# 项目1

# 认知逻辑函数与Multisim 10仿真

**知识目标**

① 理解数制与码制的概念。
② 掌握基本逻辑运算，会分析复合逻辑运算。
③ 掌握逻辑代数的基本定律、规则和常用公式。
④ 熟悉逻辑函数的化简方法。

**技能目标**

① 掌握数制之间的转换方法。
② 掌握公式法和卡诺图法化简逻辑函数。
③ 熟悉 Multisim 10 仿真软件的基本操作。
④ 掌握虚拟仪器仪表的使用方法。

## 1.1 任务1 认知数制与码制

### 1.1.1 数制

数制就是数的进位制，在日常生活中广泛应用的是十进制，在数字电路和计算机中使用的是二进制、八进制和十六进制等。

#### 1. 十进制

十进制是以 10 为基数的计数体制。在十进制中，有 0、1、2、3、4、5、6、7、8、9 共十个不同的数码，它的进位规律是逢十进一。在十进制数中，数码所处的位置不同，所代表的数值不同。如

$$(386.25)_{10}=3\times10^2+8\times10^1+6\times10^0+2\times10^{-1}+5\times10^{-2}$$

式中，$10^2$、$10^1$、$10^0$ 为整数部分百位、十位、个位的权，而 $10^{-1}$、$10^{-2}$ 为小数部分十分位、百分位的权，它们都是基数 10 的整数幂。

#### 2. 二进制

二进制是以 2 为基数的计数体制，在二进制中，只有 0 和 1 两个数码，它的进位规律是

逢二进一，各位权值是 2 的整数幂。如

$$(1\,101.11)_2=1\times2^3+1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-2}=(13.75)_{10}$$

可见，二进制数变为十进制数只需要按权展开相加即可。

### 3. 八进制

八进制是以 8 为基数的计数体制，在八进制中，有 0、1、2、3、4、5、6、7 共八个不同的数码，它的进位规律是逢八进一，各位权值为基数 8 的整数幂。如

$$\begin{aligned}(437.25)_8&=4\times8^2+3\times8^1+7\times8^0+2\times8^{-1}+5\times8^{-2}\\&=256+24+7+0.25+0.078\,125\\&=(287.328\,125)_{10}\end{aligned}$$

式中，$8^2$、$8^1$、$8^0$、$8^{-1}$、$8^{-2}$ 分别为八进制数各位的权。

### 4. 十六进制

十六进制是以 16 为基数的计数体制。在十六进制中，有 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F 十六个不同的数码，其中 A、B、C、D、E、F 分别代表 10、11、12、13、14、15。它们的进位规律是逢十六进一。各位权值为 16 的整数幂。如

$$(3A6.D)_{16}=3\times16^2+10\times16^1+6\times16^0+13\times16^{-1}$$

式中，$16^2$、$16^1$、$16^0$、$16^{-1}$ 分别为十六进制数各位的权。

表 1-1 中列出了二进制、八进制、十进制、十六进制几种不同数制的对照表。

**表 1-1　几种不同数制的对照表**

| 十进制 | 二进制 | 八进制 | 十六进制 | 十进制 | 二进制 | 八进制 | 十六进制 |
|---|---|---|---|---|---|---|---|
| 0 | 0000 | 0 | 0 | 8 | 1000 | 10 | 8 |
| 1 | 0001 | 1 | 1 | 9 | 1001 | 11 | 9 |
| 2 | 0010 | 2 | 2 | 10 | 1010 | 12 | A |
| 3 | 0011 | 3 | 3 | 11 | 1011 | 13 | B |
| 4 | 0100 | 4 | 4 | 12 | 1100 | 14 | C |
| 5 | 0101 | 5 | 5 | 13 | 1101 | 15 | D |
| 6 | 0110 | 6 | 6 | 14 | 1110 | 16 | E |
| 7 | 0111 | 7 | 7 | 15 | 1111 | 17 | F |

## 1.1.2　不同数制间的转换

### 1. 非十进制数转换为十进制数

由二进制、八进制、十六进制数转换为十进制数，只要将它们按权展开，求各位数值之和，即可得到对应的十进制数。如

$$(1\,011.01)_2=1\times2^3+0\times2^2+1\times2^1+1\times2^0+0\times2^{-1}+1\times2^{-2}=8+2+1+0.25=(11.25)_{10}$$

$$(172.01)_8=1\times8^2+7\times8^1+2\times8^0+0\times8^{-1}+1\times8^{-1}=64+56+2+0.012\,5=(122.012\,5)_{10}$$

$$(8ED.C7)_{16}=8\times16^2+14\times16^1+13\times16^0+12\times16^{-1}+7\times16^{-2}=(2\,285.777\,3)_{10}$$

### 2. 十进制数转换成非十进制数

十进制数转换为非十进制数时，要将其整数部分和小数部分分别转换，结果合并为目的数制形式。

（1）整数部分的转换

整数部分的转换方法是采用连续“除基取余”，一直除到商数为 0 为止。最先得到的余数为整数部分的最低位。

**【例 1-1】** 将$(25)_{10}$转换为二进制形式。

**解：**采用“除 2 取余”法

| | | | | |
|---|---|---|---|---|
| 2 | 25 | ---- | 1 | 低位 |
| 2 | 12 | ---- | 0 | ↑ |
| 2 | 6 | ---- | 0 | |
| 2 | 3 | ---- | 1 | |
| 2 | 1 | ---- | 1 | 高位 |
| | 0 | | | |

