第1章 数字逻辑基础

两类信号：模拟信号;数字信号.

在时间上和幅值上均连续的信号称为模拟信号;

在时间上和幅值上均离散的信号称为数字信号.

处理数字信号的电路称为数字电路.

1.1 数制与数制转换

所谓“数制”，指进位计数制，即用进位的方法来计数.

数制包括计数符号（数码）和进位规则两个方面。

常用数制有十进制、十二进制、十六进制、六十进制等。

1.1.1 十进制

(1) 计数符号：0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

(2)进位规则：逢十进一

例： $1987.45=1\times10^3+9\times10^2+8\times10^1+7\times10^0+4\times10^{-1}+5\times10^{-2}$

(3) 十进制数按权展开式

$$(N)_{10} = \sum_{i=-m}^{n-1} a_i \times 10^i$$

系数 权

1.1.2. 二进制

(1) 计数符号：0, 1 .

(2)进位规则：逢二进一.

(3) 二进制数按权展开式

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

数字电路中采用二进制的原因：

1）数字装置简单可靠；

2）二进制数运算规则简单；

3）数字电路既可以进行算术运算，也可以进行逻辑运算.

1.1.3. 十六进制和八进制

十六进制数计数符号：0, 1, . . , 9, **A,B,C,D,E,F**.

十六进制数进位规则：逢十六进一.

按权展开式：

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

例:

$$(6D.4B)_{16} = 6\times16^{1} + D\times16^{0} + 4\times16^{-1} + B\times16^{-2}$$
$$= 6\times16^{1} + 13\times16^{0} + 4\times16^{-1} + 11\times16^{-2}$$

八进制数计数符号：0, 1, . . . 6, 7。

八进制数进位规则：逢八进一。

按权展开式: $$(\mathbf{N})_8 = \sum_{i=-m}^{n-1} \mathbf{a}_i \times \mathbf{8}^i$$

例： $(63.45)_8 = 6\times 8^1 + 3\times 8^0 + 4\times 8^{-1} + 5\times 8^{-2}$

1.1.4 二进制数与十进制数之间的转换

(1) 二进制数转换为十进制数(按权展开法)

例： $\mathbf{(1011.101)_2 = 1\times 2^3 + 1\times 2^1 + 1\times 2^0 + 1\times 2^{-1} + 1\times 2^{-3}}$

$\mathbf{= 8 + 2 + 1 + 0.5 + 0.125}$

$=11.625$

（2）十进制数转换为二进制数(提取2的幂法)

例： $(45.5)_{10} = 32 + 8 + 4 + 1 + 0.5$

$= 1\times 2^{5} + 0\times 2^{4} + 1\times 2^{3} + 1\times 2^{2} + 0\times 2^{1} + 1\times 2^{0} + 1\times 2^{-1}$

$= (101101.1)_{2}$

- 数制转换还可以采用基数连乘、连除等方法.

1.1.5 二进制数与十六进制数及八进制之间的转换

1.2 几种简单的编码

1.2.1 二 - 十进制码 (BCD码)(**Binary Coded Decimal codes**)

用四位二进制代码来表示一位十进制数码,这样的代码称为二-十进制码,或BCD码.

四位二进制有16种不同的组合,可以在这16种代码中任选10种表示十进制数的10个不同符号,选择方法很多.选择方法不同,就能得到不同的编码形式.

常见的**BCD**码有8421码、5421码、2421码、余3码等。

常用BCD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

1 有权BCD编码：每位数码都有确定的位权的码，

例如：8421码、5421码、2421码.

如：5421码1011代表5+0+2+1=8;

2421码1100代表2+4+0+0=6.

* 5421BCD码和2421BCD码不唯一.

例：2421BCD码0110也可表示6

* 在表中：

① 8421BCD码和代表0~9的二进制数一一对应；

② 5421BCD码的前5个码和8421BCD码相同，后5个码在前5个码的基础上加1000构成，这样的码，前5个码和后5个码一一对应相同，仅高位不同；

③ 2421BCD码的前5个码和8421BCD码相同，后5个码以中心对称取反，这样的码称为自反代码.

例：

4→0100　　5→1011

0→0000　　9→1111

2 无权BCD码：每位数码无确定的位权，例如：余3码.

余3码的编码规律为：在8421BCD码上加0011,

例 6的余3码为： 0110+0011=1001

1.2.2 格雷码(Gray码)

格雷码为无权码,特点为：相邻两个代码之间仅有一位不同,其余各位均相同.具有这种特点的代码称为循环码,格雷码是循环码.

格雷码和四位二进制码之间的关系：

设四位二进制码为$B_3B_2B_1B_0$，格雷码为$R_3R_2R_1R_0$，

则

$$R_3=B_3$$
$$R_2=B_3\oplus B_2$$
$$R_1=B_2\oplus B_1$$
$$R_0=B_1\oplus B_0$$

其中 $\oplus$为异或运算符，其运算规则为：若两运算数相同，结果为“0”；两运算数不同，结果为“1”。而异或运算满足：

若： $A=B\oplus C$

则： $B=A\oplus C$，$C=A\oplus B$

构成Gray码的方法：

1、先列出二进制码，再转换成Gray码；

2、直接列出Gray码，方法：

对n位Gray码，首个码字为00…01,即前n位（有效位）为0，最后一位为附加的虚拟位。从首个码字开始，虚拟位按1、0、1、0…规律变化。而下一个码字所对应的有效位码是否变化，取决于前一个码字从对应位的下一位到虚拟位至是否为100…0，对有效位的最低位，则上个码字的虚拟位为1，就改变一次状态，即翻转一次。

1.2.3 奇偶校验码

具有检错能力的代码

原代码的基础上增加一个码位使代码中含有的1的个数均为奇数（称为奇校验）或偶数（称为偶校验），通过检查代码中含有的1的奇偶性来判别代码的合法性。

1.2.4 字符数字码

字符数字码能表示计算机键盘上能看到的各种符号和功能。

美国信息交换的标准代码（简称ASCII）是应用最为广泛的字符数字码。

1.3 算术运算

1.3.1 二进制加法

0+0=0
1+0=0+1=1
1+1=10
1+1+1=11

$$\begin{array}{r} 1001\ (9) \\ +1011\ (11) \\ \hline 10100\ (20) \end{array}$$

1.3.2　有符号数的表示方法

表示二进制数的方法有三种，即原码、反码和补码

A_6	A_5	A_4	A_3	A_2	A_1	A_0	
0	1	0	1	1	0	1	$=+45_{10}$

符号位(+)　　真实二进制数

B_6	B_5	B_4	B_3	B_2	B_1	B_0	
1	0	1	0	0	1	1	$=-45_{10}$

符号位(－)　　补码

用补码系统表示有符号数

1.3.3　补码系统中的加法

第一种情况：两个正数相加。

+9 →	0	1001	(被加数)
+4 →	0	0100	(加数)
	0	1101	(和=+13)

