第 1 章

数字电路基础知识

【学习目标】1. 了解数字信号和数字电路的基本概念、特点等；
2. 理解数字逻辑电路的基本概念、基础知识、逻辑代数基础知识及化简方法；
3. 掌握逻辑代数的重要定律和规则；逻辑函数的化简方法和卡诺图化简。

【内容描述】本章主要介绍数字系统中常用的数制及其转换、码制和编码，逻辑函数及其简化，还介绍逻辑代数、逻辑函数及函数化简。在逻辑门电路，介绍门电路的外部特性、典型门电路的结构和工作特点。组合逻辑电路，介绍组合逻辑电路的分析方法、常用的组合逻辑电路和组合逻辑电路的设计方法

1.1 数字电路的概念和特点

1.1.1 数字电路的基本概念

用数字信号完成对数字量进行算术运算和逻辑运算的电路，称为数字电路(或数字系统)。由于它具有逻辑运算和逻辑处理功能，所以又称数字逻辑电路。数字电路与数字电子技术广泛地应用于电视、雷达、通信、电子、计算机、自动控制、航天等科学技术各个领域。数字电路可分为组合逻辑电路和时序逻辑电路。

组合逻辑电路，简称组合电路，它由最基本的逻辑门电路组合而成。其特点是：输出值只与当时的输入值有关。电路没有记忆功能，输出状态随着输入状态的变化而变化，类似于电阻性电路，如加法器、译码器、编码器、数据选择器等都属于此类。

时序逻辑电路，简称时序电路，它是由最基本的逻辑门电路加上反馈逻辑回路(输出到输入)或器件组合而成的电路。与组合电路相比，最本质的区别在于时序电路具有记忆功能。时序电路的特点是：输出值不仅与当时的输入值有关，而且与过去的状态有关。它类似

于含储能元件的电感或电容的电路,如触发器、锁存器、计数器、移位寄存器、存储器等电路都是时序电路的典型器件。

1.1.2 数字电路的特点

1. 算术/逻辑双功能

数字电路以二进制逻辑代数为数学基础,使用二进制数字信号,既能进行算术运算又能方便地进行逻辑运算(如与、或、非、判断、比较、处理等),因此极其适合于运算、比较、存储、传输、控制、决策等应用。

2. 简单可靠

数字电路可用于基础的数字逻辑电路的制作,其功能简单可靠且准确性高。

3. 易实现性

集成度高、功耗低是数字电路突出的优点之一。电路的设计、维修、维护灵活方便,随着集成电路技术的高速发展,数字逻辑电路的集成度越来越高,集成电路块的功能随着小规模集成电路(SSI)、中规模集成电路(MSI)、大规模集成电路(LSI)、超大规模集成电路(VLSI)的发展也从元件级、器件级、部件级、板卡级上升到系统级。电路的组成只需采用一些标准的集成电路块单元连接而成。对于非标准的特殊电路,还可以使用可编程序逻辑阵列电路,通过编程的方法实现任意的逻辑功能。

1.2 常用数制及其转换

按某种进位的原则进行计数的,称为进位计数制,简称数制。不管以哪种原则进行计数,其计数和运算都有相同的规律及特点。在我们的日常生活中最常用的是十进制,然而在计算机中只包含0和1两个数字符号。人们输入计算机的十进制被转换成二进制进行计算,计算后的结果又由二进制转换成十进制,这都由操作系统自动完成。数制也称计数制,是用一组固定的符号和统一的规则来表示数值的方法。人们通常采用的数制有十进制、二进制、八进制和十六进制。

1.2.1 常用数制

1. 十进制(D)

十进制计数法是相对二进制计数法而言的,是我们日常使用最多的计数方法,如0、1、2、3、4、5、6、7、8、9共十个数。它的进位原则是“逢十进一”,即每相邻的两个计数单位之间的进率都为十,就叫做“十进制计数法”。

实际上，任意一个十进制数 N 都可以展开为

$$(N)_{10}=(K_{n-1}\times 10^{n-1}+\cdots+K_1\times 10^1+K_0\times 10^0+K_{-1}\times 10^{-1}+\cdots+K_{-m}\times 10^{-m})=\sum_{i=-m}^{n-1}K_i\times 10^i$$

式中　$(N)_{10}$——任意的十进制数 N；

K_i——称为第 i 位的基数，即第 i 位的数字符号（简称数符），它可以是 0～9 中任意一个具体的数；

10^i——称为第 i 位的权。每位数字符号的权值是基数为 10 的不同次幂。

【例 1-1】

$$(247.76)_{10}=2\times 10^2+4\times 10^1+7\times 10^0+7\times 10^{-1}+6\times 10^{-2}$$

上面的等式是由十进制数 247.76 通过权位展开的形式，称为多项式表示法（或称按权展开式）。十进制数除了 $(247.76)_{10}$ 的形式之外，还可以写成 247.76D（D 表示十进制）的形式。

2. 二进制（B）

二进制的进位原则是“逢二进一”，基数是 2。二进制数的数符只有 0，1。任意二进制数 N 都可以用多项式的写法，其形式如下：

$$(N)_2=\pm(K_{n-1}\times 2^{n-1}+\cdots+K_1\times 2^1+K_0\times 2^0+K_{-1}\times 2^{-1}+\cdots+K_{-m}\times 2^{-m})=\sum_{i=-m}^{n-1}K_i\times 2^i$$