所以

$$(25)_{10}=(11\ 001)_2$$

（2）小数转换的转换

方法是采用连续“乘基取整”，一直进行到乘积的小数部分为 0 或满足要求的精度为止。最先得到的整数为小数部分的最高位。

**【例 1-2】** 将$(0.437)_{10}$转换为二进制形式。

**解：**采用“乘 2 取整”法

| | | |
|---|---|---|
| $0.437\times2=0.874$ | 整数部分为 0 …………最高位 | |
| $0.874\times2=1.748$ | 整数部分为 1 | ↓ |
| $0.748\times2=1.496$ | 整数部分为 1 | |
| $0.496\times2=0.992$ | 整数部分为 0 | |
| $0.992\times2=1.984$ | 整数部分为 1…………最低位 | |

所以

$$(0.437)_{10}=(0.011\ 01)_2$$

如果一个十进制数既有整数部分又有小数部分，可将整数部分和小数部分分别按要求进行等值转换，然后合并就可得到结果。

**【例 1-3】** 将十进制数$(174.437)_{10}$转换为八进制数和十六进制数。（保留小数点后 5 位）

**解：**① 整数部分采用“乘基取余”法，它们的基数分别为 8 和 16。

| | | | |
|---|---|---|---|
| 8 | 174 | ------ | 6 |
| 8 | 21 | ------ | 5 |
| 8 | 2 | ------ | 2 |
| | 0 | | |

| | | | |
|---|---|---|---|
| 16 | 174 | ------ | E |
| 16 | 10 | ------ | A |
| | 0 | | |

所以

$$(174)_{10}=(256)_8=(AE)_{16}$$

② 小数部分采用“乘基取整”法

$0.437\times8=3.496$　　　　整数部分为 3

$$0.496\times8=3.968 \qquad 整数部分为 3$$
$$0.968\times8=7.744 \qquad 整数部分为 7$$
$$0.744\times8=5.952 \qquad 整数部分为 5$$
$$0.952\times8=7.616 \qquad 整数部分为 7$$

所以

$$(0.437)_{10}=(0.337\ 57)_8$$

$$0.437\times16=6.992 \qquad 整数部分为\ 6$$
$$0.992\times16=15.872 \qquad 整数部分为\ F$$
$$0.872\times16=13.952 \qquad 整数部分为\ D$$
$$0.952\times16=15.232 \qquad 整数部分为\ F$$
$$0.232\times16=3.712 \qquad 整数部分为\ 3$$

所以

$$(0.437)_{10}=(0.6FDF3)_{16}$$

由此可得

$$(174.437)_{10}=(256.33757)_8=(AE.6FDF3)_{16}$$

**3. 二进制与八进制、十六进制间相互转换**

（1）二进制转换为八进制、十六进制

二进制数转换成八进制数（或十六进制数）时，其整数部分和小数部分可以同时进行转换。其方法是：以二进制数的小数点为起点，分别向左、向右每三位（或四位）分一组。对于小数部分，最低位一组不足三位（或四位）时，必须在有效位右边补 0，使其足位；然后，把每一组二进制数转换成八进制（或十六进制）数，并保持原排序。对于整数部分，最高位一组不足位时，可在有效位的左边补 0，也可不补。

**【例题 1-4】** 将$(1011010111.10011)_2$转换为八进制和十六进制数。

**解：**$(\underline{001}\ \underline{011}\ \underline{010}\ \underline{111}.\underline{100}\ \underline{110})_2=(1\ 327.46)_8$

$(\underline{0010}\ \underline{1101}\ \underline{0111}.\underline{1001}\ \underline{1000})_2=(2D7.98)_{16}$

（2）八进制数或十六进制数转换成二进制数

八进制（或十六进制）数转换成二进制数时，只要把八进制（或十六进制）数的每一位数码分别转换成三位（或四位）的二进制数．并保持原排序即可。整数最高位一组左边的 0 及小数最低位一组右边的 0 可以省略。

**【例 1-5】** 将$(35.24)_8$，$(3AB.18)_{16}$转换为二进制形式。

**解：**$(35.24)_8=(\underline{011}\ \underline{101}.\ \underline{010}\ \underline{100})_2=(11101.0101)_2$

$(3AB.18)_{16}=(\underline{0011}\ \underline{1010}\ \underline{1011}.\ \underline{0001}\ \underline{1000})_2=(1110101011.00011)_2$

由上可见，非十进制数转换成十进制数可采用按权展开法，十进制数转换成二进制数时可采用基数乘除法，二进制数与八进制数、十六进制数转换时可采用分组转换法。两个非十进制数之间相互转换时，若它们满足 2 的 $n$ 次幂，则可通过二进制数来进行转换。

### 1.1.3 码制

在数字系统中，二进制代码常用来表示特定的信息。将若干个二进制代码 0 和 1 按一定

规则排列起来，表示某种特定含义的代码，称为二进制代码，或称二进制码。如用一定位数的二进制代码表示数字、文字和字符等。下面介绍几种数字电路中常用的二进制代码。

### 1. 二–十进制代码

将十进制数的 0～9 十个数字用二进制数表示的代码，称为二–十进制码，又称 BCD 码。

由于 4 位二进制数码有 16 种不同组合，而十进制数只用到其中的 10 种组合，因此二–十进制数代码有多种方案。表 1-2 给出了几种常用的二进制代码。

**表 1-2　几种常用的二进制代码**

| 十 进 制 数 | 8421 码 | 5421 码 | 2421 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 | 1000 | 1011 | 1000 |
| 6 | 0110 | 1001 | 1100 | 1001 |
| 7 | 0111 | 1010 | 1101 | 1010 |
| 8 | 1000 | 1011 | 1110 | 1011 |
| 9 | 1001 | 1100 | 1111 | 1100 |

若某种代码的每一位都有固定的“权值”，则称这种代码为有权代码；否则叫无权代码。所以，判断一种代码是否是有权代码，只须检验这种代码的每个码组的各位是否具有固定的权值。如果发现一种代码中至少有 1 个码组的权值不同，这种代码就是无权码。