符号位

第二种情况：正数与一个比它小的负数相加

符号位

+9 →	0	1001	(被加数)
−4 →	1	1100	(加数)
$\cancel{1}$	0	0101	

这个进位忽略,结果为00101(和=+5)

第三种情况：正数与比它大的负数相加

$$
\begin{array}{rcl}
-9 \longrightarrow & 1 & 0111 \\
+4 \longrightarrow & 0 & 0100 \\
\hline
 & 1 & 1011 \quad (\text{和}= -5)
\end{array}
$$

负的符号位

第四种情况：两个负数相加

符号位

$$
\begin{array}{rcl}
-9 \longrightarrow & 1 & 0111 \\
-4 \longrightarrow & 1 & 1100 \\
\hline
\cancel{1} & 1 & 0011
\end{array}
$$

这个进位忽略,结果为10011(和=−13)

1. 4 逻辑代数中的逻辑运算

研究数字电路的基础为逻辑代数，由英国数学家**George Boole**在1847年提出的，逻辑代数也称布尔代数.

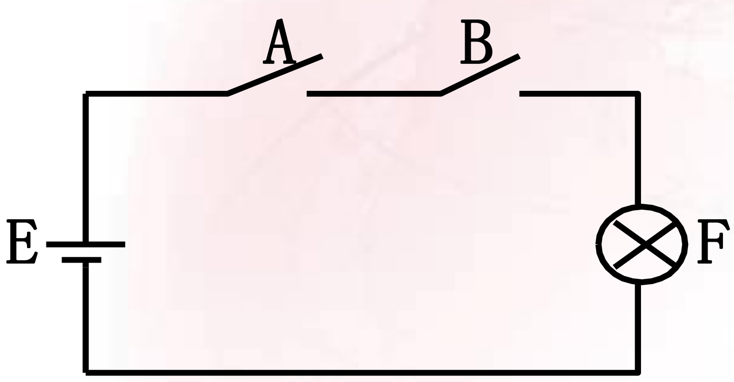
在逻辑代数中, 变量常用字母**A,B,C,……Y,Z, a,b, c,……x.y.z**等表示，变量的取值只能是“0”或“1”.

逻辑代数中只有三种基本逻辑运算, 即“与”、“或”、“非”。

1.4.1　基本逻辑运算

1.　与逻辑运算

定义：只有决定一事件的全部条件都具备时，这件事才成立；如果有一个或一个以上条件不具备，则这件事就不成立。这样的因果关系称为“与”逻辑关系。



与逻辑电路

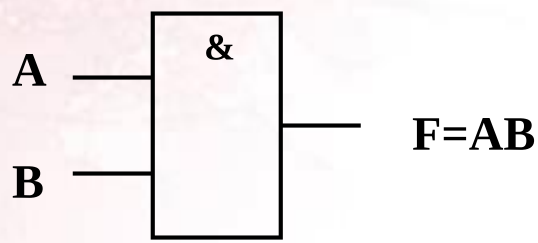
与逻辑电路状态表

开关A状态	开关 B状态	灯F状态
断	断	灭
断	合	灭
合	断	灭
合	合	亮

若将开关断开和灯的熄灭状态用逻辑量“0”表示；将开关合上和灯亮的状态用逻辑量“1”表示，则上述状态表可表示为：

与逻辑真值表

A	B	F=A · B
0	0	0
0	1	0
1	0	0
1	1	1



与门逻辑符号

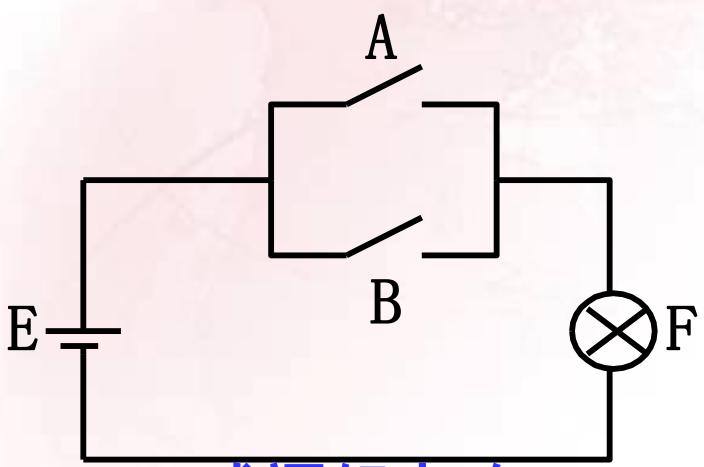
与门的逻辑功能概括：

1）有“0”出“0”；

2）全“1”出“1”。

2. 或逻辑运算

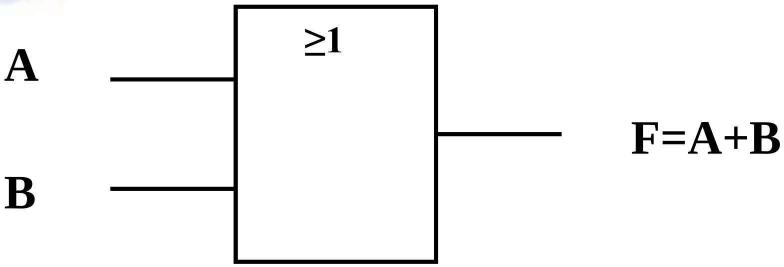
定义：在决定一事件的各种条件中, 只要有一个或一个以上条件具备时，这件事就成立; 只有所有的条件都不具备时, 这件事就不成立. 这样的因果关系称为“或”逻辑关系。



或逻辑电路

或逻辑真值表

A	B	F=A+ B
0	0	0
0	1	1
1	0	1
1	1	1

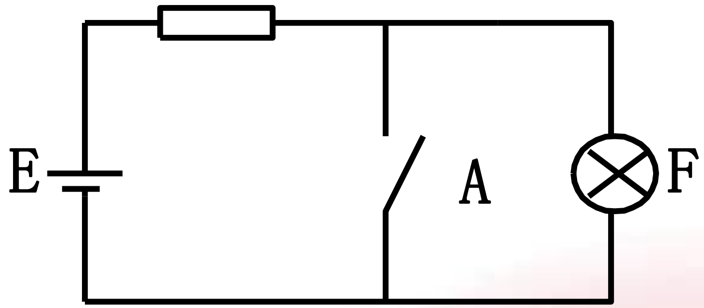


或门逻辑符号

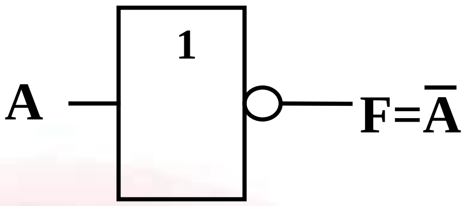
或门的逻辑功能概括为:
1）有“1”出“1”;
2）全“0” 出“0”.

