

2.2 逻辑函数的代数法化简

2.2.1 逻辑函数的表达式及变换

2.2.2 逻辑函数的最简形式

2.2.3 逻辑函数的代数法化简

2.2.1 逻辑函数的表达式及变换

$$L = \bar{A} \bar{C} D + AC\bar{D} + AD$$

“与或” 式

$$L = AC + \bar{C}D$$

“与或” 式

$$L = (A + \bar{C})(C + D)$$

“或与” 式

$$L = \overline{\overline{A} \overline{C} \overline{C} \overline{D}}$$

“与非—与非” 式

$$L = \overline{\overline{(A + \bar{C})} + \overline{(\bar{C} + D)}}$$

“或非—或非” 式

$$L = \overline{\overline{A} C + \bar{C} \bar{D}}$$

“与或非” 式

2.2.1 逻辑函数的表达形式及变换

$$L = \bar{A}\bar{C}D + AC\bar{D} + AD$$

$$= (\bar{A}\bar{C} + A)D + A(C\bar{D} + D)$$

$$= (A + \bar{C})D + A(C + D)$$

$$= AD + \bar{C}D + AC = AC + \bar{C}D(\text{化简})$$

$$L = \overline{\overline{AC + \bar{C}D}} = \overline{\overline{AC}\overline{\bar{C}D}}(\text{与非一与非})$$

$$L' = (A + C)(\bar{C} + D) = A\bar{C} + AD + CD = A\bar{C} + CD$$

$$L = (A + \bar{C})(C + D)(\text{或与式})$$

$$L = \overline{\overline{(A + \bar{C})(C + D)}} = \overline{\overline{(A + \bar{C})} + \overline{(C + D)}}(\text{或非一或非式})$$

$$L = \overline{\overline{(A + \bar{C})} + \overline{(C + D)}} = \overline{\bar{A}C + \bar{C}\bar{D}}(\text{与或非式})$$

2.2.1 逻辑函数的表达形式及变换

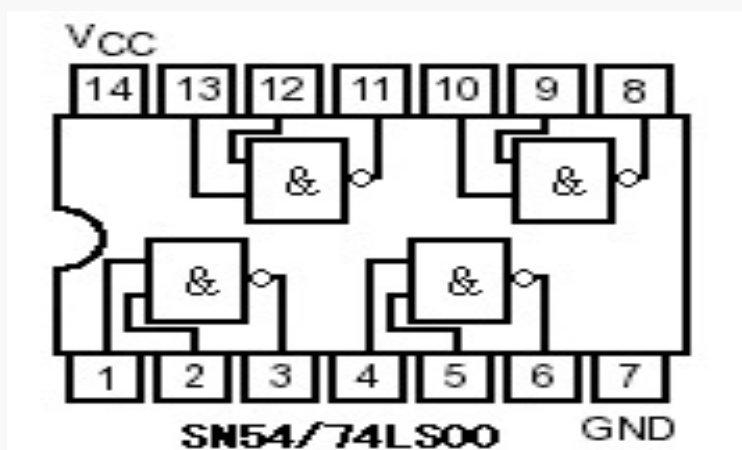
例：用与非门实现逻辑函数 $L_1 = AC + \overline{C}D$

将逻辑函数与或式变换与非-与非表达式

方法：将逻辑函数两次求反后用摩根定律

$$L_1 = AC + \overline{C}D = \overline{\overline{AC + \overline{C}D}} = \overline{\overline{AC} \overline{\overline{C}D}} = \overline{\overline{AC} \overline{CD}}$$

