以看出,只有输入全为 0 时,输出才为 1,否则输出为 0。其运算规律可归纳为“有 1 则 0,全 0 才 1”。

实现或非运算的电路称为或非门,或非逻辑和或非门的逻辑符号如图 3.8 所示。

3. “异或”运算

异或运算是指两个输入变量的取值相同时输出为 0,取值不相同输出为 1。若输入变量为 A 、 B ,则其逻辑表达式为

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

运算符“ $\oplus$ ”表示异或运算。异或运算的真值表如表 3.6 所示。

表 3.6 异或运算的真值表

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

由表 3.6 可得出异或运算的运算法则为

$$0 \oplus 0 = 0, \quad 0 \oplus 1 = 1, \quad 1 \oplus 0 = 1, \quad 1 \oplus 1 = 0。$$

即,异或逻辑的运算规律为“相同为 0,相异为 1”。这种逻辑关系与二进制数不进位相加的逻辑关系相同。因不进位相加的器件称为半加器,所以,异或门又称为半加器。

实现异或运算的电路称为异或门,异或逻辑和异或门的逻辑符号如图 3.9 所示。

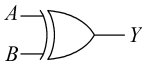


图 3.9 异或逻辑和异或门的逻辑符号

4. “同或”运算

同或运算是指两个输入变量的取值相同时输出为 1,取值不相同输出为 0。若输入变量为 A 、 B ,则其逻辑表达式为

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

运算符“ $\odot$ ”表示同或运算。同或运算的真值表如表 3.7 所示。

表 3.7 同或运算的真值表

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

由表 3.7 可得出同或运算的运算法则为

$$0 \odot 0 = 1, \quad 0 \odot 1 = 0, \quad 1 \odot 0 = 0, \quad 1 \odot 1 = 1。$$

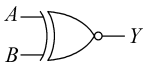


图 3.10 同或逻辑和同或门的逻辑符号

即,同或逻辑的运算规律为“相同为 1,相异为 0”。即同或运算是异或运算的非,同理,异或运算也是同或运算的非。

实现同或运算的电路称为同或门,同或逻辑和同或门的逻辑符号如图 3.10 所示。

由于同或运算是异或运算的非运算,所以有

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

或

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

由上面的讨论可见,在数字逻辑电路中,对逻辑关系的描述方法有真值表、逻辑表达式、逻辑图等,这几种表示方法是等效的。

3.1.4 逻辑函数的表示方法

表示逻辑函数的方法有很多种,常用的方法有真值表、逻辑表达式、卡诺图和逻辑图等。

1. 真值表

真值表是将输入逻辑变量的所有可能取值与相应的输出变量函数值排列在一起而组成的表格。由于每个输入变量有 0 与 1 两种取值, n 个输入变量就有 2^n 个不同的取值组合。

真值表由两部分组成,左边一栏列出输入变量的所有取值组合,为了不漏掉某一个特定的组合状态,通常各变量取值组合按二进制代码的顺序给出;右边一栏为逻辑函数值。

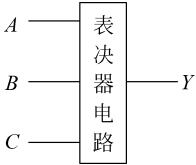


图 3.11 表决器电路示意图

例如,设计三人表决器,其电路示意图如图 3.11 所示。

A 、 B 、 C 是电路的输入,并用“1”表示赞同,用“0”表示反对;输出 Y 表示表决结果,当多数人赞同时,用 $Y=1$ 表示通过;当少数人赞同或无人赞同时,用 $Y=0$ 表示否决。三人表决器的真值表如表 3.8 所示。

表 3.8 三人表决器的真值表

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

事实上,在前面各种逻辑运算中,已经使用过真值表(见表 3.1~表 3.7)。

真值表是一种十分有用的逻辑工具。在逻辑问题的分析和设计中,将经常用到这一工具。

2. 逻辑表达式

逻辑表达式是将逻辑变量用与、或、非等运算符号按一定规则组合起来表示逻辑函数的一种方法。由真值表可以用最小项法和最大项法推导出相应的逻辑表达式。

(1) 最小项法

最小项法是把使输出为 1 的输入组合写成乘积项的形式,其中取值为 1 的输入用原变量表示,取值为 0 的输入用反变量表示,然后把这些乘积项加起来。

由表 3.8 可知,使输出为 1 的输入组合是 $ABC=011$, $ABC=101$, $ABC=110$ 和 $ABC=111$,它们对应的乘积项分别是 $\bar{A}BC$, $A\bar{B}C$, $AB\bar{C}$ 和 ABC ,把它们加起来后即得到表决器电路的逻辑表达式为

$$Y = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

式中,输出变量和输入变量之间的关系是由与和或逻辑关系组成的,且每一个输入变量均以原变量或反变量的形式在与逻辑关系中出现一次,具有这种特征的逻辑关系式称为标准与或式,也称为最小项表达式。