式中　K_i——称为第 i 位的基数，即第 i 位的数符，并且它只能是 0 和 1 两种状态之中的任意一个；

2^i——称为第 i 位的权第 i 位数字符号的权值是基数 2 的不同次幂；

m,n——称为整数，m 为小数位上的整数，n 为整数位上的整数。

【例 1-2】

$$(10110.110)_2=1\times 2^4+0\times 2^3+1\times 2^2+1\times 2^1+0\times 2^0+1\times 2^{-1}+1\times 2^{-2}+0\times 2^{-3}=(22.75)_{10}$$

每个二进制数如 10110.110，除了可以写成上述形式之外，还可以写成 10110.11B（B 表示二进制）的形式。

二进制是计算技术和电子技术中广泛采用的一种数制。二进制数是用 0 和 1 两个数码来表示的数。二进制在运算过程中遵循“逢二进一”的运算原则，这使得数制的运算更加简单，不容易发生错误；在数字系统中由于二进制数只有“0”和“1”两种状态，不容易受到其他信号的干扰，二进制具有运算简单和稳定的特点。

3. 八进制（O）

八进制数的基数是 8，一共由 8 个数码组成，分别是 0～7。它的进位原则是“逢八进一”。任意一个八进制数 N 都可以用多项式表示法表示为

$$(N)_8=\pm(K_{n-1}\times 8^{n-1}+\cdots+K_1\times 8^1+K_0\times 8^0$$

$$+K_{-1}\times 8^{-1}+\cdots+K_{-m}\times 8^{-m})=\sum_{i=-m}^{n-1}K_i\times 8^i$$

式中　K_i——第 i 位的数字符号(或称第 i 位的基数),它可以是 0～7 中的任意一个;

8^i——第 i 位数字符号的权值是基数 8 的不同次幂;

m,n——整数,m 为小数位上的整数,n 为整数位上的整数。

【例 1-3】

$$(224.36)_8=2\times 8^2+2\times 8^1+4\times 8^0+3\times 8^{-1}+6\times 8^{-2}=(148.46875)_{10}$$

每个八进制数如 34,除了可以写成$(34)_8$ 的形式之外,还可以写成 34O(O 表示八进制)的形式。

4. 十六进制(H)

十六进制数的基数是 16,一共由 16 个数码组成,分别是 0、1、2、3、4、5、6、7、8、9、A(10)、B(11)、C(12)、D(13)、E(14)、F(15)。它的进位原则是“逢十六进一”。任意一个十六进制数 N 都可以用多项式表示法表示为

$$(N)_{16}=\pm(K_{n-1}\times 16^{n-1}+\cdots+K_1\times 16^1+K_0\times 16^0$$
$$+K_{-1}\times 16^{-1}+\cdots+K_{-m}\times 16^{-m})=\sum_{i=-m}^{n-1}K_i\times 16^i$$

式中　K_i——称为第 i 位的基数,即第 i 位的数字符号,它可以是 0、1、2、3、4、5、6、7、8、8、9、A、B、C、D、E、F 中的任意一个;

16^i——称为第 i 位的权,第 i 位数字符号的权值是基数为 16 的不同次幂;

m,n——整数,m 为小数位上的整数,n 为整数位上的整数。

【例 1-4】

$$(87.2)_{16}=8\times 16^1+7\times 16^0+2\times 16^{-1}=(135.125)_{10}$$

每个十六进制数如 79,除了可以写成$(79)_{16}$的形式之外,还可以写成 79H(H 表示十六进制)的形式。

1.2.2　数制转换

在各种数制之间的转换可分为两个部分,即整数部分和小数部分。

1. 任意进制数转换成十进制数

(1) 二进制数转换成十进制数。方法:利用权位展开。

【例 1-5】

$$(1011)_2=1\times 2^3+0\times 2^2+1\times 2^1+1\times 2^0=(11)_{10}$$

(2)八进制数转换成十进制数。方法:利用权位展开。

【例 1-6】

$$(37.4)_8=3\times 8^1+7\times 8^0+4\times 8^{-1}=(31.5)_{10}$$

(3)十六进制数转换成十进制数。方法:利用权位展开。

【例 1-7】

$(9B)_{16}=9\times16^1+B\times16^0=(155)_{10}$

2. 十进制数转换成任意进制数

转换过程中分为整数部分和小数部分：

①整数部分除以基数取余数，直到商为 0 为止(所得的余数倒过来读)；

②小数部分乘以基数取整数，直到小数为 0 为止，当遇到小数部分与基数相乘无法为 0 时，按要求保留位数即可。

【例 1-8】 $(13.625)_{10}=(?)_2$

(1) 整数部分

2 | 13 …… 1
2 | 6 …… 0
2 | 3 …… 1
2 | 1 …… 1
0

(2) 小数部分

0.625
× 2
1.250 …… 1
× 2
0.500 …… 0
× 2
1.000 …… 1

即：$(13.625)_{10}=(1101.101)_2$

【例 1-9】 $(13.625)_{10}=(?)_8$

(1) 整数部分

8 | 13 …… 5
8 | 1 …… 1
0

(2) 小数部分

0.625
× 8
5.000 …… 5

即：$(13.625)_{10}=(15.5)_8$

3. 二进制数转换为八进制数和十六进制数

由于二进制数与八进制数和十六进制数之间存在着一种比较特殊的关系，它们之间满足 2^3、2^4，因此在将二进制数转换成八进制数或十六进制数的时候，从二进制数的小数点开始分别向两侧每三位(转换成八进制数时)或每四位(转换成十六进制数时)一组，若整数部分最高位不足一组，则在左边补 0 凑成一组；若小数部分最低位不足一组，则在右边补 0 凑成一组，然后将其转换成八进制数或十六进制数。