用与非门实现逻辑函数

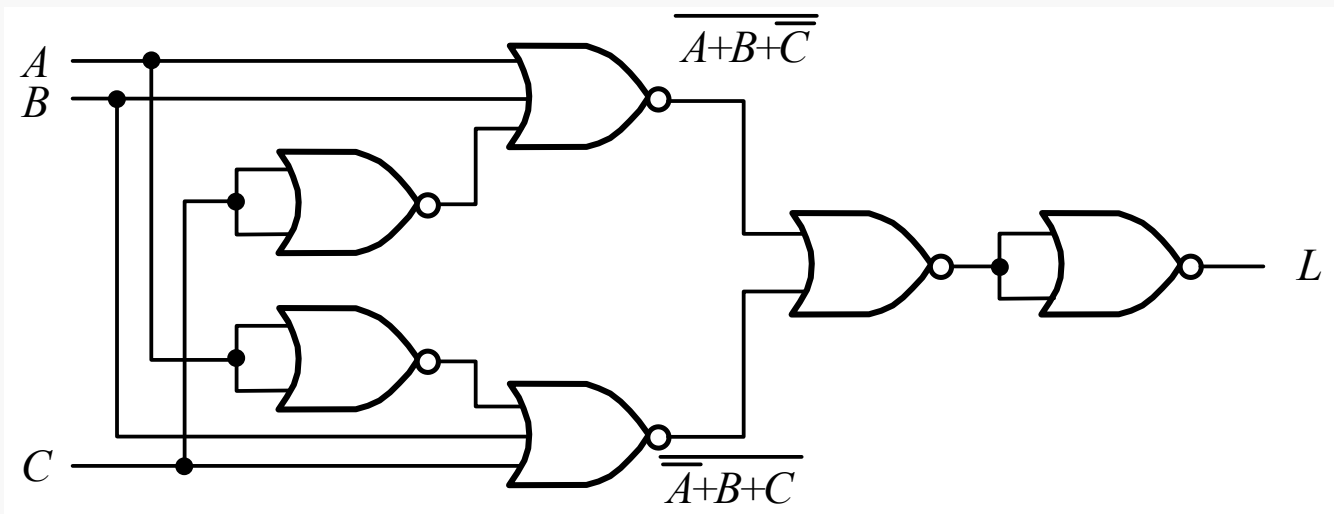


2.2.1 逻辑函数的表达形式及变换

例：对逻辑函数表达式 $L = \overline{A}\overline{B}C + A\overline{B}\overline{C}$

进行变换，仅用**或非门**画出该表达式的逻辑图。

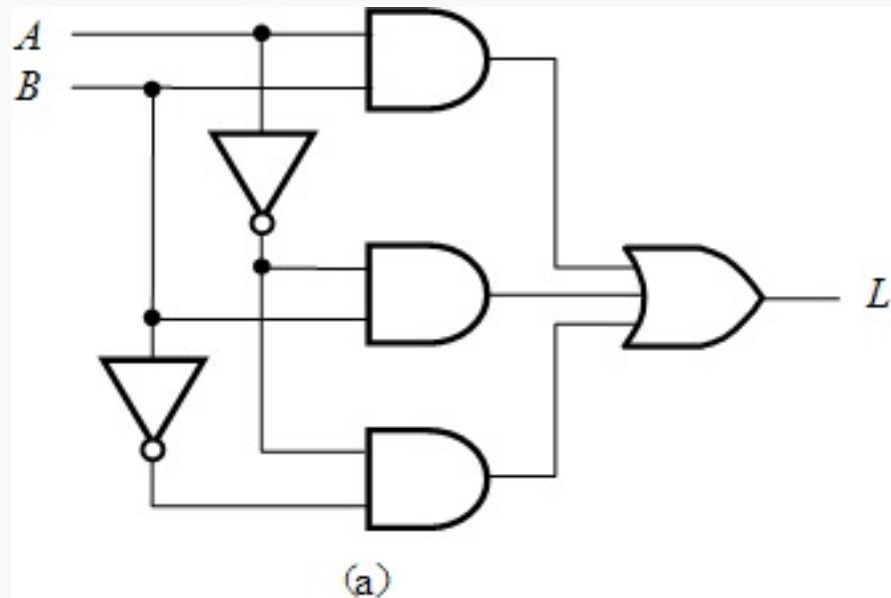
$$\begin{aligned}\text{解： } L &= \overline{A}\overline{B}C + A\overline{B}\overline{C} = \overline{\overline{\overline{A}\overline{B}C}} + \overline{\overline{\overline{A}\overline{B}\overline{C}}} \\ &= \overline{A + B + \overline{C}} + \overline{\overline{A} + B + C} \\ &= \overline{\overline{A + B + \overline{C}}} + \overline{\overline{\overline{A} + B + C}}\end{aligned}$$



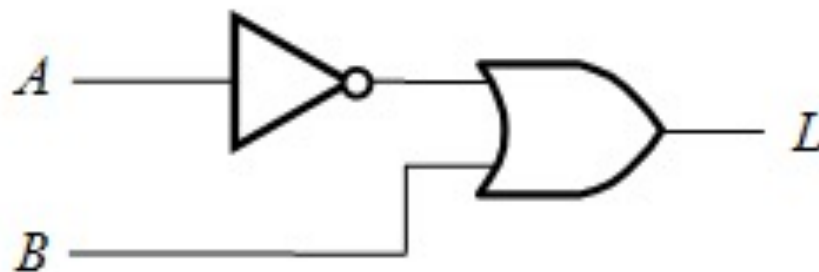
2.2.2 逻辑函数的最简形式——化简的意义

化简逻辑函数

$$\begin{aligned} L &= AB + \bar{A}B + \bar{A}\bar{B} \\ &= (A + \bar{A})B + \bar{A}\bar{B} \\ &= 1 \cdot B + \bar{A}\bar{B} \\ &= B + \bar{A} \end{aligned}$$