3. 非逻辑运算

定义:假定事件F成立与否同条件A的具备与否有关,若A具备,则F不成立;若A不具备,则F成立. F和A之间的这种因果关系称为“非”逻辑关系.



非逻辑电路



非门逻辑符号

非逻辑真值表

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

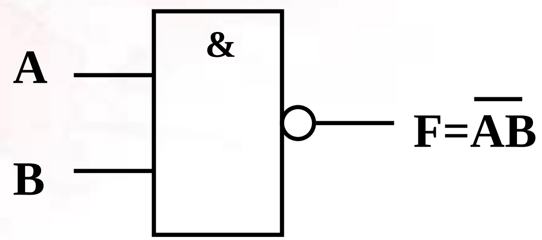
- 与门和或门均可以有多个输入端.

1.4.2 复合逻辑运算

1. 与非逻辑（将与逻辑和非逻辑组合而成）

与非逻辑真值表

A	B	$F=\overline{A\cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

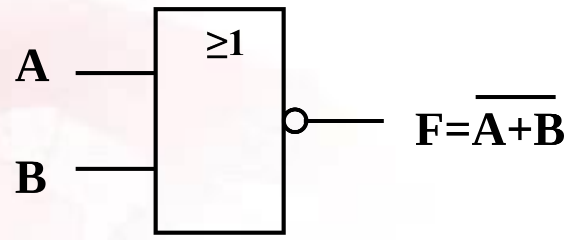


与非门逻辑符号

2. 或非逻辑（将或逻辑和非逻辑组合而成）

或非逻辑真值表

A	B	$F=\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0



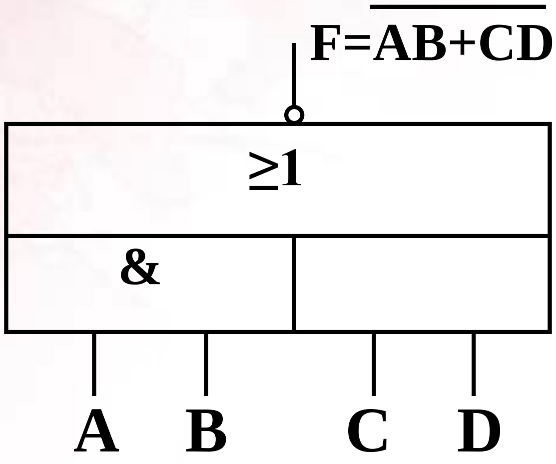
或非门逻辑符号

3. 与或非逻辑（由与、或、非三种逻辑组合而成）

与或非逻辑函数式：

$F=\overline{AB+CD}$

与或非门的逻辑符号

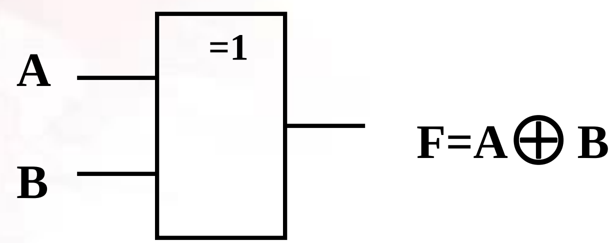


4. 异或逻辑

异或逻辑的函数式为： $F=\overline{A}B+A\overline{B} = A\oplus B$

异或逻辑真值表

A	B	$F=A\oplus B$
0	0	0
0	1	1
1	0	1
1	1	0



异或门逻辑符号

异或逻辑的功能为:

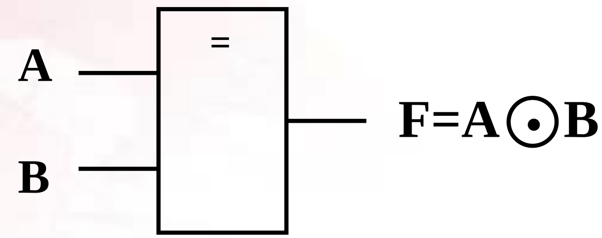
1） 相同得“0”；
2） 相异得“1”.

5. 同或逻辑

同或逻辑式为:$F = \bar{A}\,\bar{B} + A\,B = A \odot B$

同或逻辑 真值表

A	B	$F=A\odot B$
0	0	1
0	1	0
1	0	0
1	1	1



同或门逻辑符号

对照异或和同或逻辑真值表, 可以发现: 同或和异或互为反函数, 即:

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

- 表1.15给出了门电路的几种表示方法，本课程中，均采用“国标”。国外流行的电路符号常见于外文书籍中，特别在我国引进的一些计算机辅助分析和设计软件中，常使用这些符号。

1.4.3 正逻辑与负逻辑

门电路的输入、输出为二值信号,用“0”和“1”表示.这里的“0”、“1”一般用两个不同电平值来表示.

若用高电平V_H表示逻辑“1”,用低电平V_L表示逻辑“0”,则称为正逻辑约定,简称正逻辑;

若用高电平V_H表示逻辑“0”,用低电平V_L表示逻辑“1”,则称为负逻辑约定,简称负逻辑.

对一个特定的逻辑门，采用不同的逻辑表示时，其门的名称也就不同.

正负逻辑转换举例

电平真值表			正逻辑(与非门)			负逻辑(或非门)		
V_{i1}	V_{i2}	V_o	A	B	Y	A	B	Y
V_L	V_L	V_H	0	0	1	1	1	0
V_L	V_H	V_H	0	1	1	1	0	0
V_H	V_L	V_H	1	0	1	0	1	0
V_H	V_H	V_L	1	1	0	0	0	1

1.5 逻辑代数的基本定律和规则

1.5.1 逻辑函数的相等

设有两个逻辑：$F_1=f_1(A_1,A_2,\ldots,A_n)$

$F_2=f_2(A_1,A_2,\ldots,A_n)$

如果对于$A_1,A_2,\ldots,A_n$ 的任何一组取值（共2^n组），

F_1 和 F_2均相等，则称F_1和 F_2相等.

因此，如两个函数的真值表相等，则这两个函数一定相等.

