

## 数电精编复习笔记目录

|                     |    |
|---------------------|----|
| 第一章 数制与编码.....      | 1  |
| 第二章 逻辑代数基础.....     | 5  |
| 第三章 集成逻辑门.....      | 9  |
| 第四章 组合逻辑电路.....     | 11 |
| 第五章 触发器.....        | 22 |
| 第六章 时序逻辑电路.....     | 30 |
| 第七章 脉冲波形的产生与整形..... | 34 |
| 第八章 可编程逻辑器件.....    | 38 |

# 第一章 数制与编码

## 1.2 数制

1.2.1 进位计数制 每一种进位计数制允许使用的数符总数的基数/底数

十进制数 (Decimal): 进位基数  $R = 10$ , “逢十进一”, 每位数符只能取  $0 \sim 9$ , 每位的权是 10 的幂

二进制数 (Binary): 进位基数  $R = 2$ , “逢二进一”, 每位数符只能取 0, 1, 每位的权是 2 的幂

八进制数 (Octal): 进位基数  $R = 8$ , “逢八进一”, 每位数符只能取  $0 \sim 7$ , 每位的权是 8 的幂

十六进制数 (Hexa decimal): 进位基数  $R = 16$ , “逢十六进一”, 每位数符只能取  $0 \sim 15$ , 每位的权是 16 的幂

1.2.2 进位计数制之间的转换

(1)  $B \rightarrow D$  按权展开 (任意进制  $\rightarrow$  十进制 都可按权展开)

(2)  $D \rightarrow B$  ① 除 2 取余法 (用于整数转换) 余数从后 往前读

② 乘 2 取整法 (用于小数转换) 整数从前往后读

小结:  $D \rightarrow B$  应分为整数和小数部分, 分别进行转换, 最后合并

(3)  $B \leftrightarrow O$  (八),  $B \leftrightarrow H$  (十六)

$B \rightarrow O$  每三位划为 1 组 每组中的二进制数写为八进制数

$B \rightarrow H$  每四位划为 1 组 每组中的二进制数写为十进制数

$O \rightarrow B$  把每位八进制数写为二进制数

$H \rightarrow B$  把每位十六进制数写为二进制数

注: 八进制与十六进制互换可以做为二进制的中介

## 1.3 编码

1.3.1 带符号数的编码

原码: 符号位+数值位 (若未达到规定字长, 则在数值位前补 0)

反码:  $\begin{cases} \text{正数的反码} = \text{原码} \\ \text{负数的反码} = \text{相应正数反码的符号位取1, 数值位各位取反} \end{cases}$

补码:  $\begin{cases} \text{正数和0的补码} = \text{原码} \\ \text{负数的补码 其原码 符号位不变 对数值位各位取反, 再在最低位加上1} \end{cases}$

注: 几位字长的二进制原码, 反码, 补码所表示的十进制范围为:

$\begin{cases} \text{原码: } -(2^{n-1}-1) \sim +(2^{n-1}-1) \text{ 若 } n=4, \text{ 则为 } -7 \sim +7 \\ \text{反码: } -(2^{n-1}-1) \sim +(2^{n-1}-1) \text{ 若 } n=4, \text{ 则为 } -7 \sim +7 \\ \text{补码: } -2^{n-1} \sim +(2^{n-1}-1), \text{ 不含0, 若 } n=4, \text{ 则为 } -8 \sim +7 \end{cases}$

补码的计算:

$$7-5: [7]_{\text{补}} = 0111, [-5]_{\text{补}} = 1011 \quad \therefore [7-5]_{\text{补}} = [7]_{\text{补}} + [-5]_{\text{补}} = 10010$$

$$\text{用于符号位}=0, [7-5]_{\text{补}} = 0010, \therefore 7-5 = +2$$

$$[3]_{\text{补}} = 0011, [-4]_{\text{补}} = 1100 \quad \therefore [3-4]_{\text{补}} = [3]_{\text{补}} + [-4]_{\text{补}} = 1111$$

$$\text{由于符号位}=1, \therefore [3-4]_{\text{补}} = 1001, \therefore 3-4 = -1$$

### 1.3.2 二一十进制编码 (BCD 码)

① 8421BCD 码: 0000,0001,0010,0011,0100,0101,0110,0111,1000,1001

(0~9) 其余不需要 (用 4 位字长的二进制码表示一位十进制数)

② 5421BCD 码: 从高 → 低 各位权值分别为 5.4.2.1

0000,0001,0010,0011,0100,1000,1001,1010,1011,1100

③ 余 3 码 由 8421BCD 码每个码组 (每 4 位一组) 加上 0011 得到

### 1.3.3 可靠性编码

1.Gray 码: 又称循环码, 单位距离码, 具有相邻性 (任何两组相邻的代码中, 仅有一位码不同), 循环性 (首尾两个码也有循环性)

Gray 码具有 “反射” 特性, 因此方便记忆

二进制码  $\leftrightarrow$  Gray 码

|             |             |                     |
|-------------|-------------|---------------------|
| 0000        | 0000        |                     |
| <u>0001</u> | <u>0001</u> | 1 位对称, 2 位变 1       |
| 0010        | 0011        |                     |
| <u>0011</u> | <u>0010</u> | 1, 2 位对称, 3 位变 1    |
| 0100        | 0110        |                     |
| 0101        | 0111        |                     |
| 0110        | 0101        |                     |
| <u>0111</u> | <u>0100</u> | 1, 2, 3 位对称, 4 位变 1 |
| 1000        | 1100        |                     |
| 1001        | 1101        |                     |
| 1010        | 1111        |                     |
| 1011        | 1110        |                     |
| 1100        | 1010        |                     |
| 1101        | 1011        |                     |
| 1110        | 1001        |                     |
| 1111        | 1000        |                     |

2. 奇偶校验码：由信息位和校验位组成；优点是编码简单，设备量少

信息位是位数不限的任何一种二进制代码

校验位：  
 奇校验由校验位要使得信息位和校验位中“1”的个数为奇数  
 偶校验由校验位要使得信息位和校验位中“1”的个数为偶数

| 十进制数 | 8421 奇校验码 |     | 8421 偶校验码 |     |
|------|-----------|-----|-----------|-----|
|      | 信息位       | 校验位 | 信息位       | 校验位 |
| 1    | 0001      | 0   | 0001      | 1   |
| 2    | 0010      | 0   | 0010      | 1   |
| 3    | 0011      | 1   | 0011      | 0   |
| 4    | 0100      | 0   | 0100      | 1   |
| 5    | 0101      | 1   | 0101      | 0   |
| 6    | 0110      | 1   | 0110      | 0   |

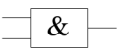
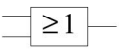
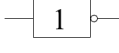
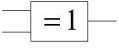
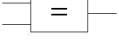
|   |      |   |      |   |
|---|------|---|------|---|
| 7 | 0111 | 0 | 0111 | 1 |
| 8 | 1000 | 0 | 1000 | 1 |