(2) 最大项法

最大项法是把使输出为 0 的输入组合写成和项的形式,其中取值为 0 的输入用原变量表示,取值为 1 的输入用反变量表示,然后把这些和项乘起来。

由表 3.8 可知,使输出为 0 的输入组合是 $ABC=000$, $ABC=001$, $ABC=010$ 和 $ABC=100$,它们对应的和项分别是 $(A+B+C)$, $(A+B+\bar{C})$, $(A+\bar{B}+C)$ 和 $(\bar{A}+B+C)$,把它们乘起来后即得到表决器电路的逻辑表达式为

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

式中,输出变量和输入变量之间的关系是由或和与逻辑关系组成的,且每一个输入变量均以原变量或反变量的形式在或逻辑关系中出现一次,具有这种特征的逻辑关系式称为标准或与式,也称为最大项表达式。

任何一个具体的因果关系都可以用逻辑表达式来表示,逻辑表达式简洁方便,有利于逻辑函数的化简和变换。

3. 卡诺图

卡诺图是逻辑函数的一种图形表示。一个逻辑函数的卡诺图就是将此函数的最小项表达式中的各最小项相应地填入一个方格图内,此方格图称为卡诺图。卡诺图在逻辑函数化简中十分有用,将在后面结合逻辑函数的化简问题进行详细介绍。

4. 逻辑图

逻辑图是用逻辑符号表示逻辑函数的一种方法。其中,每一个逻辑符号就是一个最简单的逻辑图。根据表决器的最小项表达式 $Y = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$,选择与或门为基本的器件就可以搭建能够实现表决器功能的逻辑图,如图 3.12 所示。

同样地,根据表决器的最大项表达式 $Y = (A+B+C)(A+B+\bar{C})(A+\bar{B}+C)(\bar{A}+B+C)$,也可以搭建以或与门为基本器件的表决器的逻辑图,如图 3.13 所示。

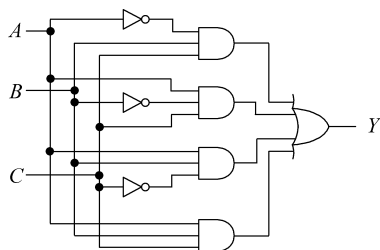


图 3.12 用与或门搭建的表决器的逻辑图

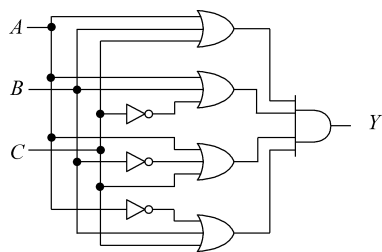


图 3.13 用或与门搭建的表决器的逻辑图

由图 3.12 和图 3.13 可见,两张逻辑图的结构完全不同,但它们能够实现相同的逻辑功能。由此可得,不同结构的逻辑图实现的逻辑功能有可能相同。

逻辑符号与实际器件有着明显的对应关系,能够方便地按逻辑图构成实际电路。因此,逻辑图也称为电路原理图。

上述表示逻辑函数的不同方法各有特点,它们各适用于不同场合。但针对某个具体问题而言,仅是同一问题的不同描述形式,彼此可以很方便地相互变换。这些表示方法可利用 Logisim 软件来仿真,用 Logisim 软件仿真的方法和结果请参阅附录 A 的内容。

3.2 布尔代数的基本定律和基本规则

3.2.1 基本定律

布尔代数的基本定律包括公理、基本公式和常用公式。

1. 公理

公理就是“0”和“1”的运算法则,它是完成运算或变换的最基本的出发点。它包括前面介绍的 3 种基本逻辑运算关系的运算法则,也就是与运算、或运算、非运算的规则。即

与运算规则: $0 \cdot 0 = 0, 0 \cdot 1 = 0, 1 \cdot 0 = 0, 1 \cdot 1 = 1$ 。

或运算规则: $0 + 0 = 0, 0 + 1 = 1, 1 + 0 = 1, 1 + 1 = 1$ 。

非运算规则: $\overline{0} = 1, \overline{1} = 0$ 。

2. 基本公式

根据布尔代数的公理,可以推导出如表 3.9 所示的布尔代数的基本公式。

表 3.9 布尔代数的基本公式

名 称	公 式	
0-1 律	$A \cdot 1 = A$	$A + 0 = A$
	$A \cdot 0 = 0$	$A + 1 = 1$
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	$A(B + C) = AB + AC$	$A + BC = (A + B)(A + C)$
摩根定律(反演律)	$\overline{A \cdot B} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A} \cdot \overline{B}$
非-非律(还原律)	$\overline{\overline{A}} = A$	