1.5.2 基本定律

①0—1律　$A \cdot 0=0$ ；　$A+1=1$

②自等律　$A \cdot 1=A$ ；　$A+0=A$

③重迭律　$A \cdot A=A$ ；　$A+A=A$

④互补律　$A \cdot \overline{A}=0$ ；　$A+\overline{A}=1$

⑤交换律　$A \cdot B= B \cdot A$ ；　$A+B=B+A$

⑥结合律　$A(BC)=(AB)C$ ；　$A+(B+C)=(A+B)+C$

⑦分配律　$A(B+C)=AB+AC$ ；　$A+BC=(A+B)(A+C)$

⑧反演律　$\overline{A+B}=\overline{A} \cdot \overline{B}$ ；　$\overline{AB}=\overline{A}+\overline{B}$

⑨还原律　$\overline{\overline{A}}= A$

反演律也称德・摩根定理, 是一个非常有用的定理.

1.5.3 逻辑代数的三条规则

（1）代入规则

任何一个含有变量x的等式，如果将所有出现x的位置，都用一个逻辑函数式F代替，则等式仍然成立.

例： 已知等式 $\overline{A+B}=\overline{A}\cdot\overline{B}$,有函数式F=B+C, 则用F代替等式中的B,

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

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

由此可以证明反演定律对n变量仍然成立.

（2） 反演规则

设F为任意逻辑表达式, 若将F中所有运算符、常量及变量作如下变换:

·	+	0	1	原变量	反变量
↓	↓	↓	↓	↓	↓
+	·	1	0	反变量	原变量

则所得新的逻辑式即为F的反函数，记为$\overline{F}$。

例 已知 $F=\overline{A}\,B + A\,\overline{B}$, 根据上述规则可得:

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

例　已知　$F=A+\overline{B+\overline{C}+\overline{D+\overline{E}}}$，则

$$\overline{F}=\overline{A}\;\overline{\overline{B}\;C\;\overline{\overline{D}\;E}}$$

由F求反函数注意：

1）保持原式运算的优先次序；

2）原式中的不属于单变量上的非号不变；

(3) 对偶规则

设F为任意逻辑表达式,若将F中所有运算符和常量作如下变换:

$$
\begin{array}{cccc}
\cdot & + & 0 & 1 \\
\downarrow & \downarrow & \downarrow & \downarrow \\
+ & \cdot & 1 & 0
\end{array}
$$

则所得新的逻辑表达式即为F的对偶式,记为F'.

例　有$F=A\overline{B}+C\overline{D}$　　　$F'=(A+\overline{B})(C+\overline{D})$

例　有 $F=A+\overline{B+\overline{C}+\overline{D+\overline{\overline{E}}}}$　　　$F'=A\overline{B\overline{C}\,\overline{D\overline{E}}}$

对偶是相互的, F和F’互为对偶式. 求对偶式注意:

1） 保持原式运算的优先次序;

2）原式中的长短“非”号不变;

3）单变量的对偶式为自己。

对偶规则：若有两个逻辑表达式F和G相等，则各自的对偶式F’和G’也相等。

使用对偶规则可使得某些表达式的证明更加方便。

例 ：

已知 $A(B+C)=AB+AC \xrightarrow{\text{对偶关系}} A+BC=(A+B)(A+C)$

1.5.4 逻辑代数的常用公式

1）消去律 $AB+A\overline{B}=A$

证明：

$$AB+A\overline{B}=A(B+\overline{B})=A\cdot 1=A \xrightarrow{\text{对偶关系}} (A+B)(A+\overline{B})=A$$

2）吸收律1 $A+AB=A$

证明：

$$A+AB=A(1+B)=A\cdot 1=A \xrightarrow{\text{对偶关系}} A(A+B)=A$$

3）吸收律2　　$A+\overline{A}B=A+B$

证明：

$$A+\overline{A}B=(A+\overline{A})(A+B)=1\bullet(A+B) \xrightarrow{\text{对偶关系}} A(\overline{A}+B)=AB$$
$$=A+B$$

4）包含律　　$AB+\overline{A}C+BC=AB+\overline{A}C$

证明：

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

$\xrightarrow{\text{对偶关系}}$ $(A+B)(\overline{A}+C)(B+C)$
$=(A+B)(\overline{A}+C)$

5）关于异或和同或运算

对偶数个变量而言，

有 $A_1 \oplus A_2 \oplus \ldots \oplus A_n = \overline{A_1 \odot A_2 \odot \ldots \odot A_n}$

对奇数个变量而言，

有 $A_1 \oplus A_2 \oplus \ldots \oplus A_n = A_1 \odot A_2 \odot \ldots \odot A_n$

异或和同或的其他性质:

$A \oplus 0 = A$

$A \oplus 1 = \overline{A}$

$A \oplus A = 0$

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

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

$A \odot 1 = A$

$A \odot 0 = \overline{A}$

$A \odot A = 1$

$A \odot (B \odot C) = (A \odot B) \odot C$

$A + (B \odot C) = (A+B) \odot (A+C)$

利用异或门可实现数字信号的极性控制.

同或功能常由异或门实现.

1.6 逻辑函数的标准形式

1.6.1 常用的逻辑函数式

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

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

$=\overline{\overline{A\overline{B}}\cdot\overline{\overline{A}C}}$ 与非一与非式

$=\overline{\overline{A+C}+\overline{\overline{A}+\overline{B}}}$ 或非一或非式

$=\overline{AB+\overline{A}\,\overline{C}}$ 与或非式

1.6.2 函数的“与－或”式和“或－与”式

“与－或”式，指一个函数表达式中包含若干个与”项，这些“与”项的“或”表示这个函数。

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

“或－与”式，指一个函数表达式中包含若干个“或”项，这些“或”项的“与”表示这个函数。

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

1.6.3　最小项和最大项

1　最小项　（最小项是“与”项）

1）最小项特点

① n个变量构成的每个最小项，一定是包含n个因子的乘积项；

② 在各个最小项中，每个变量必须以原变量或反变量形式作为因子出现一次，而且仅出现一次。

例　有A、B两变量的最小项共有四项(2^2)：

$\bar{A}\bar{B}$　　$\bar{A}B$　　$A\bar{B}$　　AB

例　有A、B、C三变量的最小项共有八项(2^3)：

$\bar{A}\bar{B}\bar{C}$、$\bar{A}\bar{B}C$、$\bar{A}B\bar{C}$、$\bar{A}BC$、$A\bar{B}\bar{C}$、$A\bar{B}C$、$AB\bar{C}$、ABC

（2）　最小项编号

任一个最小项用 m_i 表示，m表示最小项，下标 i 为使该最小项为1的变量取值所对应的等效十进制数。

例 ：有最小项 $\bar{A}BC$, 要使该最小项为1，A、B、C的取值应为0、1、1，二进制数 011所等效的十进制数为 3，所以 $\bar{A}BC = m_3$

(3) 最小项的性质

① 变量任取一组值，仅有一个最小项为1，其他最小项为0；

② n变量的全体最小项之和为1；

③ 不同的最小项相与，结果为0；

④ 两最小项相邻，相邻最小项相“或”，可以合并成一项，并可以消去一个变量因子。

相邻的概念：两最小项如仅有一个变量因子不同，其他变量均相同，则称这两个最小项相邻.