注：奇偶校验码是能检查出，奇数个代码出错，

但无法确谁是哪一位出错，因此没有纠错能力

## 第二章 逻辑代数基础

### 1. 逻辑符号:

|      |                                                                                   |                                                                                   |
|------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|
| 与门:  |  |  |
| 或门:  |  |  |
| 非门:  |  |  |
| 异或门: |  |  |
| 同或门: |  |  |
| 与非门: |  |  |
| 或非门: |  |  |

### 2. 逻辑代数的基本定律:

分配律:  $A+B \cdot C = (A+B) \cdot (A+C)$ ,  $A(B+C) = AB+AC$

反演律:  $\overline{A+B} = \overline{A} \cdot \overline{B}$ ,  $\overline{A \cdot C} = \overline{A} + \overline{C}$

### 3. 反演规则: 用于求一个函数的反函数

方法: 把  $\cdot$  换成  $+$ ,  $+$  换成  $\cdot$ , 0 换 1, 1 换 0, 原变量换成反变量, 反变量换成原变量

Eg:  $F = A + \overline{B} + \overline{C + \overline{D} + E}$ , 则  $\overline{F} = \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{\overline{D} + E}$

注: ① 不能破坏原式运算顺序, 优先改变括号里的, 必须“先与后或”

② 若非号不再单变量上, 则保持不变

### 4. 对偶原则: 用于求一个函数的对偶函数

方法: 把  $\cdot$  换成  $+$ ,  $+$  换成  $\cdot$ , 0 换 1, 1 换 0, 而变量不改变

Eg:  $F = \overline{\overline{A} \cdot \overline{B} \cdot \overline{C}}$ , 则  $F_d = \overline{\overline{A} + \overline{B} + \overline{C}}$

注：① 求对偶式时，运算顺序不能变

② 非号全不变（ $\bar{A}$ 不用换成A）

### 5.若干常用公式

$$\begin{array}{l} \left[ \begin{array}{l} 0 \cdot A = 0 \\ 1 \cdot A = A \\ A \cdot A = A \\ \bar{A} \cdot A = 0 \\ B \cdot A = AB \end{array} \right. \quad \left[ \begin{array}{l} 0 + A = A \\ 1 + A = 1 \\ A + A = A \\ \bar{A} \cdot A = 1 \\ B + A = A + B \end{array} \right. \quad \left[ \begin{array}{l} A \cdot (B \cdot C) = (A \cdot B) \cdot C \\ A \cdot (B + C) = A \cdot B + A \cdot C \\ \overline{A \cdot B} = \bar{A} + \bar{B} \\ \overline{A + B} = \bar{A} \cdot \bar{B} \end{array} \right. \\ \left[ \begin{array}{l} A + (B + C) = (A + B) + C \\ A + B \cdot C = (A + B) \cdot (A + C) \\ \overline{(\bar{A})} = A, \bar{1} = 0, \bar{0} = 1 \end{array} \right. \end{array}$$

### 6.五种常用形式的转换

$\nabla$  与或式二次取反，放进去内层非号，得到与非与非式

$$\text{Eg: } F = AB + \bar{A}C = \overline{\overline{AB + \bar{A}C}} = \overline{\overline{AB} \cdot \overline{\bar{A}C}} = \overline{\overline{AB} \cdot A \cdot \bar{C}}$$

$\nabla$  或与式二次取反，放进去内层非号，得到与非与非式

$\nabla$  或与式二次取反，先放进内层非号，再将所得的或非或非式的内层两个非号放进去，得到与或非式

$$\nabla \left[ \begin{array}{l} \text{与非与非式将最外层长非号放进去，得到} \left[ \begin{array}{l} \text{与或式} \\ \text{或与式} \end{array} \right. \\ \text{或非或非式} \end{array} \right.$$

### 7.关于异或和同或

$$\text{① } F = A \oplus B = \bar{A}\bar{B} + \bar{A}B$$

$$\text{② } F = A \odot B = AB + \bar{A}\bar{B}$$

$$\text{③ } \overline{A \oplus B} = A \odot B, \overline{A \odot B} = A \oplus B$$

$$\textcircled{4} \text{ 公式 } \begin{cases} A \oplus \bar{B} = \bar{A} \oplus B = \overline{A \oplus B} \\ A \odot \bar{B} = \bar{A} \odot B = \overline{A \odot B} \end{cases}$$

## 8. 逻辑函数的两种标准形式

### 8.1 最小项和标准或与式

最小项的性质：①  $\sum_{i=0}^{2^n-1} m_i = 1$ ; ②  $m_i \cdot m_j = 0$ ; ③  $n$  变量的每个最小项有几个相邻项为

$$\bar{A}BC, A\bar{B}C, ABC$$

最小项表达式（标准与或式/标准与项/标准积之和式）

即：在一个与或表达式中，所有与项均为最小项

$$\text{形如 } F(A, B, C) = ABC + A\bar{B}\bar{C} + \bar{A}BC = m_1 + m_4 + m_3 = \sum m_{(347)}$$

注：表现在真值表里即为：把所有使  $F=1$  的与项相或

### 8.2 最大项和标准或与式

最大项的性质：①  $\sum_{i=0}^{2^n-1} M_i = 0$ ; ②  $M_i + M_j = 1 (i+j)$  ③  $n$  变量的每个最大项有几个相

邻项

8.3 最大项与最小项的关系：  $\bar{m}_i = M_i, \bar{M}_i = m_i$

## 9. 逻辑函数的化简方法

9.1 最简与或式：  $\begin{cases} \text{与项最少，且每个与项中变量数最少的与或表达式} \\ \text{或项最少，且每个或项中变量数最少的或与表达式} \end{cases}$

两种化简方法：① 代数化简法 ② K 图化简法

○ 要画成最简与或式，则在 K 图中“圈 1”

$$\text{Eg: } F = \bar{B}CD + AB\bar{D} + \bar{B}C\bar{D} + ABC\bar{D} + ABCD$$

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

|    |   |   |   |   |
|----|---|---|---|---|
| 01 |   |   | 1 |   |
| 11 | 1 |   | 1 | 1 |
| 10 | 1 | 1 |   | 1 |

在能使每个与项为“1”的框里填 1

$$\text{则 } F = \overline{B}C + \overline{A}B\overline{D} + AB\overline{C} + ACD$$

○ 要画成最简或与式，则在 K 图中“圈 o”

填图：在能使每个或项为“0”的框里填 0

10. 具有无关项的逻辑函数的化简

Eg: 将以下函数化简为最简与或非式

$$\begin{cases} F = \sum m(2, 4, 6, 8) \\ \overline{A}\overline{B}\overline{C} + AB\overline{C}D = Q \rightarrow \text{这里指的是约束条件} \end{cases}$$

一旦  $\overline{A}\overline{B}\overline{C} = 1$  或  $\overline{A}\overline{B}\overline{C}.AB\overline{C}D$  都=1，则不满足条件，所以要在 K 图中将所有使得