【例 1-10】 $(1101101010.0110101)_2=(?)_8=(?)_{16}$

解：$(001\ 101\ 101\ 010.011\ 010\ 100)_2=(1\ 5\ 5\ 2.3\ 2\ 4)_8=(1552.324)_8$

$(0011\ 0110\ 1010.0110\ 1010)_2=(3\ 6\ A.6\ A)_{16}=(36A.6A)_{16}$

4.八进制数、十六进制数转换为二进制数

【例 1-11】 $(226.74)_8=(?)_2$　$(B6A.63)_{16}=(?)_2$

解：$(226.74)_8=(010\ 010\ 110.111\ 100)_2$

$(B6A.63)_{16}=(1011\ 0110\ 1010.0110\ 0011)_2$

1.2.3 常用代码

若干位二进制数字符号按一定的组合方式(即码制)组合起来以表示数(包括大小和符号)和字符等信息,这就是编码。编码的方式很多种,以下介绍几种常用的数的编码与字符的编码。

在日常生活中最常用的是十进制数,而在数字电路和计算机中数字只能用二进制表示。用二进制表示的十进制数位数过多不便于读写,为了解决这一矛盾,可以把十进制数的每一位数字符号用若干位二进制数字符号表示,这种编码方式称为“二—十进制编码”,简称 BCD 码。

常见的 BCD 码有 8421 码、2421 码、5421 码和余 3 码。其中,8421 码是我们在计算机信息处理中最基本和最常见的一种 BCD 码。它是将十进制数的每个数字符号分别用 4 位二进制数字符号来表示。该 4 位二进制数每位都有固定的权值,从左至右各位权值分别为 8,4,2,1,故称为 8421 码。表 1-1 列出了几种 BCD 码与十进制数的对应关系。

表 1-1　BCD 码与十进制数的对应关系

十进制数	有权码			无权码
	8421 码	2421 码	5421 码	余 3 码
0	0000	0000	0000	0011
1	0001	0001	0001	0100
2	0010	0010	0010	0101
3	0011	0011	0011	0110
4	0100	0100	0100	0111
5	0101	1011	1000	1000
6	0110	1100	1001	1001
7	0111	1101	1010	1010
8	1000	1110	1011	1011
9	1001	1111	1100	1100

在 BCD 码中,一般分为有权码和无权码两大类。8421 码是最常用的有权码,余 3 码是无权码,其是由 8421 码加 3 后得到的。BCD 码的表示方法也很简单,就是将十进制数的各位数字符号分别用 4 位二进制数字符号表示出来。例如:

$$(34.27)_{10}=(00110100.00100111)_{8421BCD}$$
$$(01010111.00100110)_{8421BCD}=(57.26)_{10}$$

1.3　逻辑关系

在数字电路中，输入信号是“条件”，输出信号是“结果”，因此输入信号和输出信号存在着一定的因果关系，就是逻辑关系。可以用逻辑表达式、图形符号和真值表来描述它们之间的关系。

反映和处理逻辑关系的数学工具，就是逻辑代数。逻辑代数又称布尔代数或开关代数，它是分析和设计数字电路的基础。

逻辑代数和普通代数相似，都是用字母代表变量，如 A、B、C、…、X、Y、Z 等，称为逻辑变量；不同的是，逻辑变量只有两个取值：0 和 1，而在逻辑代数中 0 和 1 并不是表示具体数的大小，而是表示两种不同的状态，如高电平或低电平，电机的停止和启动，灯的亮和灭等状态。

1.3.1　三种基本逻辑关系

在逻辑代数中，有 3 种基本逻辑运算：“与”运算、“或”运算、“非”运算。

1.“与”运算

只有当出现的两种情况都同时满足时结果才会发生，这种关系就称为与逻辑关系。

如图 1-1 所示是与逻辑电路图，A、B 是两个开关，Y 是灯。当开关 A 和开关 B 同时闭合时，电路接通，Y 才会亮；当开关 A 和开关 B 中只有一个闭合时，电路都无法接通，Y 始终不亮。

若用 1 表示开关闭合和灯亮，0 表示开关断开和灯灭，则可以得到表 1-2 所示的与逻辑真值表。表 1-2 是由输入变量 A 和 B 可能出现的取值组合和相应的函数值排列而成的真值表。某个函数出现的组合个数 N 与输入变量个数 n 的关系为：$N=2^n$。例如，某个逻辑函数有 A、B、C 三个输入变量，那么这个函数可能出现 $N=2^3$ 项组合。