（1）8421BCD 码

8421BCD 码是有权码，各位的权值分别为 8、4、2、1。虽然 8421BCD 码的权值与 4 位自然二进制码的权值相同，但二者是两种不同的代码。8421BCD 码只取用了 4 位自然二进制代码的前 10 种组合。

（2）5421BCD 码和 2421BCD 码

5421BCD 码和 242lBCD 码也是有权码，各位的权值分别为 5、4、2、1 和 2、4、2、1。用 4 位二进制数表示 1 位十进制数，每组代码各位加权系数的和为其表示的十进制数。

242lBCD 码是一种自补代码，所谓自补特性是指将任意一个十进制数符 D 的代码的各位取反，正好是与 9 互补（9−D）的那个十进制数符的代码。如将 4 的代码 0100 取反，得到的 1011 正好是 9−4=5 的代码。这种特性称为自补特性，具有自补特性的代码称为自补码。

（3）余 3BCD 码

余 3BCD 码是 862lBCD 码的每个码组加 3（0011）形成的。其中的 0 和 9，1 和 8，2 和 7，3 和 6，4 和 5，各对码组相加均为 1111，余 3BCD 码也是自补代码，简称余 3 码。余 3 码各位无固定权值，故属于无权码。

**【例 1-6】** 分别将十进制数$(753)_{10}$转换为 8421BCD 码、5421BCD 和余 3BCD 码。

**解：** $(753)_{10}=(011101010011)_{8421BCD}$

$(753)_{10}=(101010000011)_{5421BCD}$

$(753)_{10}=(101010000110)_{余3BCD}$

## 2. 可靠性代码

代码在形成和传输过程中难免要产生错误，为了使代码形成时不易出差错或出错时容易发现并校正，须采用可靠性编码。常用的可靠性编码有格雷码、奇偶校验码等，下面分别介绍。

（1）格雷码

格雷码是一种典型的循环码，属于无权码，它有许多形式（如余 3 循环码等）。循环码有两个特点：一个是相邻性，是指任意两个相邻代码仅有一位数码不同；另一个是循环性，是指首尾的两个代码也具有相邻性。因为格雷码的这些特性可以减少代码变化时产生的错误，所以它是一种可靠性较高的代码。在自动化控制中生产设备多采用格雷码，如光电话码器，它可将光电读取头和代码盘之间的位移转换成相应的代码，以控制机械运动的行程和速度。

使用二进制数虽然直观、简单，但对码盘的制作和安装要求十分严格，否则易出错。例如，当二进制码盘从 0111 变化为 1000 时，4 位二进制数码必须同时变化，若最高位光电转换稍微早一些，就会出现错码 1111，这是不允许的。而采用格雷码码盘时，从 0100 变化为 1100 只有最高位变化，从而有效避免了由于安装和制作误差所造成的错码。

十进制数 0～15 的 4 位二进制格雷码见表 l-3，显然它符合循环码的两个特点。

**表 1-3　4 位二进制格雷码**

| 十进制数 | 格雷码 | 十进制数 | 格雷码 |
|---|---|---|---|
| 0 | 0000 | 8 | 1100 |
| 1 | 0001 | 9 | 1101 |
| 2 | 0011 | 10 | 1111 |
| 3 | 0010 | 11 | 1110 |
| 4 | 0110 | 12 | 1010 |
| 5 | 0111 | 13 | 1011 |
| 6 | 0101 | 14 | 1001 |
| 7 | 0100 | 15 | 1000 |

（2）奇偶校验码

奇偶校验码是最简单的检错码，它能够检测出传输码组中的奇数个码元错误。

奇偶校验码的编码方法：在信息码组中增加 1 位奇偶校验位，使得增加校验位后的整个码组具有奇数个 l 或偶数个 l 的特点。如果每个码组中 1 的个数为奇数，则称为奇校验码；如果每个码组中 1 的个数为偶数，则称为偶校验码。

例如，十进制数 5 的 842lBCD 码 0101 增加校验位后，奇校验码是 10101，偶校验码是 00101，其中最高位分别为奇校验位 1 和偶校验位 0。

**思考与练习**

1-1-1　简述二进制、八进制、十六进制数如何转换成十进制数。

1-1-2　简述十进制数如何转成二进制、八进制、十六进制数。

1-1-3　什么是 BCD 码，什么是 8421BCD 码？

1-1-4　什么是格雷码？什么是奇偶校验码？为什么说它们是可靠性代码？

# 1.2　任务 2　逻辑运算的分析

在数字电路中，l 位二进制数码的 0 和 l 不仅可以表示数量的大小，而且还可以表示两种不同的逻辑状态。例如，可以用 l 和 0 分别表示一件事情的有和无，或者表示电路的通和断、电灯的亮和灭等。这种只有两种对立逻辑状态的逻辑关系称为二值逻辑。

所谓“逻辑”，就是指事物间的因果关系。当两个二进制数码表示不同的逻辑状态时，它们之间可以按照指定的某种因果关系进行推理运算，这种运算就称为逻辑运算。逻辑代数（又称布尔代数）是按一定的逻辑规律进行运算的代数，是分析和设计数字电路最基本的数学工具。逻辑代数虽然和普通代数一样也用字母表示变量，但逻辑代数中逻辑变量的取值只有 l 和 0 两个值，且 0 和 1 不表示数量的大小，只表示两种对立的逻辑状态。