$\overline{A}\overline{B}\overline{C} = 1$  和  $AB\overline{C}D = 1$  的框画“X”，并当成无关项使用

| AB \ CD | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00      | X  | 1  | 0  | 1  |
| 01      | X  | 0  | X  | 0  |
| 11      | 0  | 0  | 0  | 0  |
| 10      | 1  | 1  | 0  | 0  |

$$F \text{ 的最简与或式为: } F = \overline{D} \cdot (\overline{A} + \overline{C})(\overline{A} + \overline{B})$$

$$F \text{ 的最简或非或非式为: } F = \overline{\overline{\overline{D}} \cdot (\overline{A} + \overline{C})(\overline{A} + \overline{B})} = \overline{D + \overline{\overline{A} + \overline{C}} + \overline{\overline{A} + \overline{B}}}$$

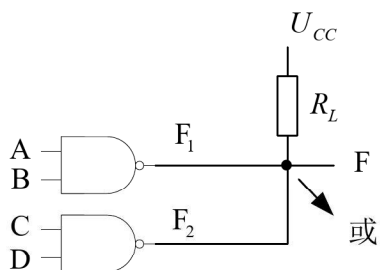
$$F \text{ 的最简与或非式为: } F = \overline{D + A \cdot C + A \cdot B}$$

注：要求最简与非式，或非式，与或非式，全都归结为先求最简与或式/最简或与式

### 第三章 集成逻辑门

#### 1. 集电极开路门（OC 门）及“线与”功能

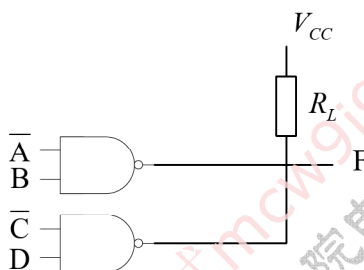
Eg:



两个 OC 门可实现线与

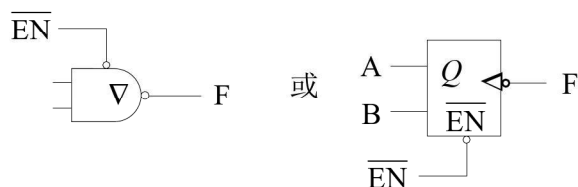
$$F = \overline{AB} + \overline{CD} = \overline{AB \cdot CD}$$

Eg:



$$F = \overline{AB} + \overline{CD} = \overline{AB \cdot CD}$$

#### 2. 三态门（TS 门）



符号:

Eg:



## 第四章 组合逻辑电路

### 4.1 组合逻辑电路的分析

步骤：① 写出输出端的逻辑函数表达式

② 列真值表

③ 用文字概括出电路的功能（根据输入与输出之间的逻辑关系）

一位全加器真值表:

| $A_i$ | $B_i$ | $C_i$ | $C_{i+1}$ (向高位的进位) | $S_i$ (本位和) |
|-------|-------|-------|--------------------|-------------|
| 0     | 0     | 0     | 0                  | 0           |
| 0     | 0     | 1     | 0                  | 1           |
| 0     | 1     | 0     | 0                  | 1           |
| 0     | 1     | 1     | 1                  | 0           |
| 1     | 0     | 0     | 0                  | 1           |
| 1     | 0     | 1     | 1                  | 0           |
| 1     | 1     | 0     | 1                  | 0           |
| 1     | 1     | 1     | 1                  | 1           |

一位半加器真值表:

| $A_i$ | $B_i$ | $C_i$ | $S_i$ |
|-------|-------|-------|-------|
| 0     | 0     | 0     | 0     |
| 0     | 1     | 0     | 1     |
| 1     | 0     | 0     | 1     |
| 1     | 1     | 1     | 0     |

### 4.2 组合逻辑电路的设计

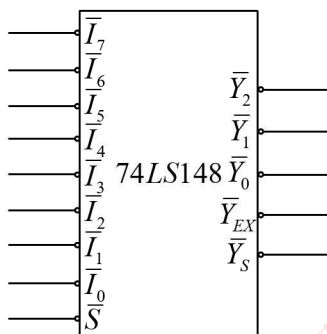
步骤：① 根据题目描述的命题设输入输出变量，并列出真值表

② 根据真值表填 K 图，并根据要求圈“1”或圈“0”

③ 写出输出的逻辑表达式（可以是最简式）

## 4.3 常用中规模组合逻辑器件

### 4.3.1 编码器



$\bar{S}$ ：输出使能端，低电平有效

$\bar{S} = 0$  电路工作

$\bar{S} = 1$  电路不工作，输出全“1”。

$\bar{Y}_s$ ：选通输出端，低电平有效。

$\bar{Y}_s = \bar{I}_0 \cdot \bar{I}_1 \cdot \bar{I}_2 \cdot \bar{I}_3 \cdot \bar{I}_4 \cdot \bar{I}_5 \cdot \bar{I}_6 \cdot \bar{I}_7 \cdot \bar{S}$  “电路工作，无编码信号输入”

$\bar{Y}_{EX}$ ：扩展输出端，低电平有效

$\bar{Y}_{EX} = \bar{Y}_s \cdot \bar{S} = \overline{(I_0 + I_1 + I_2 + I_3 + I_4 + I_5 + I_6 + I_7)} \cdot \bar{S}$  “电路工作，有编码信号输入”

| 74LS148 功能表    |                  |                  |                  |                  |                  |                  |                  |                  |                  |                  |                  |                     |                  |
|----------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|---------------------|------------------|
| 输入             |                  |                  |                  |                  |                  |                  |                  |                  | 输出               |                  |                  |                     |                  |
| $\overline{S}$ | $\overline{I_7}$ | $\overline{I_6}$ | $\overline{I_5}$ | $\overline{I_4}$ | $\overline{I_3}$ | $\overline{I_2}$ | $\overline{I_1}$ | $\overline{I_0}$ | $\overline{Y_2}$ | $\overline{Y_1}$ | $\overline{Y_0}$ | $\overline{Y_{EX}}$ | $\overline{Y_s}$ |
| 1              | X                | X                | X                | X                | X                | X                | X                | X                | 1                | 1                | 1                | 1                   | 1                |
| 0              | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                   | 0                |
| 0              | 0                | X                | X                | X                | X                | X                | X                | X                | 0                | 0                | 0                | 0                   | 1                |
| 0              | 1                | 0                | X                | X                | X                | X                | X                | X                | 0                | 0                | 1                | 0                   | 1                |
| 0              | 1                | 1                | 0                | X                | X                | X                | X                | X                | 0                | 1                | 0                | 0                   | 1                |
| 0              | 1                | 1                | 1                | 0                | X                | X                | X                | X                | 0                | 1                | 1                | 0                   | 1                |

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 0 | X | X | X | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | X | X | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | X | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |

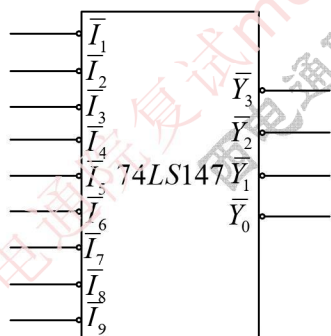
| $\overline{Y}_s$ | $\overline{Y}_{EX}$ | 电路工作状态                                                     |
|------------------|---------------------|------------------------------------------------------------|
| 1                | 1                   | 电路不工作, $\overline{Y}_2 \sim \overline{Y}_0$ 输出全为 1         |
| 0                | 1                   | 电路工作, 无编码信号输入, $\overline{Y}_2 \sim \overline{Y}_0$ 输出全为 1 |
| 1                | 0                   | 电路工作, 有编码信号输入。                                             |

#### 74LS147 (二-十进制优先编码器)

把  $\overline{I}_0 \sim \overline{I}_9$  的十个状态分别编码成十个 BCD 码。其中  $\overline{I}_9$  的优先权最高,  $\overline{I}_0$  的优先权最低。

编码功能: 输入:  $\overline{I}_0 \sim \overline{I}_9$  低电平有效

输出:  $\overline{Y}_3 \sim \overline{Y}_0$  以反码的形式输出

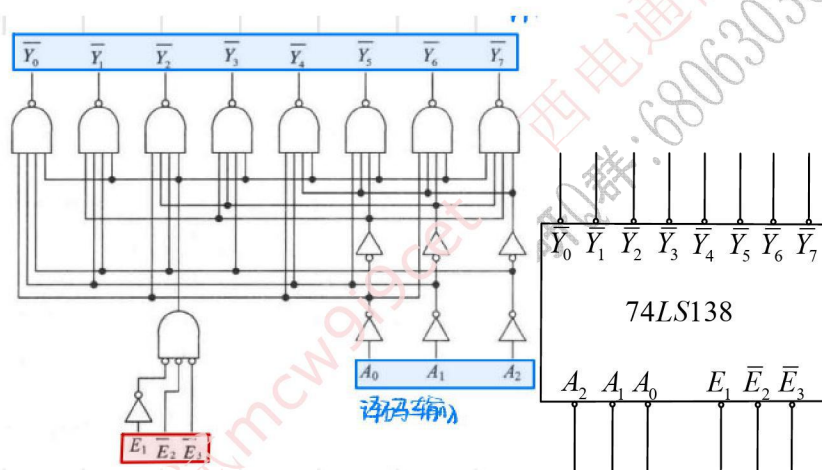


| 74LS147 功能表      |                  |                  |                  |                  |                  |                  |                  |                  |                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| $\overline{I}_1$ | $\overline{I}_2$ | $\overline{I}_3$ | $\overline{I}_4$ | $\overline{I}_5$ | $\overline{I}_6$ | $\overline{I}_7$ | $\overline{I}_8$ | $\overline{I}_9$ | $\overline{Y}_3$ | $\overline{Y}_2$ | $\overline{Y}_1$ | $\overline{Y}_0$ |
| 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                |
| X                | X                | X                | X                | X                | X                | X                | X                | 0                | 0                | 1                | 1                | 0                |
| X                | X                | X                | X                | X                | X                | X                | 0                | 1                | 0                | 1                | 1                | 1                |

|   |   |   |   |   |   |   |   |   |  |   |   |   |   |
|---|---|---|---|---|---|---|---|---|--|---|---|---|---|
| X | X | X | X | X | X | 0 | 1 | 1 |  | 1 | 0 | 0 | 0 |
| X | X | X | X | X | 0 | 1 | 1 | 1 |  | 1 | 0 | 0 | 1 |
| X | X | X | X | 0 | 1 | 1 | 1 | 1 |  | 1 | 0 | 1 | 0 |
| X | X | X | 0 | 1 | 1 | 1 | 1 | 1 |  | 1 | 0 | 1 | 1 |
| X | X | 0 | 1 | 1 | 1 | 1 | 1 | 1 |  | 1 | 1 | 0 | 0 |
| X | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  | 1 | 1 | 1 | 0 |

### 4.3.2 译码器

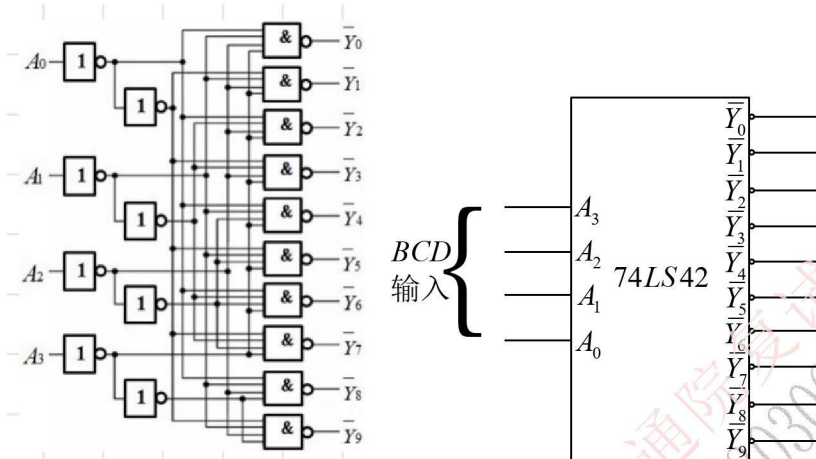
74LS138 (3 线-8 线译码器)



| $E_1$ | $\overline{E_2} + \overline{E_3}$ | $A_2$ | $A_1$ | $A_0$ | $\overline{Y_0}$ | $\overline{Y_1}$ | $\overline{Y_2}$ | $\overline{Y_3}$ | $\overline{Y_4}$ | $\overline{Y_5}$ | $\overline{Y_6}$ | $\overline{Y_7}$ |
|-------|-----------------------------------|-------|-------|-------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| 0     | X                                 | X     | X     | X     | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                |
| X     | 1                                 | X     | X     | X     | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                |
| 1     | 0                                 | 0     | 0     | 0     | 0                | 1                | 1                | 1                | 1                | 1                | 1                | 1                |
| 1     | 0                                 | 0     | 0     | 1     | 1                | 0                | 1                | 1                | 1                | 1                | 1                | 1                |
| 1     | 0                                 | 0     | 1     | 0     | 1                | 1                | 0                | 1                | 1                | 1                | 1                | 1                |
| 1     | 0                                 | 0     | 1     | 1     | 1                | 1                | 1                | 0                | 1                | 1                | 1                | 1                |
| 1     | 0                                 | 1     | 0     | 0     | 1                | 1                | 1                | 1                | 0                | 1                | 1                | 1                |
| 1     | 0                                 | 1     | 0     | 1     | 1                | 1                | 1                | 1                | 1                | 0                | 1                | 1                |

|   |   |       |                 |
|---|---|-------|-----------------|
| 1 | 0 | 1 1 0 | 1 1 1 1 1 1 0 1 |
| 1 | 0 | 1 1 1 | 1 1 1 1 1 1 1 0 |