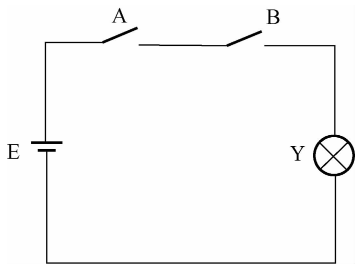


图 1-1　与逻辑电路图

表 1-2　与逻辑真值表

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

与运算的逻辑表达式为

$$Y=A\cdot B\quad 或\quad Y=AB$$

与逻辑的运算规律为：输入有 0 出 0，全 1 出 1。与运算的逻辑符号如 1-2 中(a)和(b)所示。

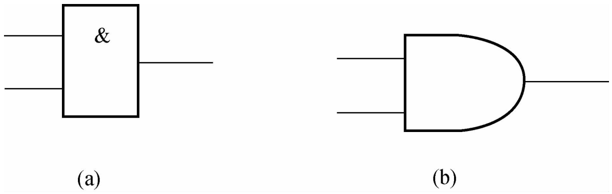


图 1-2 "与"运算的符号

(a)国际符号；(b)国际流行符号

2."或"运算

在决定事物结果的几个条件中，只要有一个或一个以上的输入条件满足，结果就会发生，这种关系就称为或逻辑关系。

如图 1-3 所示为或逻辑电路图。图中，开关 A 和开关 B 并联，控制灯 Y 的亮与灭。只要开关 A 或 B 任意一个闭合，灯 Y 都会亮。

若用 1 表示开关闭合和灯亮，0 表示开关断开和灯灭，则可得到或逻辑的真值表，如表 1-3 所示。

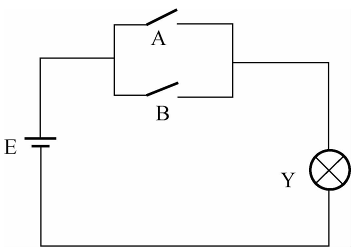


图 1-3 或逻辑电路图

表 1-3 或逻辑真值表

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

或运算的逻辑表达式为

$$Y=A+B$$

或逻辑的运算规律为：有 1 出 1，全 0 出 0。或运算的符号如图 1-4 中(a)和(b)所示。

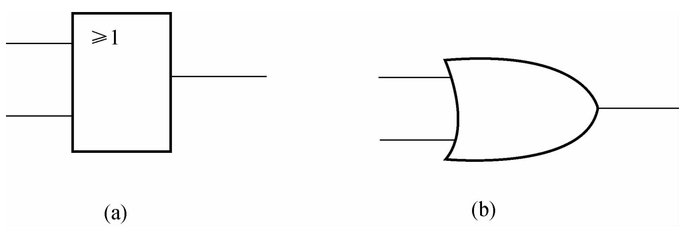


图 1-4 或逻辑的符号

(a)国际符号；(b)国际流行符号

3．“非”运算

在事件中，结果总是和输入条件相反，这种关系称为非逻辑关系。如图 1-5 所示电路中，开关 A 是否闭合对于灯亮是非逻辑关系。真值表如表 1-4 所示。在电路中，灯 Y 和开关 A 并联，当开关 A 闭合时 Y 被短路，灯 Y 不亮；当开关 A 断开时电路正常，灯 A 亮。该电路的真值表如表 1-4 所示，表中 1 表示开关闭合或灯亮，0 表示开关断开或灯不亮。

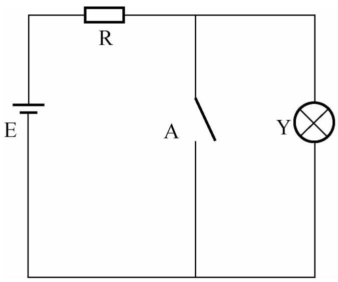


图 1-5　非逻辑电路图

表 1-4　非逻辑真值表

A	$\overline{A}$
0	1
1	0

非运算的逻辑表达式为

$$Y=\overline{A}$$

非逻辑的运算规律是：有 0 出 1，有 1 出 0。非运算的符号如图 1-6 的(a)和(b)所示。

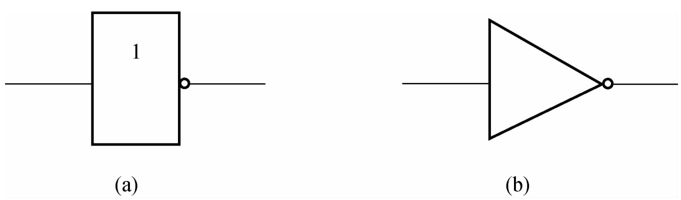


图 1-6　非逻辑的符号

(a)国际符号；(b)国际流行符号

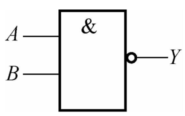
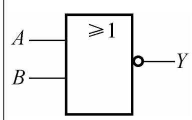
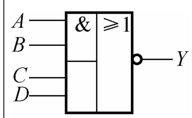
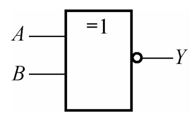
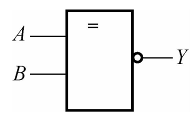
1.3.2　常见的几种复合逻辑关系

在前面介绍了三种最基本的逻辑关系：与、或、非，现在进一步学习几种常见的复合逻辑关系。复合逻辑关系是由最基本的与、或、非复合而成的。它们的逻辑表达式、逻辑符号和逻辑真值表如表 1-5 所示。

表 1-5　几种常见的复合逻辑关系

逻辑名称	与非	或非	与或非	异或	同或
逻辑表达式	$Y=\overline{AB}$	$Y=\overline{A+B}$	$Y=\overline{AB+CD}$	$Y=\overline{A}B+A\overline{B}$ 可简写成：$Y=A\oplus B$	$Y=AB+\overline{A}\,\overline{B}$ 可简写成：$Y=A\odot B$

续表

逻辑名称	与非	或非	与或非	异或	同或
逻辑符号					
真值表	A B | Y 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0	A B | Y 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0	A B C D | Y 0 0 0 0 | 1 0 0 0 1 | 1 ⋮ ⋮ ⋮ ⋮ | ⋮ 1 1 1 1 | 1	A B | Y 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0	A B | Y 0 0 | 1 1 1 | 0 2 0 | 0 1 1 | 1
逻辑运算规律	有 0 出 1 全 1 出 0	有 1 出 0 全 0 出 1	与项为 1 结果出 0 其余输出全为 1	不同出 1 相同出 0	不同出 0 相同出 1