在逻辑代数中，有三种基本逻辑运算关系：与逻辑运算、或逻辑运算和非逻辑运算。

## 1.2.1　基本逻辑运算

### 1. 与逻辑运算

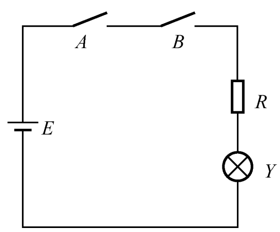



图 1-1　与逻辑电路示意图

当决定某一事件的所有条件都满足，该事件才发生，这种因果关系叫与逻辑关系，也称与运算或者逻辑乘。

与运算对应的逻辑电路可以用两个串联开关 $A$、$B$ 控制电灯 $Y$ 的亮和灭来示意，如图 1-1 所示。若用 1 代表开关闭合和灯亮。用 0 代表开关断开和灯灭，电路的功能可以描述为：只有当 $A$、$B$ 两个开关都闭合（$A$=l、$B$=1）时，电灯 $Y$ 才亮（$Y$=1），否则，灯就灭。这种灯的亮与灭和开关的通与断之间的逻辑关系就是与逻辑。其对应关系见表 1-4，这种表格叫真值表。

**表 1-4　与逻辑真值表**

| *A* | *B* | *Y* |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

所谓真值表，就是将输入变量的所有可能的取值组合对应的输出变量值一一列出来的表

格。若输入有 $n$ 个变量，则有 $2^n$ 种取值组合存在，输出对应地有 $2^n$ 个值。在逻辑分析中，真值表是描述逻辑功能的一种重要形式。

由真值表可以将与门电路的逻辑功能归纳为：“有 0 出 0，全 l 出 1”。

$Y$ 和 $A$、$B$ 间的关系可以用下式表示

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

此逻辑表达式读作“$Y$ 等于 $A$ 与 $B$”，为了简便，有时把符号“.”省掉，写成 $Y = AB$。

对于多变量的与运算可以用下式表示

$$Y = ABC \cdots$$

在数字电路中，常把能够实现与运算逻辑功能的电路叫与门，其逻辑符号如图 1-2 所示。

### 2. 或逻辑运算

决定某一事件的所有条件中，只要满足一个条件，则该事件就发生，这种因果关系称为或逻辑关系，也称或运算或者逻辑加。

或运算对应的逻辑电路可以用两个并联开关 $A$、$B$ 控制电灯 $Y$ 的亮和灭来示意，如图 1-3 所示。若仍用 1 代表开关闭合和灯亮，用 0 代表开关断开和灯灭，电路的功能可以描述为：只要 $A$、$B$ 两个开关中至少有一个闭合时，电灯 $Y$ 就亮；否则，灯就灭。其真值表见表 1-5。

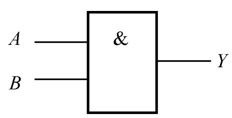



图 1-2　与门逻辑符号

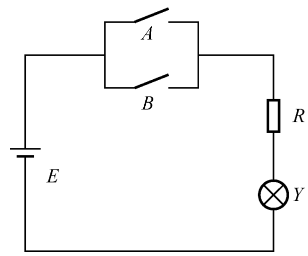



图 1-3　或逻辑电路示意图

**表 1-5　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

$$Y=A+B \tag{1-2}$$

对于多变量的或运算可用下式表示

$$Y = A + B + C + \cdots$$

在数字电路中，把能实现或运算的电路称为或门，其逻辑符号如图 1-4 所示。

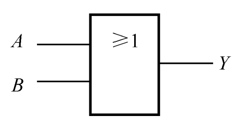



图 1-4　或门的逻辑符号

或门的逻辑功能可归纳为：“有 1 出 1，全 0 出 0”。

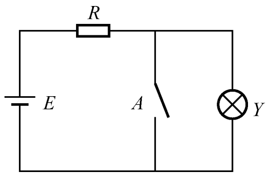



图 1-5　非门逻辑电路示意图

### 3. 非逻辑运算

非运算表示这样的逻辑关系，当某一条件具备了，事件便不会发生，而当此条件不具备时，事件一定发生。

非运算对应的逻辑关系可以用图 1-5 所示电路来示意。若仍用 1 代表开关闭合和灯亮，用 0 代表开关断开和灯灭，电路的功能可以描述为：若开关 $A$ 闭合，灯 $Y$ 就灭；反之，灯就亮。其真值表见表 1-6。

表 1-6　非逻辑真值表

| $A$ | $Y$ |
|---|---|
| 0 | 1 |
| 1 | 0 |

由该表可知，$Y$ 和 $A$ 之间的逻辑关系为：“有 0 出 1，有 1 出 0”。

$Y$ 和 $A$ 之间的关系可用下式表示

$$Y=\overline{A} \tag{1-3}$$

此逻辑表达式读作“$Y$ 等于 $A$ 非”。通常称 $A$ 为原变量，$\overline{A}$ 为反变量，二者共同称为互补变量。

在数字电路中，常把能完成非运算的电路叫非门或者反相器，非门只有一个输入端，其逻辑符号如图 l-6 所示。

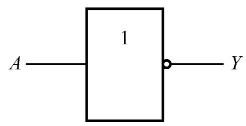



图 1-6　非门逻辑符号

## 1.2.2　复合逻辑运算

将与、或、非三种基本的逻辑运算进行组合，可以得到各种形式的复合逻辑运算，常见的复合运算有：与非运算、或非运算、与或非运算、异或运算、同或运算等。

当这三种基本逻辑运算组合同时出现在一个逻辑表达式中时，要注意三者的优先次序是：非、与、或。例如，逻辑函数 $Y=A\overline{B}+C$ 中，$B$ 变量先“非”，然后再和变量 $A$ 相“与”，相与的结果再和变量 $C$ 相“或”，最后得到 $Y$。