二十一进制译码器（BCD/4 线-10 线）

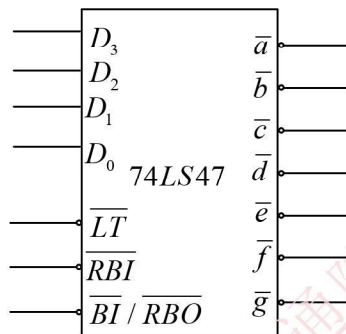


| 序号 | 输入             |                |                |                | 输出               |                  |                  |                  |                  |                  |                  |                  |                  |                  |
|----|----------------|----------------|----------------|----------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
|    | A <sub>3</sub> | A <sub>2</sub> | A <sub>1</sub> | A <sub>0</sub> | $\overline{Y}_0$ | $\overline{Y}_1$ | $\overline{Y}_2$ | $\overline{Y}_3$ | $\overline{Y}_4$ | $\overline{Y}_5$ | $\overline{Y}_6$ | $\overline{Y}_7$ | $\overline{Y}_8$ | $\overline{Y}_9$ |
| 0  | 0              | 0              | 0              | 0              | 0                | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                |
| 1  | 0              | 0              | 0              | 1              | 1                | 0                | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                |
| 2  | 0              | 0              | 1              | 0              | 1                | 1                | 0                | 1                | 1                | 1                | 1                | 1                | 1                | 1                |
| 3  | 0              | 0              | 1              | 1              | 1                | 1                | 1                | 0                | 1                | 1                | 1                | 1                | 1                | 1                |
| 4  | 0              | 1              | 0              | 0              | 1                | 1                | 1                | 1                | 0                | 1                | 1                | 1                | 1                | 1                |
| 5  | 0              | 1              | 0              | 1              | 1                | 1                | 1                | 1                | 1                | 0                | 1                | 1                | 1                | 1                |
| 6  | 0              | 1              | 1              | 0              | 1                | 1                | 1                | 1                | 1                | 1                | 0                | 1                | 1                | 1                |
| 7  | 0              | 1              | 1              | 1              | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 0                | 1                | 1                |
| 8  | 1              | 0              | 0              | 0              | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 0                | 1                |
| 9  | 1              | 0              | 0              | 1              | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 0                |
| 伪码 | 1              | 0              | 1              | 0              | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                |
|    | 1              | 0              | 1              | 1              | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                |
|    | 1              | 1              | 0              | 0              | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                |
|    | 1              | 1              | 0              | 1              | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                |
|    | 1              | 1              | 1              | 0              | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                | 1                |

|  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

### 74LS47 (BCD-7 段数码管译码器驱动器)

功能用于将 BCD 码转化为数码块中的数字，通过他来解码，可直接把数字转换为数码管的数字



$\overline{LT}$ : 试灯输入，为了检查数码管各段是否可正常发光

当  $\overline{LT} = 0$ ，任何输入、输出均为全亮

$\overline{BI}$ : 灭灯输入，是为控制多位数码显示的灭灯所设置的

$\overline{BI} = 0$ ，不论  $\overline{LT}, D_3 - D_0$  任何状态，输出均为高电平且熄灭

$\overline{RBI}$ : 灭零输入，为了使不希望显示的 0 熄灭而设定

$D_3 = D_2 = D_1 = D_0 = 0$  时输出本应为 0，但在  $\overline{RBI} = 0$  作用下，输出均为高电平即熄灭

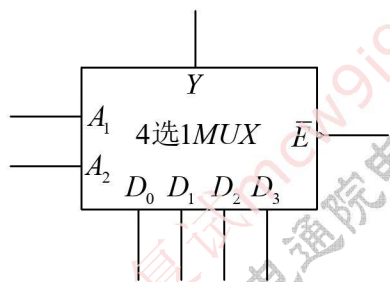
$\overline{RBO}$ : 灭零输出，它和灭灯输入  $\overline{BI}$  共用一端，配合使用，可实现多位数码显示的灭零控制

| 输入              |                  |         | $\overline{BI} / \overline{RBO}$ | 输出            | 功能及显示字符 |
|-----------------|------------------|---------|----------------------------------|---------------|---------|
| $\overline{LT}$ | $\overline{RBI}$ |         |                                  |               |         |
| 0               | X                | X X X X | 1                                | 0 0 0 0 0 0 0 | 试灯输入    |
| 1               | 0                | 0 0 0 0 | 0                                | 1 1 1 1 1 1 1 | 纹波灭零输入  |
| X               | X                | X X X X | 0                                | 1 1 1 1 1 1 1 | 熄灭输入    |

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | X | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | X | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | X | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | X | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | X | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | X | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | X | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | X | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | X | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |

#### 4.3 数据选择器

74LS153 (双 4 选 1 数据选择器)

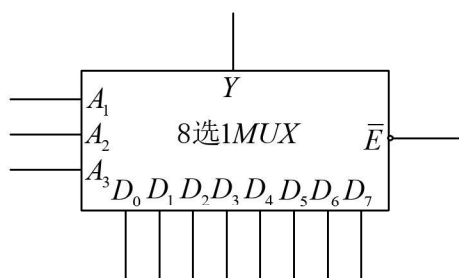


$\bar{E}$ : 选通控制端

$\bar{E}=0$  时, 数据选择器工作;  $\bar{E}=1$  时,  $Y=0$ , 输出无效

逻辑函数式:  $Y = [D_0(\bar{A}_1\bar{A}_0) + D_1(\bar{A}_1A_0) + D_2(A_1\bar{A}_0) + D_3(A_1A_0)] \cdot \bar{E}$

74LS151 (8 选 1 数据选择器)



$\bar{E}$ :选通控制端

$\bar{E}=0$ 时, 数据选择器工作;  $\bar{E}=1$ 时,  $Y=0$ , 输出无效

逻辑函数式:  $Y = D_0(\bar{A}_2\bar{A}_1\bar{A}_0) + D_1(\bar{A}_2\bar{A}_1A_0) + D_2(\bar{A}_2A_1\bar{A}_0) + D_3(\bar{A}_2A_1A_0) + D_4(A_2\bar{A}_1\bar{A}_0) + D_5(A_2\bar{A}_1A_0) + D_6(A_2A_1\bar{A}_0) + D_7(A_2A_1A_0)$

#### 4.4 加法器

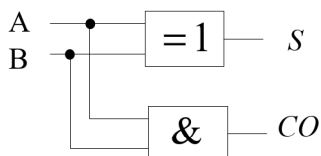
半加器: 不考虑低位进位, 将两个 1 位二进制数 A 和 B 相加

输入: 两个加数 A、B

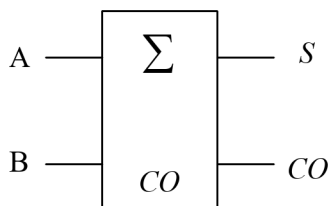
输出: 相加的和 S, 向高位的进位 CO

| 输入 |   | 输出 |    |
|----|---|----|----|
| A  | B | S  | CO |
| 0  | 0 | 0  | 0  |
| 0  | 1 | 1  | 0  |
| 1  | 0 | 1  | 0  |
| 1  | 1 | 0  | 1  |