这些公式中,有一些是与普通代数不同的,在运用中要特别注意。

证明这些公式是极容易的,最直接的方法,就是将变量的各种可能取值组合代入等式中进行计算,如果等号两边的值相等,则等式成立,否则就不成立。这种证明方法称为真值表

法。例如,用真值表法对摩根定律的证明过程如表 3.10 所示。

表 3.10 证明摩根定律的真值表

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

3. 常用公式

应用基本公式,可以推导出一些其他的常用公式。直接运用这些常用公式可以给逻辑函数的化简带来很多方便。表 3.11 给出了布尔代数的一些常用公式。

表 3.11 布尔代数的常用公式

名 称	公 式	
合并律	$AB + A\overline{B} = A$	$(A+B) \cdot (A+\overline{B}) = A$
吸收律	$A + AB = A$	$A \cdot (A+B) = A$
消因律	$A + \overline{A}B = A + B$	$A \cdot (\overline{A} + B) = A \cdot B$
包含律	$AB + \overline{A}C + BC = AB + \overline{A}C$	$(A+B)(\overline{A}+C)(B+C) = (A+B)(\overline{A}+C)$

下面给出表 3.11 中左侧公式的证明过程。

合并律证明:

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

可见,如果两乘积项中分别含有 B 和 $\overline{B}$ 形式而其他因子相同,则可消去变量 B ,合并成一项。

吸收律证明:

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

可见,在两个乘积项中,如果一个乘积项是另一乘积项(如 AB)的因子,则另一个乘积项是多余的。

消因律证明:

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

可见,在两个乘积项中,如果一个乘积项的反函数(如 $\overline{A}$)是另一个乘积项的因子,则这个因子是多余的。

包含律证明:

$$\begin{aligned}
 AB + \overline{A}C + BC &= AB + \overline{A}C + (A + \overline{A})BC \\
 &= AB + \overline{A}C + ABC + \overline{A}BC \\
 &= AB(1 + C) + \overline{A}C(1 + B) \\
 &= AB + \overline{A}C
 \end{aligned}$$

推论:

$$AB + \overline{A}C + BCDE = AB + \overline{A}C$$

可见,如果一个与或表达式中的两个乘积项中,一项含有原变量(如 A),另一项含有反变量(如 $\overline{A}$),而这两个乘积项的其他因子正好是第三个乘积项(或第三个乘积项的部分因子),则第三个乘积项是多余的。

根据对偶规则,可以得到表 3.11 中的右侧公式。对偶规则将在下一节进行介绍。

3.2.2 基本规则

布尔代数的基本规则包括代入规则、反演规则和对偶规则。

1. 代入规则

代入规则是指在任何逻辑等式中,如果将等式两端的某个变量都以一个逻辑函数代入,则等式仍然成立。应用代入规则,可扩大公式的应用范围。

例如, $\overline{A+B}=\overline{A}\overline{B}$,令 $Y=B+C$,则有

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

故

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

反复运用代入规则,则有

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

同理可得

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

代入规则的正确性是显然的,因为任何逻辑函数都和逻辑变量一样,只有 0 和 1 两种可能的取值。所以用一个函数取代某个变量,等式自然成立。

值得注意的是,使用代入规则时必须将等式中所有出现同一变量的地方均以同一函数代替,否则代入后的等式将不成立。

2. 反演规则

反演规则主要是用来求逻辑函数的反函数,其方法是将原函数中所有的“ $\cdot$ ”换成“ $+$ ”、“ $+$ ”换成“ $\cdot$ ”;所有的“0”换成“1”、“1”换成“0”;所有原变量变成反变量、反变量变成原变量。这样,所得到的新逻辑表达式就是原函数的反函数。例如,若

$$Y=\overline{A}B+CD+0$$

则

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

反演规则实际上是摩根定律的推广,可通过摩根定律和代入规则得到证明。

使用反演规则时,还需注意以下两点:

- ① 所求反函数运算的优先顺序要与原函数一致;
- ② 原函数中不属于单个变量上的长非号在反函数中位置保持不变。

例如,已知函数 $Y_1=\overline{A}+\overline{B}\cdot(C+\overline{D}E)$,根据反演规则得到的反函数应该是

$$\overline{Y_1}=A\cdot[B+\overline{C}\cdot(D+\overline{E})]$$

而不应该是

$$\overline{Y_1}=A\cdot B+\overline{C}\cdot D+\overline{E}$$

再例如,已知函数 $Y_2=A+B+\overline{C}D+\overline{E}$,根据反演规则得到的反函数应该是

$$\overline{Y_2}=\overline{A}\cdot\overline{B}\cdot(C+\overline{D}E)$$

3. 对偶规则

对偶规则主要是用来求逻辑函数的对偶式,其方法是将原函数中所有的“ $\cdot$ ”换成“ $+$ ”、

“+”换成“ $\cdot$ ”；所有的“0”换成“1”、“1”换成“0”。这样，所得到的新逻辑函数就是原函数的对偶式。例如：若

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

则 Y 的对偶式 Y' 为

$$Y' = \overline{A + B\bar{C}}$$

对偶规则和反演规则的不同之处在于，不需要将原变量和反变量互换。使用对偶规则时，仍需遵循反演规则的两点注意事项。

对比表 3.9 所列的公式，除还原律外，其余同一行公式中的左侧公式和右侧公式两边的表达式是互为对偶式的。因此可知，如果两个逻辑函数相等，则它们的对偶式也相等，这称为对偶规则。