### 1. 与非逻辑运算

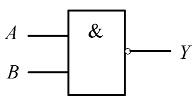



图 1-7　与非门逻辑符号

与非运算是与运算和非运算的复合运算。先进行与运算再进行非运算，其表达式为

$$Y=\overline{A\cdot B} \tag{1-4}$$

实现与非逻辑运算的电路叫与非门，其逻辑符号如图 1-7 所示。与非逻辑运算的真值表见表 1-7。

表 1-7　与非逻辑运算真值表

| $A$ | $B$ | $Y$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |

续表

| *A* | *B* | *Y* |
|---|---|---|
| 1 | 0 | 1 |
| 1 | 1 | 0 |

与非门的逻辑功能可归纳为：“有 0 出 1，全 1 出 0”。

实际应用的与非门的输入端可以有多个。

### 2. 或非逻辑运算

或非运算是或运算和非运算的复合运算。先进行或运算，后进行非运算，其表达式为

$$Y = \overline{A+B} \tag{1-5}$$

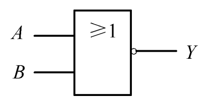



图 1-8　或非门逻辑符号

实现或非逻辑运算的电路叫或非门，其逻辑符号如图 l-8 所示。

或非逻辑运算的真值表见表 1-8。

**表 1-8　或非逻辑运算真值表**

| *A* | *B* | *Y* |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

或非门的逻辑功能可归纳为：“有 l 出 0，全 0 出 l”。

实际应用的或非门的输入端可以有多个。

### 3. 与或非逻辑运算

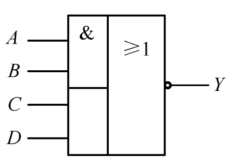



图 1-9　与或非门逻辑符号

与或非逻辑运算是与、或、非三种基本逻辑的复合运算。先进行与运算，再进行或运算，最后进行非运算，其表达式为

$$Y = \overline{AB+CD} \tag{1-6}$$

实现与或非逻辑运算的电路叫与或非门，其逻辑符号如图 1-9 所示。

与或非逻辑运算的真值表见表 1-9。

**表 1-9　与或非运算的真值表**

| 输　入 | | | | 输　出 |
|---|---|---|---|---|
| *A* | *B* | *C* | *D* | *Y* |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |

续表

| 输　入 | | | | 输　出 |
|---|---|---|---|---|
| *A* | *B* | *C* | *D* | *Y* |
| 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 |

与或非运算的逻辑功能是：只要 $A$、$B$ 或 $C$、$D$ 中有一组全为 1，输出就为 0，否则输出为 1。

**思考与练习**

1-2-1　三种基本逻辑运算分别是什么？写出它们的逻辑表达式并画出逻辑符号。
1-2-2　常用的复合逻辑运算有哪些？写出它们的逻辑表达式并画出逻辑符号。

# 1.3　任务 3　化简逻辑函数

## 1.3.1　逻辑运算的基本规律

根据基本逻辑运算规则和逻辑变量的取值只能是 0 和 1 的特点，可得出逻辑代数中的一些基本规律。

### 1. 基本运算公式

| | | |
|---|---|---|
| 0-1 律 | $A \cdot 0 = 0$ | $A + 1 = 1$ |
| 自等律 | $A \cdot 1 = A$ | $A + 0 = A$ |
| 重叠律 | $A \cdot A = A$ | $A + A = A$ |
| 互补率 | $A \cdot \overline{A} = 0$ | $A + \overline{A} = 1$ |
| 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 | $A \cdot (B + C) = AB + AC$ | $A + B \cdot C = (A + B)(A + C)$ |
| 吸收律 | $A(A + B) = A$ | $A + AB = A$ |

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

反演律（摩根定律）　$\overline{AB}=\overline{A}+\overline{B}$　　　　$\overline{A+B}=\overline{A}\cdot\overline{B}$

还原律　$\overline{\overline{A}}=A$

以上这些基本公式可以用真值表进行证明。例如，要证明反演律（也称摩根定理），可将变量 $A$、$B$ 的各种取值分别代入等式两边，其真值表见表 1-10。从真值表可以看出，等式两边的逻辑值完全对应相等，所以反演律成立。

**表 1-10　$\overline{A\cdot B}=\overline{A}+\overline{B}$ 的证明**

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

### 2. 逻辑代数运算规则

逻辑代数的运算优先顺序是：先算括号，再算非运算，然后是与运算，最后是或运算。逻辑代数运算的规则如下。

（1）代入规则

在逻辑等式中，如果将等式两边某一变量都代之以一个逻辑函数，则等式仍然成立。

例如，已知 $\overline{A\cdot B}=\overline{A}+\overline{B}$。若用 $Z=A\cdot C$ 代替等式中的 $A$，根据代入规则，等式仍然成立，即

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

（2）反演规则

已知函数 $Y$，欲求其反函数 $\overline{Y}$ 时，只要将 $Y$ 式中所有“·”换成“+”，“+”换成“·”，0 换成 1，1 换成 0，原变量换成其反变量，反变量换成其原变量，所得到的表达式就是 $\overline{Y}$ 的表达式。

利用反演规则可以比较容易地求出一个逻辑函数的反函数。

在变换过程中应注意，两个以上变量的公用的非号保持不变。运算的优先顺序为：先算括号，然后算逻辑乘，最后算逻辑加。

**【例 1-7】** 求 $Y=A+\overline{B+\overline{C}+\overline{D+\overline{E}}}+(G\cdot H)$ 的反函数

**解：** $\overline{Y}=\overline{A}\cdot\overline{\overline{B}\cdot C\cdot\overline{\overline{D}\cdot E}}\cdot(\overline{G}+\overline{H})$