逻辑函数式:  $S = A\bar{B} + \bar{A}B = A \oplus B$      $CO = AB$



逻辑图:



逻辑符号:

全加器：考虑低位进位，将两个加数和低位进位一起相加

输入：两个加数 A、B；低位的进位 CI

输出：相加的和 S，向高位的进位 CO

全加器真值表

| 输入 |   |    | 输出 |    |
|----|---|----|----|----|
| A  | B | CI | S  | CO |
| 0  | 0 | 0  | 0  | 0  |
| 0  | 0 | 1  | 1  | 0  |
| 0  | 1 | 0  | 1  | 0  |
| 0  | 1 | 1  | 0  | 1  |
| 1  | 0 | 0  | 1  | 0  |
| 1  | 0 | 1  | 0  | 1  |
| 1  | 1 | 0  | 0  | 1  |
| 1  | 1 | 1  | 1  | 1  |

逻辑函数式：

$$\begin{aligned}
 S &= \overline{A}\overline{B}CI + \overline{A}B\overline{C}I + A\overline{B}\overline{C}I + ABCI \\
 &= (\overline{A}B + A\overline{B})\overline{C}I + (\overline{A}\overline{B} + AB)CI \\
 &= (A \oplus B)\overline{C}I + (\overline{A} \oplus \overline{B})CI \\
 &= A \oplus B \oplus CI
 \end{aligned}$$

$$\begin{aligned}
 CO &= \overline{A}BCI + \overline{A}B\overline{C}I + A\overline{B}\overline{C}I + ABCI \\
 &= (\overline{A}B + AB)\overline{C}I + AB(CI + CI) \\
 &= (A \oplus B)CI + AB
 \end{aligned}$$

超进位加法器：本位两个加数相加的同时，也已获得低位的进位  $CO_{i-1}$ ，则一次就可以完成多位加法运算

$$S_i = A_i \oplus B_i \oplus CI_i = A_i \oplus B_i \oplus CO_{i-1}$$

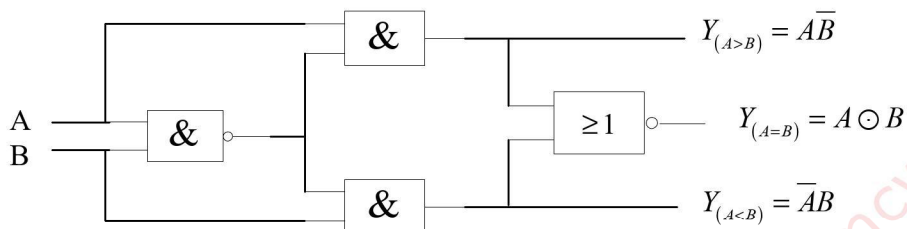
优点：工作速度快，每位的和及最后的进位基本同时产生。

缺点：电路复杂

## 4.5 数值比较器

### 4.5.1 一位数值比较器

$$\begin{aligned} \textcircled{1} A > B (A=1, B=0) \\ \textcircled{2} A < B (A=0, B=1) \\ \textcircled{3} A = B (A=B=0, A=B=1) \end{aligned} \Rightarrow \begin{cases} Y_{(A>B)} = AB \\ Y_{(A<B)} = \overline{AB} \\ Y_{(A=B)} = A \odot B \end{cases}$$



## 4.5.2 多位数值比较器

### (1) 比较方法

比较两个多位数 A 和 B，需从高 → 低逐位比较

Eg: 两个 4 位二进制数  $A[3:0]$  和  $B[3:0]$  进行比较

$$\begin{aligned} & \left[ \begin{aligned} & A_3 > B_3 \rightarrow A > B \quad A_3 < B_3 \rightarrow A < B \\ & A_3 = B_3 \rightarrow \left[ \begin{aligned} & A_2 > B_2 \rightarrow A > B \quad A_2 < B_2 \rightarrow A < B \\ & A_2 = B_2 \rightarrow \left[ \begin{aligned} & A_1 > B_1 \rightarrow A > B \quad A_1 < B_1 \rightarrow A < B \\ & A_1 = B_1 \rightarrow \left[ \begin{aligned} & A_0 > B_0 \rightarrow A > B \\ & A_0 < B_0 \rightarrow A < B \\ & A_0 = B_0 \rightarrow A = B \end{aligned} \end{aligned} \end{aligned} \end{aligned} \right. \end{aligned} \right. \end{aligned}$$

| 比较输入        |             |             |             | 级联输入    |         |         | 输出        |           |           |
|-------------|-------------|-------------|-------------|---------|---------|---------|-----------|-----------|-----------|
| $A_3 \ B_3$ | $A_2 \ B_2$ | $A_1 \ B_1$ | $A_0 \ B_0$ | $A > B$ | $A < B$ | $A = B$ | $F_{A>B}$ | $F_{A<B}$ | $F_{A=B}$ |
| $A_3 > B_3$ | X           | X           | X           | X       | X       | X       | 1         | 0         | 0         |
| $A_3 < B_3$ | X           | X           | X           | X       | X       | X       | 0         | 1         | 0         |
| $A_3 = B_3$ | $A_2 > B_2$ | X           | X           | X       | X       | X       | 1         | 0         | 0         |
| $A_3 = B_3$ | $A_2 < B_2$ | X           | X           | X       | X       | X       | 0         | 1         | 0         |
| $A_3 = B_3$ | $A_2 = B_2$ | $A_1 > B_1$ | X           | X       | X       | X       | 1         | 0         | 0         |
| $A_3 = B_3$ | $A_2 = B_2$ | $A_1 < B_1$ | X           | X       | X       | X       | 0         | 1         | 0         |

|           |           |           |           |   |   |   |   |   |   |
|-----------|-----------|-----------|-----------|---|---|---|---|---|---|
| $A_3=B_3$ | $A_2=B_2$ | $A_1=B_1$ | $A_0>B_0$ | X | X | X | 1 | 0 | 0 |
| $A_3=B_3$ | $A_2=B_2$ | $A_1=B_1$ | $A_0<B_0$ | X | X | X | 0 | 1 | 0 |
| $A_3=B_3$ | $A_2=B_2$ | $A_1=B_1$ | $A_0=B_0$ | 1 | 0 | 0 | 1 | 0 | 0 |
| $A_3=B_3$ | $A_2=B_2$ | $A_1=B_1$ | $A_0=B_0$ | 0 | 1 | 0 | 0 | 1 | 0 |
| $A_3=B_3$ | $A_2=B_2$ | $A_1=B_1$ | $A_0=B_0$ | 0 | 0 | 1 | 0 | 0 | 1 |