根据对偶规则，当已证明某两个逻辑表达式相等时，便可知道它们的对偶式也相等。

例如，已知 $AB + \bar{A}C + \bar{B}C = AB + C$ ，根据对偶规则可知等式两端表达式的对偶式也相等，即有

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

显然，利用对偶规则可以使定理、公式的证明减少一半。

3.3 逻辑函数的化简

对于某一给定的逻辑函数，其真值表是唯一的，但描述同一个逻辑函数的逻辑表达式却可以是多种多样的。例如，三人表决器使用的标准与或式和标准或与式，除此以外，还可利用布尔代数的基本定律将逻辑表达式转换为各种不同的形式，如与或式、或与式、与或非式、或非式、与非与非式、或非或非式等。

为了提高数字电路的可靠性，应尽可能地减少所用的元器件数量，这就需要在搭建逻辑图之前，通过化简找出逻辑函数的最简形式，这样既提高了可靠性，又节省了元器件数量，降低了成本。

在不同形式的逻辑函数表达式中，与或表达式最简单直观，同时，与或表达式也比较容易转换成其他形式的表达式。因此，我们将重点放在与或表达式的化简上。

常用的逻辑函数的化简方法有公式化简法和卡诺图化简法。

3.3.1 公式化简法

公式化简法的原理就是反复利用布尔代数的基本定理和基本规则，将逻辑表达式中多余的项和因子消掉。对逻辑函数的标准与或式进行化简，使逻辑表达式中所包含的乘积项最少，且每个乘积项的因子也最少，经过这样处理后的逻辑表达式称为最简与或式。

例如，将三人表决器的标准与或式 $Y = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$ 化简为最简与或式的过程如下：

利用布尔代数的重叠律公式 $A = A + A$ ，可将上述标准与或式改写成

$$Y = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC + ABC + ABC$$

再利用分配律,提取公因子,可得

$$Y = (\bar{A} + A)BC + A(\bar{B} + B)C + AB(\bar{C} + C)$$

利用互补律, $A + \bar{A} = 1$, 可得

$$Y = AB + BC + AC$$

式中所包含的乘积项已最少,且组成每个乘积项的因子也最少,所以它为最简与或式。

根据该最简与或式,选择与或门为基本的器件,可以搭建实现表决器逻辑功能的电路图,如图 3.14 所示。对比图 3.12 和图 3.14 可见,利用最简与或式搭建的逻辑图也是最简的。

图 3.14 的逻辑图是利用与门和或门来搭建的,逻辑图中包含与门和或门两种类型的器件。为了使逻辑图的结构更加简捷,在搭建逻辑图时,通常利用摩根定律将最简与或式转换成与非与非式,转换过程如下:

$$Y = \bar{\bar{Y}} = \overline{\overline{AB + BC + AC}} = \overline{\overline{AB} \overline{BC} \overline{AC}}$$

式中,输出变量和输入变量之间的逻辑关系完全是与非的关系,所以,称为与非与非式。根据该式搭建的表决器的逻辑图如图 3.15 所示。

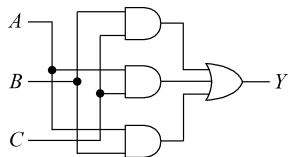


图 3.14 用最简与或式搭建的
表决器的逻辑图

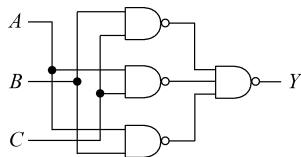


图 3.15 用与非与非式搭建的
表决器的逻辑图

公式化简法没有固定的步骤可以遵循,主要取决于对布尔代数中基本定律和基本规则的熟练掌握及灵活运用程度。尽管如此,还是可以总结出一些适用于大多数情况的常用方法,包括并项法、吸收法、消去法和配项法等。