1.4 逻辑代数的公式和定律

逻辑代数和普通代数一样，都有相对应的基本公式和定律。逻辑代数的基本公式及定律在今后的化简过程中经常会用到，这就需要我们熟记和掌握，如表 1-6 所示。

表 1-6 逻辑代数的基本公式和定律

基本公式和定律名称	逻辑与	逻辑或
0—1 律	$A\cdot 1=A$ $A\cdot 0=0$	$A+1=1$ $A+0=A$
互补律	$\overline{A}\cdot A=0$	$\overline{A}+A=1$
重叠律	$A\cdot A=A$	$A+A=A$
还原律	$\overline{\overline{A}}=A$	—
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	$A\cdot(B+C)=AB+AC$	$A+(B\cdot C)=(A+B)\cdot(A+C)$
反演律(摩根定律)	$\overline{AB}=\overline{A}+\overline{B}$	$\overline{A+B}=\overline{A}\cdot\overline{B}$

续表

基本公式和定律名称	逻辑与	逻辑或
吸收律	$A\cdot(A+B)=A$ $(A+B)(A+\overline{B})=A$ $A(\overline{A}+B)=AB$	$A+AB=A$ $AB+A\overline{B}=A$ $A+\overline{A}B=A+B$
隐含律	$(\overline{A}+B)(A+C)(B+C)=AB+\overline{A}C$ $(\overline{A}+B)(A+C)(B+C+D)$ $=(\overline{A}+B)(A+C)$	$AB+\overline{A}C+BC=AB+\overline{A}C$ $AB+\overline{A}C+BCD=AB+\overline{A}C$

在解决逻辑函数化简的过程中，上表的基本公式和定律是基础。在熟记基本公式和定律的同时还要掌握几种逻辑运算的基本规则，最常用的有代入规则、反演规则和对偶规则，下面我们通过实例来学习这些常用的运算规则。

1. 代入规则

在任一含有变量 A 的逻辑等式中，若变量 A 全部用同一逻辑函数取代，则等式仍成立，这种规则就称为代入规则。

【例 1-12】 已知等式$\overline{AB}=\overline{A}+\overline{B}$，若用 $Y=BC$ 代替等式中的 B，根据代入规则，等式仍然成立，即：

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

由此可见，摩根定律对任意多个的变量都成立。由代入规则可以推出：

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

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

2. 反演规则

在任一逻辑函数 Y 中，只要将函数中的所有“·”换成“+”，“+”换成“·”；将原变量换成反变量，反变量变成原变量，则得到 $\overline{Y}$。这个规则称为反演规则。但要注意的是，函数运算的优先顺序不能变，即：括号→乘→加，不属于单变量上的反号应保留不变。

【例 1-13】 已知 $Y=A(B+C)+CD$，求 $\overline{Y}$。

解：根据反演定理可得

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

3. 对偶规则

在任一个逻辑表达式 Y 中，如果将表达式中的“·”换成“+”，“+”换成“·”；“0”换成“1”，“1”换成“0”，变量保持不变，所得的新函数 Y'，就是函数 Y 的对偶函数。这个规则称为对偶规则。对于两个函数来说，如果原函数相等，那么它们的对偶函数、反函数也相等。

【例 1-14】 求 $Y=AB+C$ 的对偶式 Y'。

$$Y'=(A+B)\cdot C$$

1.5 逻辑函数的化简法

在前面已经学习了逻辑函数的基本公式和定律，这些都是进行逻辑函数化简的基础，是为下面学习逻辑函数化简做准备的。由逻辑函数的公式和定律可知，一个函数可以有多个不同的表达形式，例如：

$Y=AB+\overline{A}C$ 与-或表达式

$=\overline{\overline{AB}\cdot\overline{\overline{A}C}}$ 与非-与非表达式（摩根定律）

$=\overline{A\overline{B}+\overline{A}\,\overline{C}}$ 与-或-非表达式（利用反演规则并展开）

$=(\overline{A}+B)(A+C)$ 或-与表达式（将与或非式用摩根定律）

$=\overline{(\overline{\overline{A}+B})+(\overline{A+C})}$ 或非-或非表达式（将或与式用摩根定律）

在逻辑电路的设计过程中，逻辑函数最终要用逻辑电路来实现，因此化简和变换逻辑函数可以简化电路、节省器材、降低成本、提高系统的可靠性。在上面列出的 5 种逻辑表达式中，与或式和或与式是最常用的逻辑表达式。最简与或式的标准是：①含的与项最少；②各与项中含的变量数最少。最简或与项的标准是：①含的或项最少；②各或项中含的变量数最少。逻辑函数化简的方法有公式法和图形法。本节将介绍这两种化简的方法。

1.5.1 逻辑函数的公式化简法

常用的公式化简法有并项法、吸收法、消去法和配项法。

1. 并项法

利用公式 $A+\overline{A}=1$，将两项或多项合并成一项，并消去一个或多个变量。

【例 1-15】

$$\begin{aligned}Y_1&=ABC+\overline{A}BC\\&=(A+\overline{A})BC\\&=BC\end{aligned}$$

$$\begin{aligned}Y_2&=ABC+\overline{A}B+AB\overline{C}\\&=(ABC+AB\overline{C})+\overline{A}B\\&=AB(C+\overline{C})+\overline{A}B\\&=AB+\overline{A}B\\&=(A+\overline{A})B\\&=B\end{aligned}$$