相邻最小项相“或”的情况：

例： $ABC+AB\bar{C}=AB$

- 任一 n 变量的最小项，必定和其他 n 个不同最小项相邻。

2 最大项 (最大项是“或”项)

（1）最大项特点

①n个变量构成的每个最大项，一定是包含n个因子的“或”项；

② 在各个最大项中，每个变量必须以原变量或反变量形式作为因子出现一次，而且仅出现一次。

例　有A、B两变量的最大项共有四项：

$$\overline{A}+\overline{B} \qquad \overline{A}+B \qquad A+\overline{B} \qquad A+B$$

例　有A、B、C三变量的最大项共有八项：

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

(2) 最大项编号

任一个最大项用 M_i 表示，M表示最大项，下标 i 为使该最大项为0的变量取值所对应的等效十进制数。

例 ：有最大项 $\overline{A}+B+C$, 要使该最大项为0，A、B、C的取值应为1、0、0，二进制数 100所等效的十进制数为4，所以 $\overline{A}+B+C=M_4$

（3）最大项的性质

① 变量任取一组值，仅有一个最大项为0，其它最大项为1；

② n变量的全体最大项之积为0；

③ 不同的最大项相或，结果为 1；

④ 两相邻的最大项相“与”，可以合并成一项，并可以消去一个变量因子。

相邻的概念：两最大项如仅有一个变量因子不同，其他变量均相同，则称这两个最大项相邻。

相邻最大项相“与”的情况：

例： $(A+B+C)(A+B+\overline{C})=A+B$

- 任一 n 变量的最大项，必定和其他 n 个不同最大项相邻。

3 最小项和最大项的关系

编号下标相同的最小项和最大项互为反函数， 即

$$M_i = \overline{m_i} \qquad \text{或} \qquad m_i = \overline{M_i}$$

1.6.4 标准与或式和标准或与式

1 逻辑函数的标准与或式

最小项之和式为“与或”式，例：

$$F(A,B,C) = \overline{A}B\overline{C} + A\overline{B}\,\overline{C} + AB\overline{C}$$
$$= \Sigma m(2, 4, 6)$$
$$= \Sigma(2, 4, 6)$$

- 任一逻辑函数都可以表达为最小项之和的形式, 而且是唯一的.

例： $F(A,B,C) = AB + \overline{A}C$ 该式不是最小项之和形式

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

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

$= \Sigma m(1, 3, 6, 7)$

2 逻辑函数的标准或与式

逻辑函数的最大项之积的形式为“或与”式,

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

$$=\Pi M(0,2,4)$$

$$=\Pi(0,2,4)$$

- 任一逻辑函数都可以表达为最大项之积的形式, 而且是唯一的.

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

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

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

$=\Pi M(1,4,5,6)$

3 标准与或式和标准或与式的关系

若 $F=\Sigma m_i$ 则 $\overline{F}=\sum_{j\neq i} m_j$

$F=\overline{\sum_{j\neq i} m_j} = \prod_{j\neq i}\overline{m_j} = \prod_{j\neq i} M_j$

例 ： $F(A,B,C)=\Sigma(1,3,4,6,7)$

$=\Pi(0,2,5)$

1.7　逻辑函数式与真值表

真值表与逻辑表达式都是表示逻辑函数的方法。

1.7.1 由逻辑函数式列真值表

由逻辑函数式列真值表可采用三种方法，以例说明：

例：　试列出下列逻辑函数式的真值表。

F（A，B，C）=AB+BC

方法一：将A、B、C三变量的所有取值的组合（共八种），分别代入函数式，逐一算出函数值，填入真值表中。

方法二：先将函数式F表示为最小项之和的形式：

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

最后根据最小项的性质，在真值表中对应于ABC取值为011、110、111处填“1”，其它位置填“0”。

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

方法三：根据函数式F的含义，直接填表。

函数F=AB+BC表示的含义为：

1）当A和B同时为“1”（即AB=1）时，F=1

2）当B和C同时为“1”（即BC=1）时，F=1

3）当不满足上面两种情况时，F=0

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

方法三是一种较好的方法，要熟练掌握。

例： $F=\overline{(A\oplus B)(B\oplus C)}$

令： $F_1=(A\oplus B)$ ； $F_2=(B\oplus C)$

$\overline{F}=F_1F_2$

A	B	C	F_1	F_2	$\overline{F}$	F
0	0	0	0	0	0	1
0	0	1	0	1	0	1
0	1	0	1	1	1	0
0	1	1	1	0	0	1
1	0	0	1	0	0	1
1	0	1	1	1	1	0
1	1	0	0	1	0	1
1	1	1	0	0	0	1

1.7.2 由真值表写出逻辑函数式

根据最小项的性质，用观察法，可直接从真值表写出函数的最小项之和表达式。

例：已知函数F的真值表如下，求逻辑函数表达式。

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

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

解：由真值表可见，当ABC取011、101、110、111时，F为“1”。

所以，F由4个最小项组成：

$$F（A，B，C）=\Sigma m（3，5，6，7）$$

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

1.8 逻辑函数的化简

化简的意义： ①节省元器件，降低电路成本；

② 提高电路可靠性；

③ 减少连线，制作方便.

最简与或表达式的标准：

1） 所得与或表达式中，乘积项（与项）数目最少；

2） 每个乘积项中所含的变量数最少。

1.8.1 公式化简法

针对某一逻辑式,反复运用逻辑代数公式消去多余的乘积项和每个乘积项中多余的因子,使函数式符合最简标准.