(1) 并项法

利用合并律公式 $AB + A\bar{B} = A$, 将两个与项合并成一个与项,合并后消去互补变量 B 。例如,

$$Y = \bar{A}BC + \bar{A}\bar{B}C + \bar{A}B = \bar{A}B + \bar{A}B = \bar{A}$$

(2) 吸收法

利用吸收律公式 $A + AB = A$, 消去多余的项 AB 。例如,

$$Y = \bar{A}C + \bar{A}BC(E + F) = \bar{A}C$$

(3) 消去法

利用消因律公式 $A + \bar{A}B = A + B$, 消去多余的因子 $\bar{A}$ 。例如,

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

(4) 配项法

利用下面的公式进行配项,以消去更多的项,当然,还可以直接用来消项。

① 利用重叠律公式 $A + A = A$, 重复写入某一项,再与其他项进行合并。例如,

$$Y = \bar{A}\bar{B}C + \bar{A}BC + ABC = (\bar{A}\bar{B}C + \bar{A}BC) + (\bar{A}BC + ABC) = \bar{A}B + BC$$

② 利用互补律公式 $A + \bar{A} = 1$, 在某一项上乘以 $(A + \bar{A})$, 将一项拆成两项,以便和其他

项进行合并。例如,

$$\begin{aligned} Y &= A\bar{B} + B\bar{C} + \bar{B}C + \bar{A}B = A\bar{B} + B\bar{C} + (A + \bar{A})\bar{B}C + \bar{A}B(C + \bar{C}) \\ &= A\bar{B} + B\bar{C} + A\bar{B}C + \bar{A}\bar{B}C + \bar{A}BC + \bar{A}B\bar{C} \\ &= A\bar{B} + B\bar{C} + \bar{A}C \end{aligned}$$

③ 利用包含律公式 $AB + \bar{A}C + BC = AB + \bar{A}C$, 直接消去多余的项 BC , 或增加多余项 BC , 再与其他项进行合并。例如,

$$Y = AB + \bar{A}C + BCD = AB + \bar{A}C + BC + BCD = AB + \bar{A}C + BC = AB + \bar{A}C$$

上面介绍的是几种常用的方法, 举出的例子都比较简单。而实际应用中遇到的逻辑函数往往比较复杂, 化简时应灵活、交替地运用上述方法, 才能得到最后的化简结果。例如, 分别求下列逻辑函数的最简与或表达式:

$$\begin{aligned} Y_1 &= \bar{A}\bar{B} + AC + \bar{B}\bar{C} + \bar{A}\bar{B} + \bar{A}C + BC \\ &= \bar{A}(\bar{B} + C) + A(\bar{B} + C) + \bar{B}\bar{C} + BC \\ &= (\bar{A} + A)(\bar{B} + C) + \bar{B}\bar{C} + BC \\ &= \bar{B} + C + \bar{B}\bar{C} + BC \\ &= \bar{B} + C \\ Y_2 &= AD + \bar{A}\bar{D} + AB + \bar{A}C + BD + ACEF + \bar{B}E + DEF \\ &= A + AB + \bar{A}C + BD + ACEF + \bar{B}E + DEF \\ &= A + \bar{A}C + BD + \bar{B}E + DE + DEF \\ &= A + C + BD + \bar{B}E + DE \\ &= A + C + BD + \bar{B}E \end{aligned}$$

可以看出, 公式化简法的优点是不受变量数目的约束, 当对定理和规则十分熟练时化简比较方便; 缺点是没有一定的规律和步骤, 技巧性很强, 而且在很多情况下难以判断化简结果是否为最简。因此, 这种方法有较大的局限性。

3.3.2 卡诺图化简法

卡诺图是一种表示、化简和设计逻辑电路的重要工具。相比公式化简法, 卡诺图化简法简单、直观、容易掌握。但它也有局限性, 只能化简和设计输入变量少的简单数字逻辑电路。

1. 相关概念

(1) 最小项

最小项即一个包含了所有输入变量的乘积项(每个输入变量均以原变量或反变量的形式在乘积项中出现, 且仅出现一次)。它是根据输入变量的取值组合写出的。 n 个输入变量共有 2^n 种取值组合, 对应就有 2^n 个最小项。

例如, 3 个输入变量共有 $2^3 = 8$ 种取值组合, 就有 8 个最小项, 即输入变量的取值组合分别是 $ABC = 000, ABC = 001, ABC = 010, ABC = 011, ABC = 100, ABC = 101, ABC = 110$ 和 $ABC = 111$, 它们对应的乘积项分别是 $\bar{A}\bar{B}\bar{C}, \bar{A}\bar{B}C, \bar{A}B\bar{C}, \bar{A}BC, A\bar{B}\bar{C}, A\bar{B}C, AB\bar{C}$ 和 ABC 。



视频讲解



视频讲解



视频讲解

(2) 最小项编号

为了表示方便起见,常常把最小项排列起来,加以编号。编号后的最小项就用 m_i 的形式来表示,其中下标 i 即为编号。编号的方法是把使最小项值为 1 的取值组合当作二进制数,与这个二进制数等值的十进制数就是该最小项的编号。