化简的意义：根据化简后的表达式构成的逻辑电路简单，可节省器件，减少连线，减小体积，降低成本，提高电路的可靠性。



2.2.2 逻辑函数的最简形式——化简的标准

简化标准（最简的与 - 或表达式）

- 1 . 乘积项的个数最少（与门的个数少）
- 2 . 每个乘积项中包含的变量数最少（与门的输入端个数少）

$$L = \bar{A} \bar{C} D + AC\bar{D} + AD$$

$$L = AC + \bar{C}D$$

化简的主要方法

- 1、公式法（代数法）
- 2、图解法（卡诺图法）

2.2.3 逻辑函数的代数法化简

运用逻辑代数的基本定律和恒等式进行化简的方法。

并项法：

$$A + \bar{A} = 1$$

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

吸收法：

$$A + AB = A$$

$$L = \underline{\bar{A}B} + \underline{\bar{A}BCD}(E + F) = \bar{A}B$$

消去法：

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

$$L = AB + \underline{\bar{A}C} + \underline{\bar{B}C} = AB + (\bar{A} + \bar{B})C \quad \bar{A} + \bar{B} = \overline{AB}$$

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

配项法：

$$1 = A + \bar{A}$$

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

$$= \underline{AB} + \underline{\bar{A}\bar{C}} + \underline{AB\bar{C}} + \underline{\bar{A}B\bar{C}}$$

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

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

例 已知逻辑函数表达式为

$$L = \bar{A}\bar{B}\bar{D} + A\bar{B}\bar{D} + \bar{A}BD + A\bar{B}\bar{C}D + A\bar{B}CD$$

要求：(1) 最简的**与-或逻辑函数**表达式，并画出逻辑图；
(2) 仅用**与非门**画出最简表达式的逻辑图。

解：

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

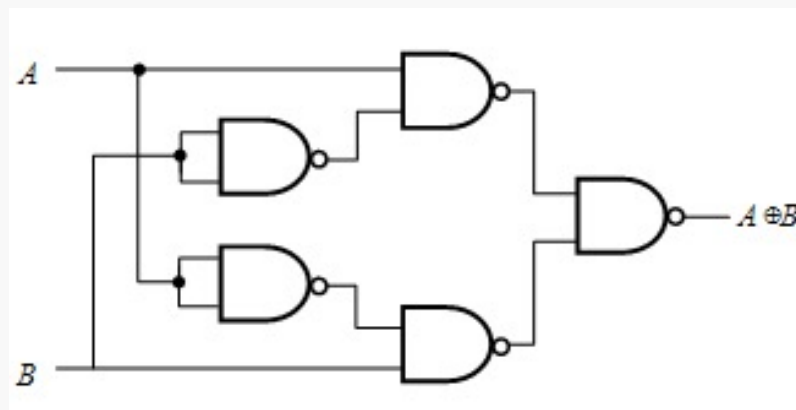
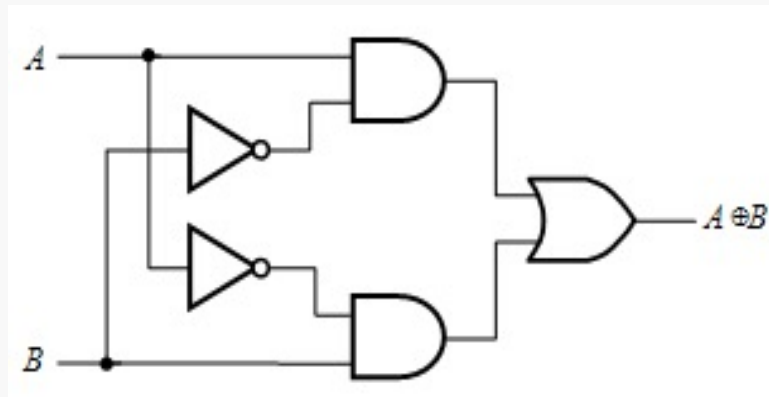
$$= \bar{A}B + A\bar{B}\bar{D} + A\bar{B}D$$

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

$$= \bar{A}B + A\bar{B} \quad (\text{与-或表达式})$$

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

$$= \overline{\bar{A}B \cdot \overline{A\bar{B}}} \quad (\text{与非-与非表达式})$$



2.2.3 逻辑代数法化简存在的问题

1. **逻辑代数与普通代数的公式易混淆**，化简过程要求对所有公式熟练掌握；
2. 代数法化简无一套完善的方法可循，它依赖于人的**经验和灵活性**；
3. 用这种化简方法技巧强，较难掌握。特别是对代数化简后得到的逻辑表达式**是否是最简式**判断有一定困难。

$$\begin{aligned} F &= \textcolor{red}{B}\overline{C} + \overline{B}C + B\overline{D} + \textcolor{red}{B}D \\ &= \underline{\underline{B\overline{C}}} + \underline{\underline{\overline{B}C}} + \underline{\underline{B\overline{D}}} + \underline{\underline{\overline{B}D}} + \underline{\underline{\textcolor{red}{C}D}} = \overline{B}C + B\overline{D} + \overline{C}D \end{aligned}$$