化简中常用方法:

(1) 并项法　　利用公式　$AB+\bar{A}B=B$

例：　$F=A\bar{B}C+\bar{A}BC+ABC+\bar{A}\bar{B}C$

在化简中注意代入规则的使用

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

$=(A\oplus B)C+(A\odot B)C$

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

(2) 吸收法　　利用公式　$A+AB=A$

例：　$F=\bar{A}+\overline{A\overline{BC}}\,\overline{B+AC+\bar{D}}+BC$

$=(\bar{A}+BC)+(\bar{A}+BC)\overline{B+AC+\bar{D}}$

反演律

$=\bar{A}+BC$

（3）消项法　　利用公式　$AB+\bar{A}C+BC=AB+\bar{A}C$

例：

$$
\begin{aligned}
F&=A\bar{B}C\bar{D}+\bar{A}E+BE+C\bar{D}E\\
&=A\bar{B}C\bar{D}+(\bar{A}+B)E+C\bar{D}E\\
&=A\bar{B}C\bar{D}+\overline{A\bar{B}}E+C\bar{D}E\\
&=A\bar{B}C\bar{D}+(\bar{A}+B)E\\
&=A\bar{B}C\bar{D}+\bar{A}E+BE
\end{aligned}
$$

（4）消因子法　　利用公式　$A+\bar{A}B=A+B$

例： $F=AB+\overline{A}C+\overline{B}C$

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

$=AB+\overline{AB}C$

$=AB+C$

(5) 配项法 利用公式 $\overline{A}+A=1$ ； $A\cdot 1=A$ 等

例： $F=AB+\overline{A}C+BC$

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

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

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

$=AB+\overline{A}C$

1.8.2 卡诺图化简法

该方法是将逻辑函数用一种称为“卡诺图”的图形来表示, 然后在卡诺图上进行函数的化简的方法.

1 卡诺图的构成

卡诺图是一种包含一些小方块的几何图形, 图中每个小方块称为一个单元, 每个单元对应一个最小项. 两个相邻的最小项在卡诺图中也必须是相邻的. 卡诺图中相邻的含义:

① 几何相邻性, 即几何位置上相邻, 也就是左右紧挨着或者上下相接;

② 对称相邻性, 即图形中对称位置的单元是相邻的.

例 三变量卡诺图

循环码

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

相邻性规则

$$\begin{tikzcd} m_1 & m_3 \arrow[l] \arrow[r] \arrow[d] & m_2 \\ & m_7 & \end{tikzcd}$$

相邻性规则

$$\begin{tikzcd} m_2 & m_0 \arrow[l] \arrow[r] \arrow[d] & m_1 \\ & m_4 & \end{tikzcd}$$

（对称）

二、四、五变量卡诺图

A \ B	0	1
0	0	1
1	2	3

AB \ 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

相邻性规则

$$\begin{array}{ccccc} & & m_3 & & \\ & & \uparrow & & \\ m_5 & \leftarrow & m_7 & \rightarrow & m_6 \\ & & \downarrow & & \\ & & m_{15} & & \end{array}$$

AB \ CDE	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

2 逻辑函数的卡诺图表示法

用卡诺图表示逻辑函数，只是把各组变量值所对应的逻辑函数F的值，填在对应的小方格中。

（其实卡诺图是真值表的另一种画法）

例： F（A，B，C）$=\overline{A}BC+A\overline{B}C+ABC$

用卡诺图表示为:

A \ BC	00	01	11	10
0	0	0	1 m_3	0
1	0	1 m_5	1 m_7	0

3　在卡诺图上合并最小项的规则

当卡诺图中有最小项相邻时（即：有标1的方格相邻），可利用最小项相邻的性质，对最小项合并。

规则为：

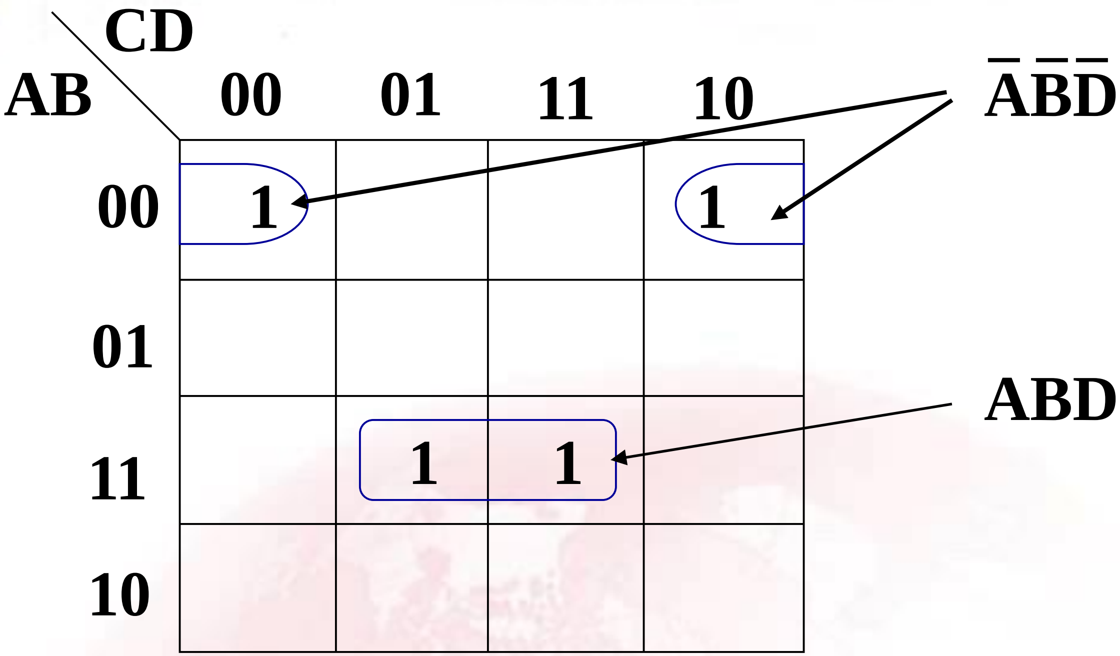
（1）　卡诺图上任何两个标1的方格相邻，可以合为1项，并可消去1个变量。

例：

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

$$\bar{A}BC+ABC=BC$$

$$A\bar{B}C+ABC=AC$$



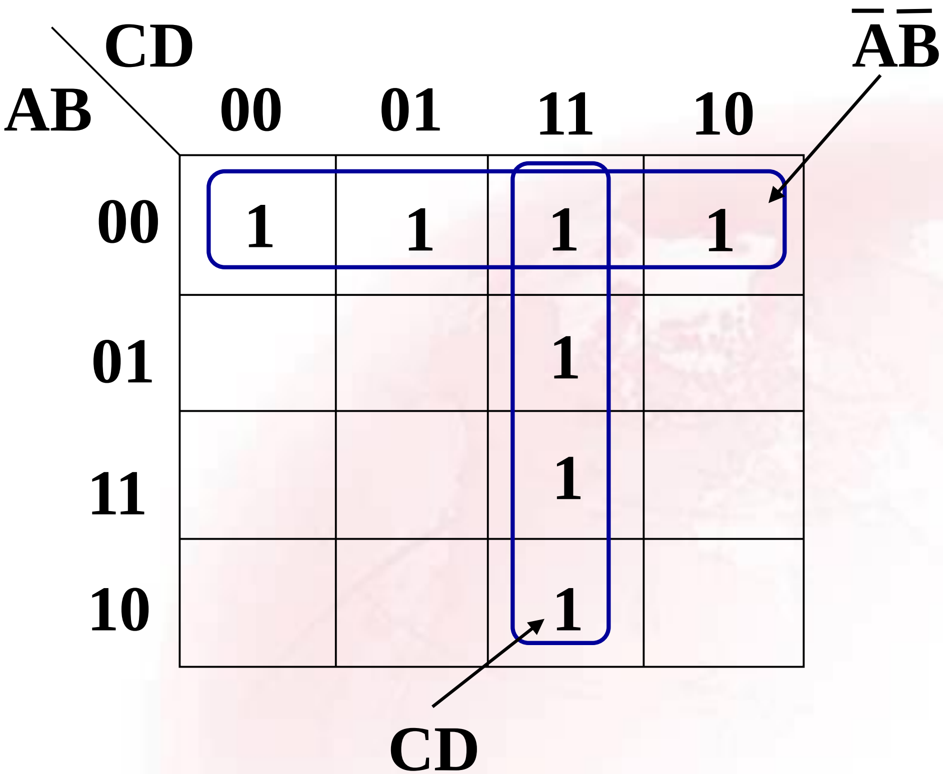
（2）卡诺图上任何四个标1方格相邻，可合并为一项，并可消去两个变量。

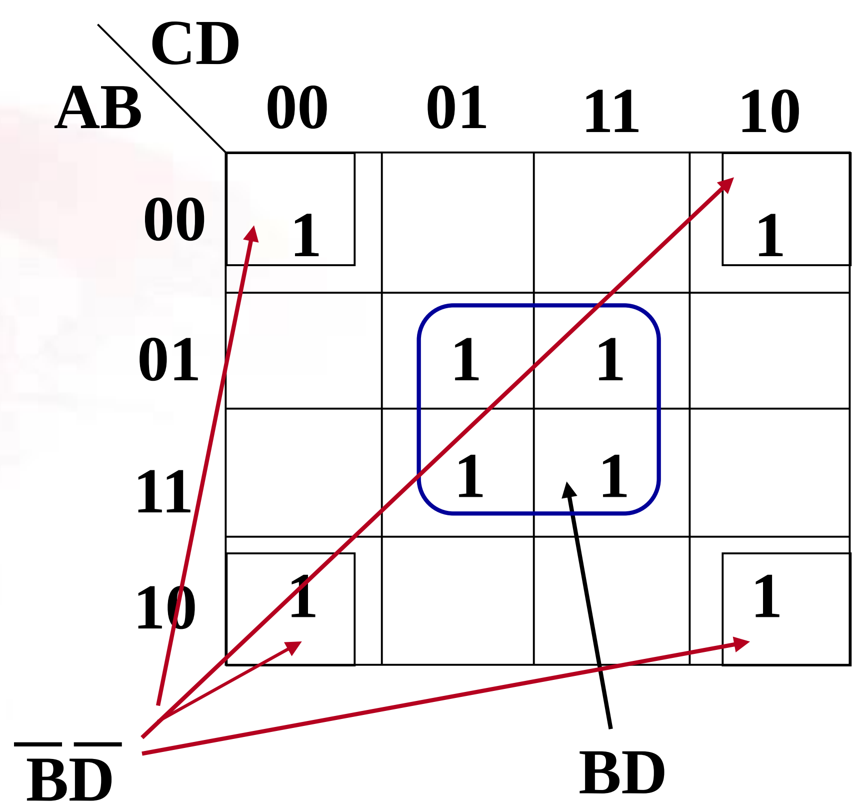
四个标1方格相邻的特点：

①同在一行或一列；　　　②同在一田字格中。

例：

同在一个田字格中





同在一行或一列

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

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

（3）卡诺图上任何八个标1的方格相邻，可以并为一项，并可消去三个变量。例：

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

$\overline{A}$

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

$\overline{B}$

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

综上所述，在n个变量的卡诺图中，只有2的 i 次方个相邻的标1方格（必须排列成方形格或矩形格的形状）才能圈在一起，合并为一项，该项保留了原来各项中n-i个相同的变量，消去i个不同变量。

4 用卡诺图化简逻辑函数（化为最简与或式）

①最简标准：项数最少，意味着卡诺图中圈数最少；
每项中的变量数最少，意味着卡诺图中的圈尽可能大。

例 将F（A，B，C）=Σm（3，4，5，6，7）化为最简与或式。

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

$F=A+BC$（最简）

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

$F=A\bar{B}+BC+AB\bar{C}$（非最简）

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

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

$=A\overline{BC}+BC$

$=A+BC\}$

② 化简步骤（结合举例说明）

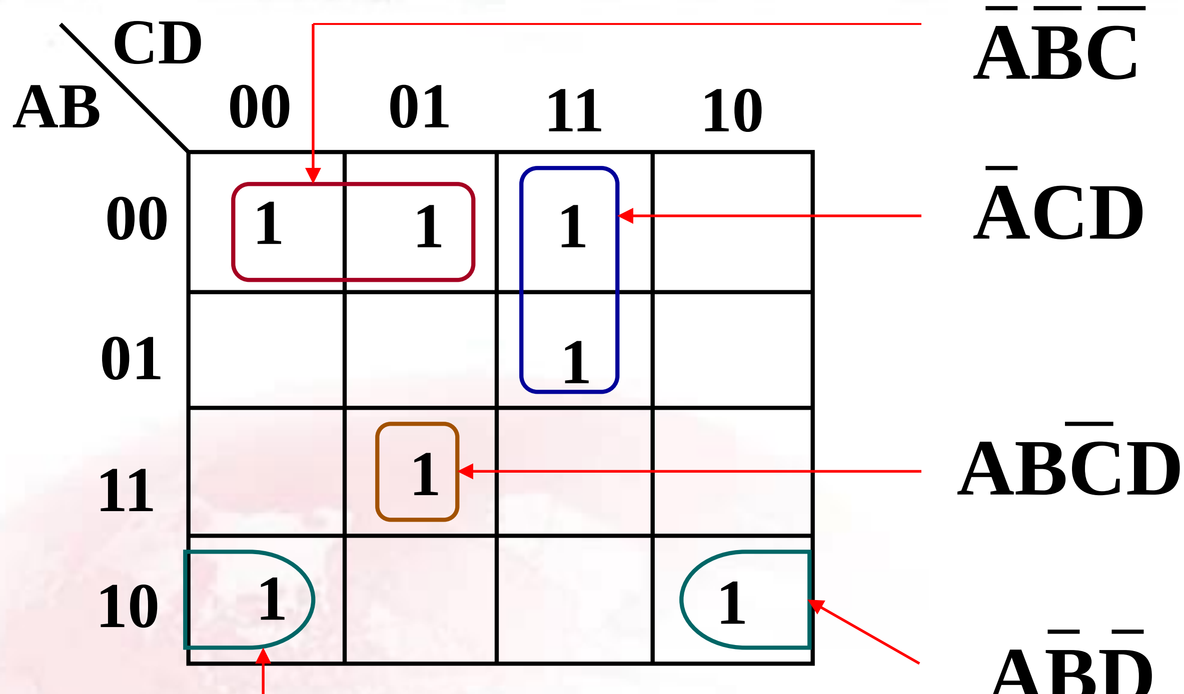
例　将F（A,B,C,D）=Σm（0,1,3,7,8,10,13）化为最简与或式。