例如,最小项 $\bar{A}BC$ 为 1 的取值组合为二进制数 011,对应的十进制数是 3,所以,代表 $\bar{A}BC$ 的最小项也可写成 m_3 。同样地,代表 ABC 的最小项也可写成 m_6 。

(3) 最小项的性质

从最小项的定义式出发,还可以证明它具有如下性质:

- ① 在输入变量的任何取值下,必有一个最小项的值为 1。
- ② 全体最小项之和为 1。
- ③ 任意两个最小项的乘积为 0。
- ④ 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。

两个最小项的相邻性指的是只有一个变量不同、且这个不同变量互为反变量的两个最小项。例如,最小项 ABC 和 $\bar{A}BC$ 仅有输入变量 A 不同,所以它们是具有相邻性的两个最小项,这两个最小项相加的结果可消去一对因子 A 和 $\bar{A}$,消去的过程如下:

$$Y = ABC + \bar{A}BC = (A + \bar{A})BC = BC$$

2. 卡诺图的结构

卡诺图(Karnaugh Map)是一种平面方格图,由莫里斯·卡诺(Maurice Karnaugh)发明。 n 个变量的卡诺图由 2^n 个小方格构成,每个小方格与一个最小项对应,并使具有逻辑相邻性的最小项在几何位置上也相邻地排列起来,所得到的图形叫作 n 变量卡诺图。卡诺图也是逻辑函数的一种表示方法,可以把卡诺图看成是真值表图形化的结果。

卡诺图一般画成正方形或矩形。卡诺图中各变量取值的顺序按照循环码规律排列,以保证在几何位置上相邻的最小项在逻辑上也是相邻的。卡诺图中最小项的排列方案不是唯一的,但任何一种排列方案都应保证能清楚地反映最小项的相邻关系,如图 3.16 所示为 2~4 变量卡诺图。

A \ B	0	1
0	m_0	m_1
1	m_2	m_3

(a) 2变量卡诺图

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

(b) 3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}

(c) 4变量卡诺图

图 3.16 2~4 变量卡诺图

由图 3.16 可以看出,将 n 变量分成了两组,例如,将 3 变量分成了 A 一组, BC 一组,将 4 变量分成了 AB 一组, CD 一组。变量的取值 0 表示相应变量的反变量,1 表示相应变量的原变量。每组的变量取值组合按循环码规律排列。例如,2 个变量的取值组合按 $00 \rightarrow 01 \rightarrow 11 \rightarrow 10$ 排列。必须注意,这里的相邻包含头、尾两组,即 00 与 10 也是相邻的。

3. 逻辑函数的卡诺图表示

任何一个逻辑函数都可以填到与之相对应的卡诺图中,称为逻辑函数的卡诺图。对于确定的逻辑函数,其卡诺图和真值表一样,都是唯一的。

(1) 由真值表画卡诺图

由于卡诺图与真值表一一对应,即真值表的某一行对应着卡诺图的某一个小方格。因此,如果真值表中的某一行函数值为 1,则卡诺图中对应的小方格填 1;如果真值表某一行的函数值为 0,则卡诺图中对应的小方格填 0,这样即可以得到逻辑函数的卡诺图。例如,由表 3.8 三人表决器的真值表,可画出三人表决器的卡诺图表示,如图 3.17 所示。

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

图 3.17 三人表决器的卡诺图表示

(2) 由最小项表达式画卡诺图

当逻辑函数为最小项表达式时,只需在卡诺图上找出和表达式中最小项对应的小方格填上 1,其余小方格填上 0,即可得到该逻辑函数的卡诺图。例如,由三人表决器的最小项表达式 $Y = \bar{A}BC + A\bar{B}C + ABC\bar{C} + ABC$,同样可画出如图 3.17 所示的三人表决器的卡诺图。

逻辑函数的最小项表达式还可用 m_i 的求和形式来表示。例如,三人表决器的最小项表达式 $Y = \bar{A}BC + A\bar{B}C + ABC\bar{C} + ABC$ 还可以表示为

$$Y(A, B, C) = m_3 + m_5 + m_6 + m_7 = \sum m(3, 5, 6, 7)$$

式中, Σ 为逻辑或。

例如,4 变量函数 $Y(A, B, C, D) = \sum m(0, 1, 3, 5, 6, 9, 11, 12, 13, 15)$ 的卡诺图如图 3.18 所示。

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

图 3.18 函数 $Y(A, B, C, D) = \sum m(0, 1, 3, 5, 6, 9, 11, 12, 13, 15)$ 的卡诺图

(3) 由逻辑函数表达式画卡诺图

当逻辑函数不是最小项表达式时,首先把逻辑函数表达式展开成最小项表达式,然后在每一个最小项对应的小方格内填 1,其余的小方格内填 0,这样就得到了该逻辑函数的卡诺图。例如,画函数 $Y = AB + BC$ 的卡诺图时,首先将其展开成最小项表达式,即