（3）对偶规则

已知逻辑函数 $Y$，求它的对偶函数 $Y'$ 时，可通过将“·”变为“+”，“+”变为“·”，“0”换成“1”，“1”换成“0”得到。

若两个逻辑函数相等，则它们的对偶式也相等，若两个逻辑函数的对偶式相等，那么这两个逻辑函数也相等。

**【例 1-8】** 求 $Y=A\cdot B+\overline{A}C+B\cdot C$ 的对偶式。

**解：** $Y'=(A+B)\cdot(\overline{A}+C)\cdot(B+C)$

### 3. 逻辑函数的表示方法

逻辑函数的表示方法有逻辑表达式、真值表、卡诺图、逻辑图、波形图五种方法。

（1）逻辑表达式

用与、或、非等逻辑运算表示逻辑函数的各变量之间关系的代数式，称为逻辑表达式。例 $Y = A + B \cdot C$ 。

（2）真值表

前述中已经用到真值表，并给出了真值表的定义。在真值表中，每个输入变量只有 0 和 1 两种取值，$n$ 个变量就有 $2^n$ 个不同的取值组合，而每种组合都有对应的输出逻辑值。一个确定的逻辑函数只有一个逻辑真值表。当函数变量较多时，一般列出简化的特性真值表。

（3）卡诺图

如果把各种输入变量的取值组合下的输出函数值填入一种特殊的（按照逻辑相邻性划分的）方格图中，即得到了逻辑函数的卡诺图。

（4）逻辑图

用逻辑符号表示逻辑函数表达式中各个变量之间的运算关系，得到的电路图形，叫逻辑电路图，简称逻辑图。如 $Y = AB + BC$ 的逻辑图如图 1-10 所示。

（5）波形图

波形图是逻辑函数输入变量每一种可能出现的取值与对应的输出值按时间顺序依次排列的图形，也称时序图。

波形图可通过实验观察，在逻辑分析和一些计算机仿真软件中，常用这种方法分析结果。图 1-11 为逻辑函数 $Y = AB + BC$ 波形图。

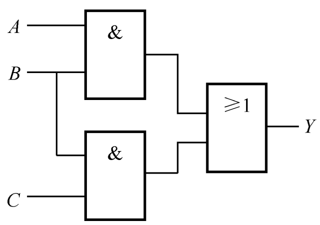



图 1-10　$Y = AB + BC$ 逻辑图

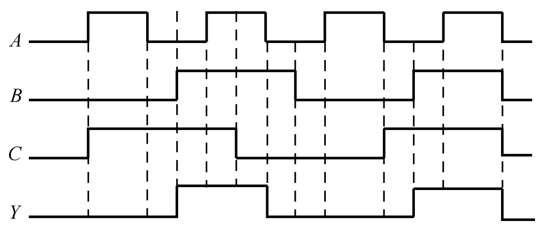



图 1-11　逻辑函数 $Y = AB + BC$ 波形图

逻辑函数的各种表示方法可以相互转换。根据真值表可以得到逻辑表达式，由逻辑表达式可以得到逻辑图，由逻辑图也可以反过来得到表达式。

### 4. 逻辑函数表达式

逻辑表达式越简单，实现它的电路也越简单，电路工作也较稳定可靠。

1）逻辑函数表达式的表示形式

一个逻辑函数的表达式可以有以下 5 种表示形式：

与−或表达式，例如 $Y = AB + \overline{B}C$

或−与表达式，例如 $Y = (A + C) \cdot (B + C)$

与非−与非表达式，例如 $Y = \overline{\overline{A\overline{B}} \cdot \overline{AC}}$

或非-或非表达式，例如 $Y = \overline{\overline{A+B} + \overline{A+\overline{C}}}$

与或非表达式，例如 $Y = \overline{A \cdot \overline{B} + A\overline{C}}$

利用逻辑代数的基本定律，可以实现上述 5 种逻辑函数表达式之间的变换。

2）逻辑函数的最简与或表达式

逻辑函数的最简与或表达式的特点：

① 乘积项个数最少。

② 每个乘积项中的变量个数也最少。

最简与或表达式的结果不是唯一的，可以从函数式的公式化简和卡诺图化简中得到验证。

3）逻辑函数的最小项表达式

（1）最小项的定义

在 $n$ 个变量的逻辑函数中，如乘积项中包含了全部变量，并且每个变量在该乘积项中以原变量或以反变量只出现一次，则该乘积项就定义为逻辑函数的最小项。$n$ 个变量的全部最小项共有 $2^n$ 个。

如三变量 $A$、$B$、$C$ 共有 $2^3$=8 个最小项：$\overline{A}\,\overline{B}\,\overline{C}$、$\overline{A}\,\overline{B}C$、$\overline{A}B\overline{C}$、$\overline{A}BC$、$A\overline{B}\,\overline{C}$、$A\overline{B}C$、$AB\overline{C}$、$ABC$。

（2）最小项的编号

为了书写方便，用 $m$ 表示最小项，其下标为最小项的编号。编号的方法是：最小项中的原变量取 1，反变量取 0，则最小项取值为一组二进制数，其对应的十进制数便为该最小项的编号。如三变量最小项 $\overline{A}B\overline{C}$ 对应的变量取值为 010，它对应的十进制数为 2，因此，最小项 $\overline{A}B\overline{C}$ 的编号为 $m_2$。其余最小项的编号依次类推。

（3）逻辑函数的最小项表达式

如一个与或逻辑表达式中的每一个与项都是最小项，则该逻辑表达式称为标准与或式，又称最小项表达式。任何一种形式的逻辑表达式都可以利用基本定律和配项法变换为标准与或式，并且标准与或式是唯一的。

**【例 1-9】** 将逻辑函数 $Y = AB + AC + BC$ 变换为最小项表达式。

**解：**（1）利用 $A + \overline{A} = 1$ 的形式作配项，补充缺少的变量：

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

② 利用 $A + A = 1$ 的形式合并相同的最小项：

$$\begin{aligned} Y &= AB\overline{C} + A\overline{B}C + \overline{A}BC + ABC \\ &= m_3 + m_5 + m_6 + m_7 \\ &= \sum m(3,5,6,7) \end{aligned}$$