2. 吸收法

利用公式 $A+1=1$ 消去多余的乘积项。

【例 1-16】

$$\begin{aligned}Y_1&=A\overline{C}+A\overline{B}\,\overline{C}+BC\\&=A\overline{C}(1+\overline{B})+BC\end{aligned}$$

$$
\begin{aligned}
&=A\overline{C}+BC\\
Y_2&=A\overline{C}+AB\overline{C}D(E+G)\\
&=A\overline{C}[1+BD(E+G)]\\
&=A\overline{C}
\end{aligned}
$$

3.消去法

利用吸收律 $A+\overline{A}B=A+B$，消去多余的因子。

【例 1-17】

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

4.配项法

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

【例 1-18】

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

在实际的解题过程中，我们经常会遇到一些综合的问题，需要综合运用上述的几种方法去进行化简，才能得到逻辑函数的最简形式。

【例 1-19】

$$
\begin{aligned}
Y&=\overline{A\overline{C}B}+\overline{A\overline{C}+B}+BC\\
&=\overline{A\overline{C}}B+\overline{A\overline{C}}\,\overline{B}+BC &&\text{（摩根定律）}\\
&=\overline{A\overline{C}}+BC &&\text{（并项法）}\\
&=\overline{A}+C+BC &&\text{（吸收法）}\\
&=\overline{A}+C
\end{aligned}
$$

1.5.2　逻辑函数的图形化简法

下面介绍的图形化简法，是将逻辑函数用卡诺图来表示，是一种更加系统并有统一规则可循的逻辑函数化简法。

1.逻辑函数的最小项的定义及性质

(1)定义。在含有 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各一次，那么该乘积项称为该函数的一个最小项。对 n 个输入变量的逻辑函数来说，共有 2^n 个最小项。

例如，含有 A、B、C 三个变量的逻辑函数，其最小项有：$\overline{A}\,\overline{B}\,\overline{C}$、$\overline{A}\,\overline{B}C$、$\overline{A}B\overline{C}$、$\overline{A}BC$、$A\overline{B}\,\overline{C}$、$A\overline{B}C$、$AB\overline{C}$、$ABC$。

(2)性质。

① 任意一个最小项,只有一组变量取值使其为 1;

②任意两个不同的最小项之积必为 0;

③n 个变量所有最小项之和为 1;

④n 个变量构成的每一个最小项都有 n 个相邻最小项。

2.最小项的卡诺图及编号

为了表达方便,最小项通常用 m_i 表示,下标 i 即最小项的编号,用十进制数表示。编号的方法是:先将最小项的原变量用 1 表示,反变量用 0 表示,构成二进制数;将此二进制数转换为相应的十进制数就是该最小项的编号。按这个原则,三变量最小项的编号如表 1-7 所示。

表 1-7　三变量最小项

最小项	变量取值 A	变量取值 B	变量取值 C	最小项编号
$\overline{A}\,\overline{B}\,\overline{C}$	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

所谓卡诺图,就是将 n 变量的全部最小项各用一个小方格表示,最小项按循环码(即相邻两组之间只有一个变量取值不同的编码)规则排列组成的方格图。如图 1-7 中的(a)和(b)分别为 3 变量函数和 4 变量函数的卡诺图。

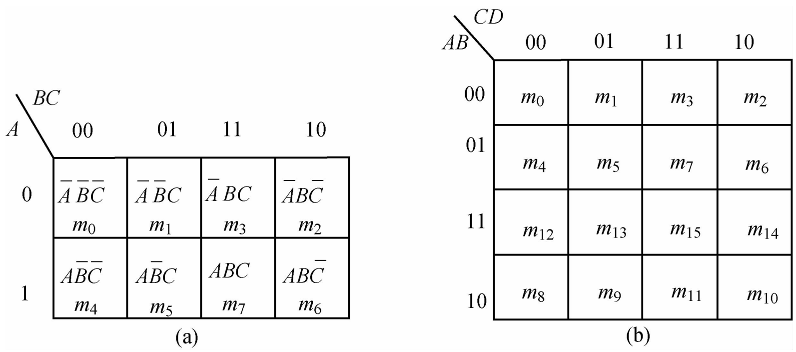


图 1-7　卡诺图

(a)3 变量函数;(b)4 变量函数

对于一组含 n 个变量的逻辑函数，共有 2^n 个最小项。如果把每个最小项用一个小方格表示，并使具有逻辑式相邻性的最小项在几何位置上也相邻地排列起来，所得到的图形就可以构成 n 个变量的卡诺图。卡诺图的特点是：在几何位置上相邻的最小项小方格在逻辑上也必定是相邻的，即相邻两项中有一个变量是互补的。

3.最小项表达式

任何一个逻辑函数都可以用若干个最小项之和的形式表示，这样的逻辑表达式称为最小项表达式，也称为逻辑函数的标准与或式。

【例 1-20】 将逻辑函数 $Y(A,B,C)=AB+\overline{B}C$ 展开，用最小项表达式表示。

解：
$$\begin{aligned}Y(A,B,C)&=AB+\overline{B}C\\&=AB(C+\overline{C})+(A+\overline{A})\overline{B}C\\&=ABC+AB\,\overline{C}+A\,\overline{B}C+\overline{A}\,\overline{B}C\end{aligned}$$

为了书写方便，通常用最小项的编号来代表最小项，即：

$$Y(A,B,C)=m_7+m_6+m_5+m_1=\sum m(1,5,6,7)$$

对于一个确定的逻辑函数，它的最小项表达式是唯一的。

【例 1-21】 求一个三变量函数 $Y(A,B,C)=\overline{AB+\overline{A}\,\overline{B}+C}+\overline{A}B$ 的最小项表达式。