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

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

$$\text{或者表示为 } Y(A, B, C) = \sum m(3, 6, 7)。$$

这样,即可按最小项表达式画出如图 3.19 所示的卡诺图。

4. 用卡诺图化简逻辑函数

卡诺图的结构特点使卡诺图具有一个重要性

图 3.19 函数 $Y = AB + BC$ 的卡诺图

质:可以从图形上直观地找出相邻最小项。当相邻的方格都为1时,可将其合并,利用公式消去一个或多个变量,合并的结果是消去不同的变量、保留相同的变量,进而达到化简逻辑函数的目的。

(1) 卡诺图中最小项合并的规律

一般来说, 2^n 个相邻最小项合并时,可消去 n 个变量,即2个相邻最小项合并成一项时,可消去1个变量;4个相邻最小项合并成一项时,可消去2个变量;8个相邻最小项合并成一项时,可消去3个变量。

(2) 用卡诺图化简逻辑函数的步骤

用卡诺图化简逻辑函数时可以按如下步骤进行:

- ① 先将函数填入相应的卡诺图中;
- ② 按作圈原则将图上填1的小方格圈起来。每个圈就是一个乘积项。
- ③ 由作圈的结果写出最简与或表达式。

(3) 作圈的原则

作圈时,应遵循如下原则:

- ① 必须按 2^i ($i=0,1,2,3,\dots,n$)的规律来圈取值为1的相邻最小项;
- ② 每个取值为1的相邻最小项至少圈一次,可以圈多次。
- ③ 圈数要最少,这样可以使化简后的逻辑函数的乘积项数最少;圈中应包含尽量多的最小项,这样可以使乘积项的因子最少。

- ④ 每个圈中应至少可以找到一个只被圈过一次的最小项,避免多余项。

画圈完成后,应当检查是否符合以上原则。尤其要注意是否漏圈了最小项(独立的最小项单独作圈);圈是否圈小了(相邻还包括上下底边相邻,左右边相邻,四角相邻);是否有多余的圈(某个圈中的1都被其他圈圈过)。只要能正确画圈,就能获得函数的最简与或表达式。

(4) 由圈写最简与或表达式的方法

由圈写最简与或表达式的方法如下。

- ① 将每个圈用一个乘积项表示。其中,圈内各最小项中相同的因子保留,相异的因子消去;相同因子取值为1时用原变量表示,取值为0时用反变量表示。
- ② 将各乘积项相或,便得到最简与或表达式。

例如,用卡诺图化简法化简逻辑函数 $Y(A,B,C,D) = \sum m(0,2,5,6,7,8,10,14,15)$ 为最简与或式的过程如下:

首先,将函数填入相应的卡诺图中,然后,按作圈原则将图上填1的小方格圈起来,如图3.20所示。

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

图 3.20 函数 $Y(A,B,C,D) = \sum m(0,2,5,6,7,8,10,14,15)$ 的卡诺图及作圈

最后,由作圈的结果写出最简与或表达式为 $Y = \overline{B}\overline{D} + BC + \overline{A}BD$ 。

同样地,逻辑函数 $Y(A, B, C, D) = \sum m(3, 4, 5, 7, 9, 13, 14, 15)$ 的卡诺图及作圈如图 3.21 所示,得到最简与或表达式为 $Y = \overline{A}B\overline{C} + \overline{A}CD + A\overline{C}D + ABC$ 。

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

图 3.21 函数 $Y(A, B, C, D) = \sum m(3, 4, 5, 7, 9, 13, 14, 15)$ 的卡诺图及作圈

需要说明的是,在有些情况下,不同圈法得到的与或表达式都是最简形式,即一个函数的最简与或表达式不是唯一的。

例如,用卡诺图化简法化简逻辑函数 $Y = A\overline{C} + \overline{A}C + \overline{B}C + B\overline{C}$, 分别得到如图 3.22(a) 或(b)所示的两种作圈形式。

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

(a)

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

(b)

图 3.22 函数 $Y = A\overline{C} + \overline{A}C + \overline{B}C + B\overline{C}$ 的卡诺图及作圈

对应得到最简与或表达式分别为 $Y = \overline{A}C + A\overline{B} + B\overline{C}$, 或 $Y = \overline{A}B + A\overline{C} + \overline{B}C$ 。两个结果虽然形式不同,但功能相同。

5. 具有无关项的逻辑函数的化简

在某些实际问题的逻辑关系中,有时会遇到这样的问题:对应于输入变量的某些取值组合,函数的值可以是任意的,或者这些输入变量的取值根本不会也不允许出现,这些输入变量取值所对应的最小项称为无关项或任意项、约束项。