## 1.3.2 公式法化简逻辑函数

运用逻辑代数的基本定律和公式对逻辑函数式进行化简的方法称为代数化简法，基本方法有以下几种。

### 1. 并项法

运用基本公式$A+\overline{A}=1$，将两项合并为一项，同时消去一个变量。如

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

### 2. 吸收法

运用吸收律$A+AB=A$和$AB+\overline{A}C+BC=AB+\overline{A}C$，消去多余项。如

① $Y=AB+AB(C+D)=AB$

② $$\begin{aligned}Y&=ABC+\overline{A}D+\overline{C}D+BD\\&=ABC+(\overline{A}+\overline{C})D+BD\\&=ABC+\overline{AC}D+BD\\&=ABC+\overline{AC}D\\&=ABC+\overline{A}D+\overline{C}D\end{aligned}$$

### 3. 消去法

利用$A+\overline{A}B=A+B$消去多余因子。如

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

### 4. 配项法

在不能直接运用公式、定律化简时，可通过乘$A+\overline{A}=1$或$A\cdot\overline{A}=0$进行配项后再化简。如

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

在实际化简逻辑函数时，需要灵活运用上述几种方法，才能得到最简与或表达式。

**【例 1-10】** 化简逻辑函数式$Y=AD+A\overline{D}+AB+\overline{A}C+\overline{C}D+A\overline{B}EF$

**解：**① 运用$A+\overline{A}=1$，将$AD+A\overline{D}$合并，得

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

② 运用$A+AB=A$，消去含有$A$因子的乘积项，得

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

③ 运用$A+\overline{A}B=A+B$，消去$\overline{A}C$中的$\overline{A}$，$\overline{C}D$中的$\overline{C}$，得

$$Y=A+C+D$$

## 1.3.3　卡诺图法化简逻辑函数

### 1. 相邻最小项

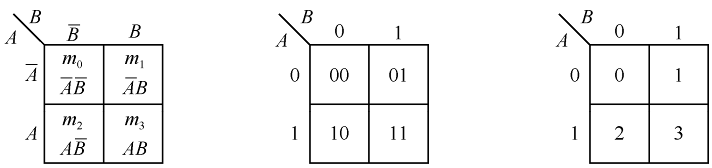
如果两个最小项中只有一个变量为互反变量，其余变量均相同时，则这两个最小项为逻辑相邻，并把它们称为相邻最小项，简称相邻项。例如，三变量最小项 $ABC$ 和 $AB\overline{C}$，其中的 $C$ 和 $\overline{C}$ 为互反变量，其余变量都相同，所以它们是相邻最小项。显然，两个相邻最小项可以合并为一项，同时消去互反变量，如 $ABC + AB\overline{C} = AB(C + \overline{C}) = AB$。合并结果为两个最小项的共有变量。

### 2. 卡诺图

卡诺图又称最小项方格图。用 $2^n$ 个小方格表示 $n$ 个变量的 $2^n$ 个最小项，并且使相邻最小项在几何位置上也相邻，按这样的相邻要求排列起来的方格图叫做 $n$ 个变量最小项卡诺图，这样的相邻原则又称卡诺图的相邻性。下面介绍 2～4 个变量最小项卡诺图的作法。

（1）二变量卡诺图

设两个变量为 $A$ 和 $B$，则全部 4 个最小项为 $\overline{A}\,\overline{B}$、$\overline{A}B$、$A\overline{B}$、$AB$，分别记为 $m_0$、$m_1$、$m_2$、$m_3$。按相邻性作出二变量卡诺图，如图 1-12 所示。

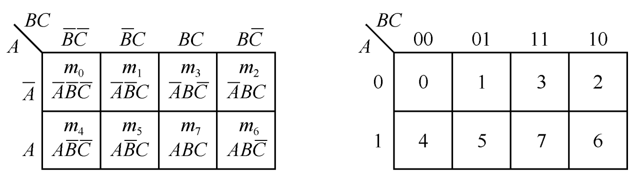



（a）方格内标最小项　（b）方格内标最小项取值　（c）方格内标最小项编号

图 1-12　二变量卡诺图

（2）三变量卡诺图

设三个变量为 $A$、$B$、$C$，全部最小项有 $2^3$=8 个，卡诺图由 8 个方格组成，按相邻性安放最小项可画出三变量卡诺图，如图 1-13 所示。




（a）方格内标最小项　（b）方格内标最小项编号

图 1-13　三变量卡诺图

应当注意，图中变量 $BC$ 的取值不是按自然二进制码（00、01、10、11）排列的，而是按格雷码（00、01、11、10）的顺序排列的，这样才能保证卡诺图中最小项在几何位置上相邻。

（3）四变量卡诺图

设四变量为 $A$、$B$、$C$、$D$，全部最小项有 $2^4$=16 个，卡诺图由 16 个方格组成，按相邻性

安放最小项可画出四变量卡诺图，如图 1-14 所示。

| $AB \backslash CD$ | $\overline{C}\overline{D}$ | $\overline{C}D$ | $CD$ | $C\overline{D}$ |
|---|---|---|---|---|
| $\overline{A}\overline{B}$ | $m_0$ | $m_1$ | $m_3$ | $m_2$ |
| $\overline{A}B$ | $m_4$ | $m_5$ | $m_7$ | $m_6$ |
| $AB$ | $m_{12}$ | $m_{13}$ | $m_{15}$ | $m_{14}$ |
| $A\overline{B}$ | $m_8$ | $m_9$ | $m_{11}$ | $m_{10}$ |