解：
$$\begin{aligned}Y(A,B,C)&=\overline{AB+\overline{A}\,\overline{B}+C}+\overline{A}B\\&=\overline{AB+\overline{A}\,\overline{B}}\,\overline{C}+\overline{A}B\\&=(A\,\overline{B}+\overline{A}B)\,\overline{C}+\overline{A}B(C+\overline{C})\\&=A\,\overline{B}\,\overline{C}+\overline{A}B\,\overline{C}+\overline{A}BC+\overline{A}B\,\overline{C}\\&=A\,\overline{B}\,\overline{C}+\overline{A}B\,\overline{C}+\overline{A}BC\\&=m_4+m_2+m_3=\sum m(2,3,4)\end{aligned}$$

4.利用卡诺图化简逻辑函数

根据逻辑函数的最小项表达式求函数的卡诺图，只要将表达式中包含的最小项对应的方格内填 1，没有包含的最小项填 0（或不填），就可以得到卡诺图。

【例 1-22】 将 $Y(A,B,C)=\overline{AB+\overline{C}}+\overline{A}B\,\overline{C}+AC$ 用卡诺图（见图 1-8）表示。

解：利用摩根定律去掉非号，直到最后得到一个与或表达式，即：

$$\begin{aligned}Y(A,B,C)&=\overline{AB+\overline{C}}+\overline{A}B\,\overline{C}+AC\\&=\overline{AB}C+\overline{A}B\,\overline{C}+AC\\&=(\overline{A}+\overline{B})C+\overline{A}B\,\overline{C}+AC\\&=\overline{A}C+\overline{B}C+\overline{A}B\,\overline{C}+AC\end{aligned}$$

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

图 1-8

图形化简法的步骤：

(1)将逻辑函数写成最小项表达式。

(2)根据最小项表达式填卡诺图，凡式中包含的最小项，对应方格填 1，其余方格填 0。

(3)合并最小项，即将相邻的 1 方格围成一组，组成一包围圈，且每一组含 2^n 个方格；每

个包围圈写成一个新的乘积项。

(4)将所有包围圈对应的乘积项相加。

5.画包围圈时应遵循的原则

(1)包围圈内的方格数一定是 2^n 个。

(2)循环相邻特性包括上下底相邻、左右边相邻和四角相邻。

(3)同一方格可以被不同的包围圈重复包围多次,但新增的包围圈中一定要有原有包围圈未曾包围的方格。

(4)一个包围圈的方格数要尽可能多,包围圈的数目要可能少。

下面通过两个例子来学习卡诺图化简的方法

【例 1-23】 化简 $Y(A,B,C,D)=\sum m(0,1,2,3,4,5,8,10,11)$。

解:(1)画出函数的卡诺图,如图 1-9 所示。

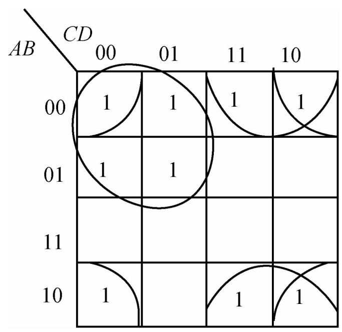


图 1-9 卡诺图

(2)按合并最小项的规律可画出三个圈。

(3)写成化简后的逻辑表达式。

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

【例 1-24】 化简 $Y(A,B,C,D)=\sum m(3,4,5,7,9,13,14,15)$

解: 画出函数的卡诺图,化简过程如图 1-10 所示。

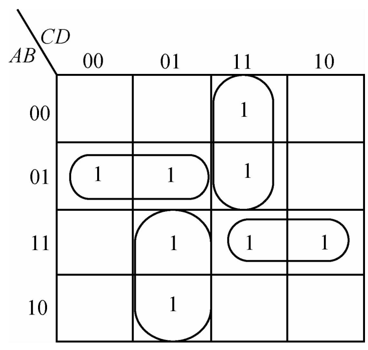


图 1-10 函数的化简过程

合并最小项得到的逻辑表达式为

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

在画卡诺图时应注意:

①卡诺图应按 2^n(n 是自然数)方格圈,圈包围 1 的个数越多越好,而圈数越少越好。

②圈中的 1 可以重复使用。

③每个圈至少有一个从来没被圈过的“1”,否则为多余圈。

1.6　具有无关项的逻辑函数的化简

1.6.1　无关项

在实际的逻辑问题中，有些变量的取值是不可能、不应该、不允许出现的，这些取值对应的最小项称为无关项，有时有称为约束项、任意项、禁止项，在卡诺图或真值表中用×来表示。

约束项的输出是任意的，可以认为是“1”，也可以认为是“0”。对于含有约束项的逻辑函数的化简，如果它对化简有利，则认为它是“1”；反之，则认为它为“0”。

逻辑函数中的约束项的表示方法如下：如果一个逻辑函数的约束项是 $\overline{A}\,\overline{B}\,\overline{C}$、$AB\,\overline{C}$、$B\,\overline{C}$、$ABC$，则可以写成下列等式：

$$\overline{A}\,\overline{B}\,\overline{C}+AB\,\overline{C}+\overline{A}\,\overline{B}\,\overline{C}+ABC=0 \quad 或 \quad \sum\nolimits_{m}(0,2,6,7)=0$$

1.6.2　具有约束项逻辑函数的化简步骤

(1)画出给定的逻辑函数的卡诺图。

(2)将约束项在卡诺图中相应的位置用“×”表示出来。

(3)根据需要将约束项作为“1”或“0”处理，以得到相邻最小项矩形组合最大为原则。

【例 1-25】　已知 $Y=\overline{A}\,\overline{B}\,\overline{C}+A\,\overline{C}D+\overline{A}\,\overline{B}\,\overline{C}D+\overline{A}BC\,\overline{D}$，约束条件为 $\overline{A}BD+CD=0$，求最简与或表达式。