解：（1）由表达式填卡诺图；

（2）圈出孤立的标1方格；

m_{13}

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



（3）找出只被一个最大的圈所覆盖的标1方格，并圈出覆盖该标1方格的最大圈；　m_7, m_{10}

（4）将剩余的相邻标1方格，圈成尽可能少，而且尽可能大的圈．m_0, m_1

（5）将各个对应的乘积项相加，写出最简与或式.

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

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

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

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

一种特殊情况：

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

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

$F=A\bar{B}+B\bar{C}+\bar{A}C$　　　$F=\bar{A}B+\bar{B}C+A\bar{C}$

得到两种化简结果，也都是最简的。

③ 化简中注意的问题

（1）每一个标1的方格必须至少被圈一次；

（2）每个圈中包含的相邻小方格数，必须为2的整数次幂；

（3）为了得到尽可能大的圈，圈与圈之间可以重叠；

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

（4）若某个圈中的所有标1方格, 已经完全被其它圈所覆盖, 则该圈为多余的.

例如:

蓝色的圈为多余的.

$$F=\bar{A}B\bar{C}+\bar{A}CD+A\bar{C}D+ABC+(BD)$$

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

④ 用卡诺图求反函数的最简与或式

方法:在卡诺图中合并标 0 方格, 可得到反函数的最简与或式.

例:

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

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

• 常利用该方法来求逻辑函数F的最简与或非式， 例如将上式F上 的非号移到右边, 就得到F的最简与或非表达式.

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

逻辑函数化简的技巧

- 对较为复杂的逻辑函数，可将函数分解成多个部分，先将每个部分分别填入各自的卡诺图中，然后通过卡诺图对应方格的运算，求出函数的卡诺图。
- 对卡诺图进行化简。

例：化简逻辑函数

$$F=(AB+\bar{A}C+B\bar{D})\oplus(A\bar{B}\bar{C}D+\bar{A}CD+BCD+\bar{B}C)$$

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

⊕

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

=

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

$$F=\bar{A}\bar{B}\bar{C}D+AB\bar{C}+BC\bar{D}+AC\bar{D}$$

1.8.3 不完全确定的逻辑函数及其化简

在某些实际数字电路中, 逻辑函数的输出只和一部分最小项有确定对应关系, 而和余下的最小项无关. 余下的最小项无论写入逻辑函数式还是不写入逻辑函数式, 都不影响电路的逻辑功能. 把这些最小项称为无关项. 用英文字母d(don’t care)表示，对应的函数值记为“×”。

包含无关项的逻辑函数称为不完全确定的逻辑函数.

利用不完全确定的逻辑函数中的无关项往往可以将函数化得更简单.

例：设计一个奇偶判别电路. 电路输入为8421**BCD**码, 当输入为偶数时, 输出为 0 ;当电路输入为奇数时, 输出为1 .

由于8421**BCD**码中无**1010~1111**这6个码, 电路禁止输入这6个码. 这6个码对应的最小项为无关项.

真值表

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

$$F(A,B,C,D)=\Sigma m(1,3,5,7,9)+\Sigma d(10\sim 15)$$

$$F(A,B,C,D)=\Sigma m(1,3,5,7,9)+\Sigma d(10\sim 15)$$

若不利用无关项（即将卡诺图中的×均作0处理），则化简结果为:

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

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

若利用无关项(即将卡诺图中的×按化简的需要任意处理, 将有些×当作0, 有些×当作1), 则化简结果为:

$$F(A,B,C,D)=D$$

完整地将函数写为:

$$\begin{cases} F(A,B,C,D)=D \\ \Sigma d(10\sim 15)=0 \end{cases}$$

例: $F=(A\oplus B)C\bar{D}+\bar{A}B\bar{C}+\bar{A}\bar{C}D$ 且AB+CD=0

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

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

注意:在无特殊说明的情况下,为使逻辑函数化的更简单,均应按上述第二种方法处理最小项.

1.8.4　逻辑函数式化简为其它形式

1 与非—与非式

由最简的与或式，经过两次求反，可得与非—与非式

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

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

$$= \overline{\overline{AB}\;\overline{\overline{A}C}}$$

2　与或非式

求出反函数 $\overline{F}$ 的最简与或式，再对 $\overline{F}$ 求反

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

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

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

3　　或与式

由最简的与或式，运用两次求对偶或两次求反可得或与式

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

利用反演规则，再对 $\overline{F}$ 求反

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

4　或非—或非式

最简的或与式，经过两次求反，可得或非—或非式

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

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

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

1.8.5 奎恩—麦克拉斯基化简法（Q—M法）

Q—M法有确定的流程，适用于任何复杂逻辑函数的化简

1．将函数化为最小项之和的形式，列出最小项编码表

2．按包含1的个数将最小项分组

3．合并相邻的最小项

4．选择最少的乘积项

5．最后确定化简结果中的乘积项

6)多输出逻辑函数的化简

实际的数字电路，常常是一个多输出电路，即对应于相同一组输入变量，存在多个输出函数。

多输出函数的化简也是以单个函数的化简方法为基础，但要考虑到整体电路最简。

例：
$$\begin{cases} F_1(A,B,C)=\Sigma m(1,4,5) \\ F_2(A,B,C)=\Sigma m(1,3,7) \end{cases}$$

若按单个函数化简方法

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

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

化简的结果为：

$$\begin{cases} F_1 = A\overline{B} + \overline{B}C \\ F_2 = \overline{A}C + BC \end{cases}$$

从整体出发，考虑函数的化简

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

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

化简的结果为：

$$\begin{cases} F_1 = \overline{A}\,\overline{B}C + A\overline{B} \\ F_2 = \overline{A}\,\overline{B}C + BC \end{cases}$$