（a）方格内标最小项

| $AB \backslash CD$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 1 | 3 | 2 |
| 01 | 4 | 5 | 7 | 6 |
| 11 | 12 | 13 | 15 | 14 |
| 10 | 8 | 9 | 11 | 10 |

（b）方格内标最小项编号

图 1-14　四变量卡诺图

图 1-14 中的横向变量 $AB$ 和纵向变量 $CD$ 都是按格雷码顺序排列的，保证了最小项在卡诺图中的循环相邻性，即同一行最左方格与最右方格相邻，同一列最上方格和最下方格也相邻。

对于五变量及以上的卡诺图，由于复杂，在逻辑函数化简中很少使用，这里不再介绍。

### 3. 用卡诺图表示逻辑函数

在具体填写一个逻辑函数的卡诺图时，将逻辑函数表达式或其真值表所确定的最小项，在其对应卡诺图的小方格内填入函数值 1；表达式中没出现的最小项或真值表中函数值为 0 的最小项所对应的小方格内填入函数值 0。为了简明起见，小方格内的函数值为 0 时，常保留成空白，什么也不填。

**【例 1-11】** 画出逻辑函数 $Y(ABCD)=\sum m(0,1,4,5,6,9,12,13,15)$ 的卡诺图。

**解：**这是一个四变量的逻辑函数，首先要画出四变量卡诺图的一般形式，然后在最小项编号为 0，1，4，5，6，9，12，13，15 的小方格内填入 1，其余小方格内填入 0 或空着，即得到了该逻辑函数的卡诺图，如图 1-15 所示。

| $AB \backslash CD$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 1 |  |  |
| 01 | 1 | 1 |  | 1 |
| 11 | 1 | 1 | 1 |  |
| 10 |  | 1 |  |  |

图 1-15　例 1-11 图

**【例 1-12】** 画出逻辑函数 $Y=AB+BC+CA$ 的卡诺图。

**解：**首先将函数 $Y$ 写成标准与或式：

$$Y=AB+BC+CA=AB(C+\overline{C})+BC(A+\overline{A})+CA(B+\overline{B})$$
$$=\overline{A}BC+A\overline{B}C+AB\overline{C}+ABC$$
$$=\sum m(3,5,6,7)$$

再画出三变量卡诺图的一般形式，按照上例题同样的方法即可得到 $Y$ 的卡诺图，如图 1-16 所示。

| $A \backslash BC$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 |  |  | 1 |  |
| 1 |  | 1 | 1 | 1 |

图 1-16　例 1-12 图

### 4. 用卡诺图化简逻辑函数

用卡诺图化简逻辑函数的原理是利用卡诺图的相邻性，找出逻辑函数的相邻最小项加以合并，消去互反变量，以达到简化目的。

（1）最小项合并规律

① 只有相邻最小项才能合并。

② 两相邻最小项可以合并为一个与项，同时消去一个变量。四个相邻最小项合并为一个与项，同时消去两个变量。$2^n$ 个相邻最小项合并为一个与项，同时消去 $n$ 个变量。

③ 合并相邻最小项时，消去的是相邻最小项中互反变量，保留的是相邻最小项中的共有变量，并且合并的相邻最小项越多，消去的变量也越多，化简后的与项就越简单。

（2）用卡诺图化简逻辑函数的原则

用卡诺图化简逻辑函数画包围圈合并相邻项时，应注意以下原则：

① 每个包围圈内相邻 1 方格的个数一定是 $2^n$ 个方格，即只能按 1、2、4、8、16 个 1 方格的数目画包围圈。

② 同一个 1 方格可以被不同的包围圈重复包围多次，但新增的包围圈中必须有原先没有被圈过的 1 方格。

③ 包围圈中的相邻 1 方格的个数尽量多，这样可消去的变量多。

④ 包围圈的个数尽量少，这样得到的逻辑函数的与项少。

⑤ 注意卡诺图的循环邻接特性。同一行最左与最右方格中的最小项相邻，同一列的最上与最下方格中的最小项相邻。

**【例 1-13】** 试用卡诺图化简逻辑函数 $Y(ABCD)=\sum m(0,1,5,6,9,11,12,13,15)$。

**解：**① 画出卡诺图如图 1-17 所示。

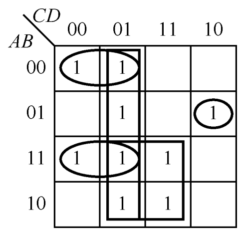



图 1-17　例 1-13 图

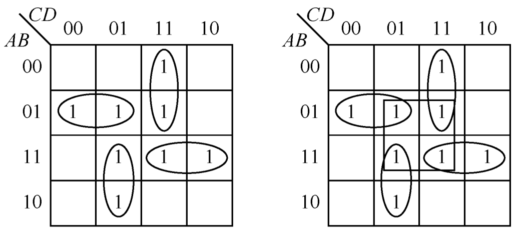
② 化简卡诺图。化简卡诺图时，一般先圈独立的 1 方格，再圈仅两个相邻的 1 方格，再圈仅 4 个相邻的 1 方格，依次类推。可得图 1-17。

③ 合并包围圈的最小项，写出最简与或表达式。

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

**【例 1-14】** 试用卡诺图化简逻辑函数 $Y=\overline{A}\,\overline{B}CD+\overline{A}B\overline{C}\,\overline{D}+A\overline{C}D+ABC+BD$。

**解：**① 画逻辑函数卡诺图，如图 1-18 所示。




（a）正确圈法　（b）不正确圈法

图 1-18　例 1-14 图

② 合并相邻最小项。注意由少到多画包围圈。

③ 写出逻辑函数的最简与或表达式：

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

如在该例题中先圈 4 个相邻的 1 方格，再圈两个相邻的 1 方格，便会多出一个包围圈，如图 1-18（b）所示，这样就不能得到最简与或表达式。