## 第五章 触发器

时序逻辑电路：任意时刻的输出状态与当前输入信号和此前状态有关，由组合逻辑电路和存储电路组成（存在反馈）

锁存器和触发器是构成时序逻辑电路的基本逻辑单元

触发器 (Flip-Flop)：这种器件平时对输入进行采样，并只有在时钟信号所确定的时刻改变其输出（边沿触发）

锁存器 (Latch)：不断监测其所有输入，并独立于时钟信号而在任何时刻都可以改变输出（电平触发）

RS 触发器

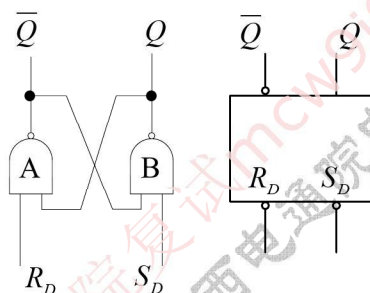
R: Reset, 清零, S: Set, 置位

输入置位和清零信号，触发器给出相应的动作

功能：看 R, R 为 0 就置 0, R 为 1 就置 1

两个互补的输出端  $Q, \bar{Q}$

一般用 Q 表示触发器的状态



| $R_D$ | $S_D$ | $Q^{n+1}$ |
|-------|-------|-----------|
| 0     | 0     | X         |
| 0     | 1     | 0         |
| 1     | 0     | 1         |
| 1     | 1     | $Q$       |

| $R_D S_D$ | 00 | 01 | 11 | 10 |
|-----------|----|----|----|----|
|           |    |    |    |    |

| $Q \backslash$ |   |   |   |   |
|----------------|---|---|---|---|
| 0              | X | 0 | 0 | 1 |
| 1              | X | 0 | 1 | 1 |

特征方程 
$$\begin{cases} Q^{n+1} = \overline{S_D} + R_D Q \\ S_D + R_D = 1 \end{cases}$$

| 激励表                     |   |       |       |
|-------------------------|---|-------|-------|
| $Q \rightarrow Q^{n+1}$ |   | $R_D$ | $S_D$ |
| 0                       | 0 | X     | 1     |
| 0                       | 1 | 1     | 0     |
| 1                       | 0 | 0     | 1     |
| 1                       | 1 | 1     | X     |

与非（全 1 为 0）

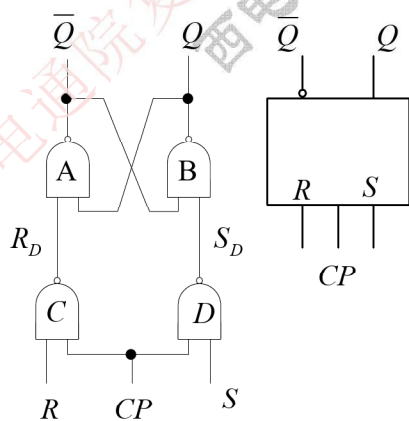


或非（全 0 为 1）



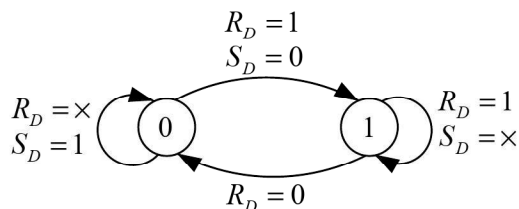
追求稳态

钟控 RS 触发器



| 激励表                     |       |       |
|-------------------------|-------|-------|
| $Q \rightarrow Q^{n+1}$ | $R_D$ | $S_D$ |

|   |   |   |   |
|---|---|---|---|
| 0 | 0 | X | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | X |



状态转移图

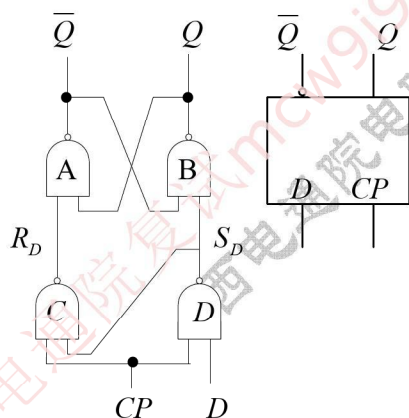
功能：看 S.S 为 0 就置 0，S 为 1 就置 1

CP=0 时，CD 输出为 1，则 R 和 S 输入什么，AB 输入均为 1，状态不改变

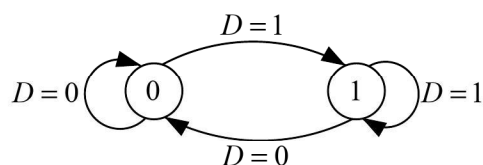
CP=1 时，RS=00 次态 Q 不变，RS=01 次态 Q 为 1，RS=10,次态 Q 为 0；RS=11

不允许

钟控 D 触发器



| 激励表                     |   |   |
|-------------------------|---|---|
| $Q \rightarrow Q^{n+1}$ | D |   |
| 0                       | 0 | 0 |
| 0                       | 1 | 1 |
| 1                       | 0 | 0 |
| 1                       | 1 | 1 |



状态转移图

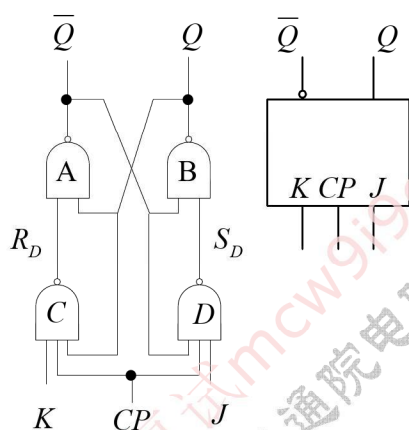
功能：(CP=1, DO 置 0, D1 置 1)

CP=0 时，CD 输出均为 1，则 D 输入什么值，AB 的输入均为 11，则触发器状态不改变

CP=1 时，D 为 0 则置 0，D 为 1 则置 1

注：由于 D 触发器在时钟脉冲作用后的次态和输入 D 一致，所以 D 触发器又可以称为数据锁存器

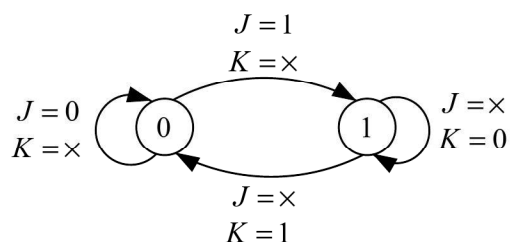
钟控 J-K 触发器



| 状态表 |   |           |
|-----|---|-----------|
| J   | K | $Q^{n+1}$ |
| 0   | 0 | $Q$       |
| 0   | 1 | 0         |
| 1   | 0 | 1         |
| 1   | 1 | $Q$       |

|     |
|-----|
| 激励表 |
|-----|

| $Q \rightarrow Q^{n+1}$ | J | K |
|-------------------------|---|---|
| 0 0                     | 0 | X |
| 0 1                     | 1 | X |
| 1 0                     | X | 1 |
| 1 1                     | X | 0 |