例如,当 8421BCD 码作为输入变量时,1010~1111 这 6 种状态所对应的最小项就是无关项。

从定义可以看出,无关项是不决定函数值的最小项,也就是说,与无关项对应的逻辑函数值既可以看成 1,也可以看成 0。因此在卡诺图或真值表中,无关项常用 ϕ 、d 或 x 来表示。在标准与或式中,用 ϕ_i 、 d_i 或 x_i 的求和形式来表示。

因为无关项的值可以根据需要取 0 或取 1,所以在用卡诺图化简逻辑函数时,充分利用无关项(作圈时,圈内的无关项取值为 1,圈外的无关项取值为 0),可以使逻辑函数进一步得到简化。

例如,求一位十进制数(采用 8421BCD 码)的四舍五入进位控制信号的最简与或表达式。

设电路输入信号 A 、 B 、 C 、 D 是一位十进制数 x 的 8421BCD 码,其中, A 为高位, D 为低位; 电路输出信号 Y 为四舍五入进位控制,即当数 $x \geq 5$ 时输出 Y 为 1。

不利用无关项和利用无关项的卡诺图作圈结果分别如图 3.23(a)和(b)所示。

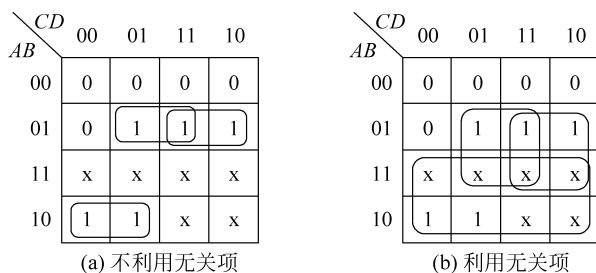


图 3.23 不利用无关项和利用无关项的卡诺图及作圈

对应得到最简与或表达式分别为 $Y = \overline{A}BD + \overline{A}BC + A\overline{B}\overline{C}$ 和 $Y = BD + BC + A$ 。

可见,充分利用无关项,可以使函数结果更简化。

例如,求逻辑函数 $Y(A, B, C, D) = \sum m(0, 3, 4, 7, 11) + \sum \phi(8, 9, 12, 13, 14, 15)$ 的最简与或表达式,卡诺图作圈结果如图 3.24 所示,化简结果为 $Y = \overline{C}\overline{D} + CD$ 。

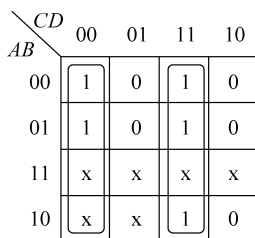


图 3.24 函数 $Y(A, B, C, D) = \sum m(0, 3, 4, 7, 11) + \sum \phi(8, 9, 12, 13, 14, 15)$ 的卡诺图及作圈

卡诺图化简法和公式化简法在功能上是等效的,但是使用卡诺图化简法更直观,更有利于初学者掌握。卡诺图的缺点是随着输入变量数量的增加,图形迅速复杂化。所以,卡诺图化简法只适于化简变量数小于 5 的逻辑函数。

习题

3.1 试用真值表、卡诺图、逻辑图表示逻辑函数 $Y = (A + B)C$ 。

3.2 用真值表证明下列等式:

(1) $A(B \oplus C) = (AB) \oplus (AC)$

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

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

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

3.3 利用反演规则和对偶规则求下列函数的反函数和对偶函数:

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

(2) $Y = [(A\overline{B} + C)D + E]G$

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

(4) $Y = \overline{AB + CD + A + \overline{C} + \overline{D} + E}$

3.4 将下列逻辑函数表示成最小项表达式的简写形式:

$$(1) Y(A, B, C) = C + ABC$$

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

3.5 用布尔代数的定理和规则证明下列等式:

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

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

$$(3) (A \oplus B) \odot (AB) = \overline{A}\overline{B}$$

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

3.6 什么叫最简与或表达式? 化简逻辑函数表达式的意义是什么?

3.7 用公式化简法求下列逻辑函数的最简与或式:

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

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

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

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

3.8 用卡诺图化简法求下列逻辑函数的最简与或表达式:

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

$$(2) Y(A, B, C, D) = \sum m(0, 1, 3, 5, 6, 9, 11, 12, 13, 15)$$

$$(3) Y(A, B, C, D) = \sum m(0, 3, 6, 9) + \sum \phi(10, 11, 12, 13, 14, 15)$$

$$(4) Y(A, B, C, D) = \sum m(2, 3, 9, 11, 12) + \sum \phi(5, 6, 8, 10, 13)$$