(1)根据约束条件求约束项：

$$\overline{A}BD+CD=0$$

配项展开为

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

即：

$$\sum d(3,5,7,11,15)=0$$

(2)根据与或表达式和约束条件画出卡诺图，如图 1-11 所示。

(3)画出卡诺圈，约束项可以为“1”或者“0”。由图 1-11 可知，本题约束项全部为“1”，可得到最简与或表达式及其约束项如下：

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

$$\overline{A}BD+CD=0(约束条件)$$

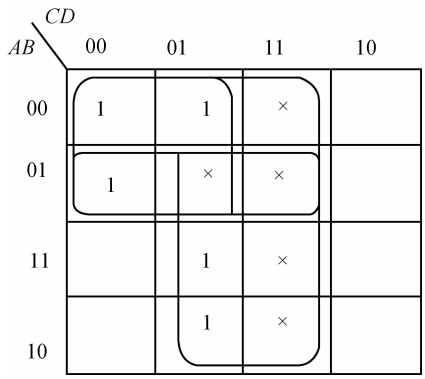


图 1-11

本章小结

本章介绍了常用数制、码制的基本概念，掌握常用数制（二、八、十、十六进制）及转换方法，了解常用二进制码、BCD 码（8421 码、5421 码、余 3 码）。掌握逻辑代数的基本概念、基本公式、基本规则，掌握逻辑函数的描述方式（真值表、逻辑表达式、电路图、卡诺图）及其相互转换方法，了解逻辑函数最简与或式的公式化简法，掌握逻辑函数（4 变量及以下）最简与或式的图形化简法。

常用的数制有二进制、八进制、十进制和十六进制等，它们之间的相互转化是本章学习的重点。

在数字系统中，逻辑运算是最基本，也是最常用的运算方法，因此在本章的学习中必须熟练地掌握逻辑代数的基本公式，灵活地运用逻辑代数的基本公式、规则及定律进行逻辑函数的化简。

常用逻辑函数的表示方法主要有 4 种，真值表、逻辑函数式、逻辑图和卡诺图，这 4 种表示方法之间可以相互转换，我们在选择是要根据具体情况选择最适当的一种表示方法。

逻辑函数的化简是本章的重点学习内容，本章共介绍了两种化简方法：公式化简法和图形化简法。

公式化简法，就是运用逻辑函数的基本公式及其推导公式对逻辑函数进行化简，其优点在于不受任何条件限制，但是由于没有固定的步骤可循，所以在化简一些比较复杂的逻辑函数时，不仅需要学生熟练地掌握和灵活地运用各种基本公式和定理，而且还需要一定的经验和运算技巧，这就要求学生通过多做习题来深化对逻辑代数基本公式及定理的理解和掌握。

图形化简法的优点在于简单、直观，而且有一定的化简步骤可循，是初学者容易掌握的化简方法。但是，当逻辑变量超过 5 个时就失去了其简单、直观的优点了。

本章习题

1.将下列二进制数转换为十进制数。

(1)$(1101101)_2$　　(2)$(111010111001)_2$

(3)$(1011.01)_2$　　(4)$(10100.001)_2$

2.将下列二进制数转换成十六进制数和八进制数。

(1)$(100100.011)_2$　　(2)$(1011101)_2$

3.将下列十进制数转换为二进制数。

(1)$(47)_{10}$　　(2)$(64)_{10}$

(3)$(101.25)_{10}$　　(4)$(125.37)_{10}$

4.将下列十六进制数为二进制数和十进制数。

(1)$(45)_{16}$　　(2)$(3AB.D9)_{16}$

(3)$(1ED.C8)_{16}$　　(4)$(EF.2C)_{16}$

5.将下列8421码转换成十进制数。

(1)$(1000.00010010)_{8421}$

(2)$(100100111000)_{8421}$

6.将下列十进制数转换成8421码。

(1)$(97)_{10}$　　(2)$(236)_{10}$

(3)$(72.68)_{10}$　　(4)$(81.39)_{10}$

7.用真值表证明下列恒等式。

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

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

8.按要求写出下列式子。

(1)$Y=AB+CDE+0$ 的对偶式

(2)$Y=ABC+AD+C$ 的对偶式

(3)$Y=AB+CDE+0$ 的反函数

(4)$Y=AB+C+AD$ 的反函数

9.填空题

(1)逻辑代数中的三种基本逻辑运算有________、________、________。

(2)逻辑代数的三种基本运算规则为________、________、________。

(3)逻辑函数的表示方法有________、________、________、________、________等。

(4)最基本的逻辑运算门电路有________、________、________三种。

10.用反演规则求下列函数的反函数。

(1)$F=(A+\overline{B})\cdot\overline{C+\overline{D}}$　　　　(2)$F=A\cdot\overline{B+\overline{C}}+\overline{A}D$

11. 写出下列函数的对偶式 F' 及反演式的函数表达式。

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

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

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

12. 用代数法化简下列函数。

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

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

13. 用公式法化简下列函数。

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

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

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

14. 用卡诺图法将下列函数化为最简与一或表达式。

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

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

(3)$F(A,B,C,D)=\sum m(2,4,5,6,7,11,12,14,15)$

(4)$F(A,B,C,D)=\sum m(0,1,2,5,8,10,11,12,13,14,15)$