状态转移图

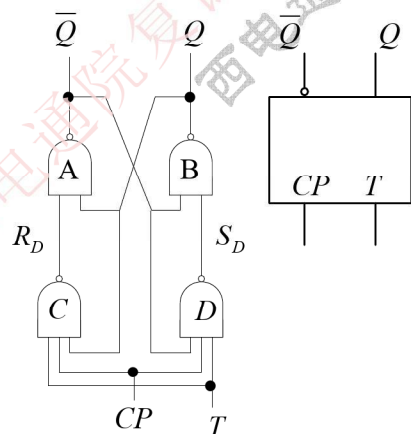
功能（看 J 状态，J0 则 0，J1 则 1）

CP=0 时，CD 输出均为 1，无论 JK 输入什么，AB 输入都为 11，所以触发器状态不变

CP=1 时，JK=00 状态不变，JK=01 置 0，JK=10 置 1，JK=11 则次态和现态翻转

小结：① 钟控 J-K 触发器对 JK 的值没有约束条件，无论取什么值在脉冲作用下都有确定的次态

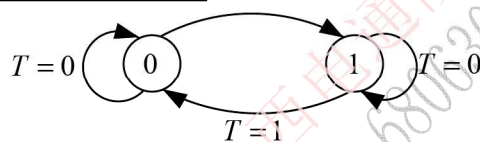
钟控 T 触发器



| 状态表 |           |
|-----|-----------|
| T   | $Q^{n+1}$ |

|   |     |
|---|-----|
| 0 | $Q$ |
| 1 | $Q$ |

| 激励表                     |   |   |
|-------------------------|---|---|
| $Q \rightarrow Q^{n+1}$ |   | T |
| 0                       | 0 | 0 |
| 0                       | 1 | 1 |
| 1                       | 0 | 1 |
| 1                       | 1 | 0 |



状态图

功能:

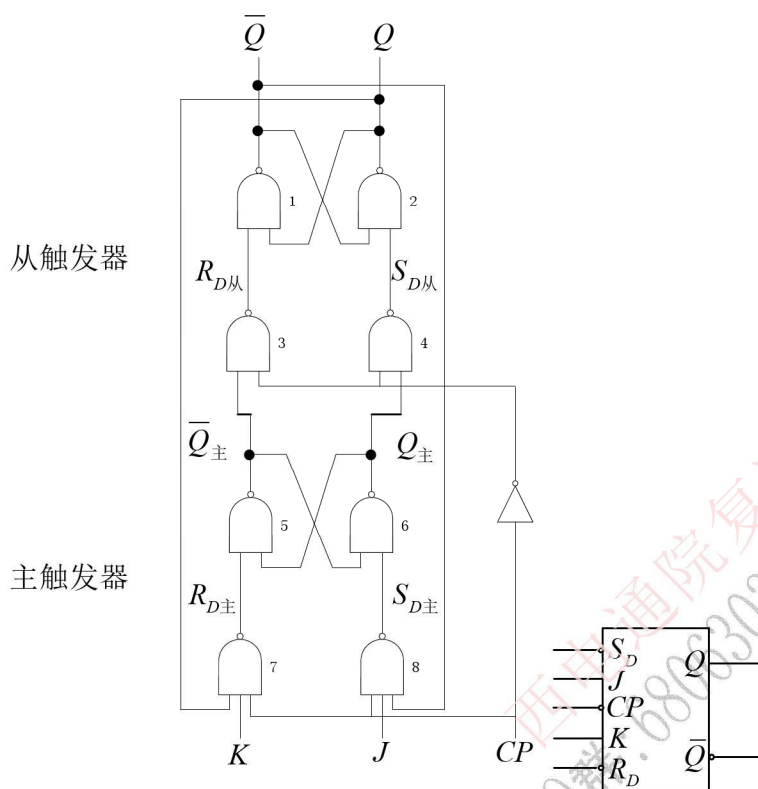
CP=0 时, CD 输出均为 11, T 输入什么值, AB 的输入均为 11, 所以触发器状态不变

CP=1 时, T=0, 状态不变; T=1 状态翻转, 即 Q 为  $0 \rightarrow 1$  或  $1 \rightarrow 0$

小结: ① 钟控 T 触发器只要有时钟脉冲触发, 状态就翻转, 相当于二进制计算器, 又称为计数触发器

### 5.3 集成触发器 (防止空翻、性能稳定)

主从 JK 触发器



功能:

当  $CP=0$  时 ( $\overline{CP}=1$ )

[主:  $JK$  无论输入什么, 5, 6 的输入均为 1,  $Q_{主}$  与  $\bar{Q}_{主}$  保持不变  
从 (打开):  $R_{D从} = Q_{主}$ ,  $S_{D从} = \bar{Q}_{主}$ ,  $Q^{n+1} = Q_{主}$  表示主从触发器状态保持一致

当  $CP=1$  时 ( $\overline{CP}=0$ )

[从 (封锁): 保持原  $Q$  不变  
主:  $R_{D主} = \bar{K}\bar{Q}$ ,  $S_{D主} = \bar{J}\bar{Q}$ ,  $Q_{主}^{n+1} = J\bar{Q} + \bar{K}Q$  主从触发器的翻转发生在下降沿

当  $CP=0$  时,  $Q^{n+1} = Q_{主}^{n+1} = J\bar{Q} + \bar{K}Q$ , 状态由主转移到从触发器

小结: ① 主从 J-K 触发器解决了空翻问题, 主触发器的输出和钟控 J-K 触发器基本相同, 当  $CP$  由  $1 \rightarrow 0$  时, 从变为主的状态, 其余时候维持不变

② 主从 J-K 触发器存在“一次翻转”现象, 即时钟脉冲输入信号发生一次反转后, 其状态保持不变, 不再受输入  $JK$  变化的影响

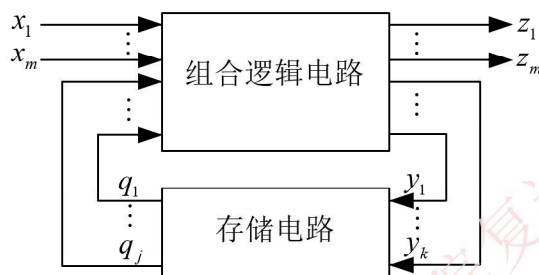
边沿触发器 (维持阻塞 D 触发器)

输出变化发生在 CLK 信号变化的一瞬间，输出由前一级 D 触发器最后时刻的输出信号决定

功能（低有效，异步控制，优先级最高）

## 第六章 时序逻辑电路

- 1.定义：在任意时刻的输出信号不仅取决于当时的输入信号，而且还取决于电路原来的状态
- 2.构成和结构特点



- ① 时序逻辑电路包含组合逻辑电路和存储电路两个部分
- ② 存储电路的输出必须反馈到组合电路的输入端，与输入信号一起，共同决定组合逻辑的输出

- ③ 逻辑方程
 
$$\begin{cases} \text{输出方程: } Z = F(X, Q) \\ \text{驱动方程: } Y = G(X, Q) \\ \text{状态方程: } Q^{n+1} = H(X, Q) \end{cases}$$

### 3.输出信号特点（米利 Mealy & 穆尔 Moore）

米利型：输出信号取决于存储电路的状态和输入变量， $Y = F(X, Q)$

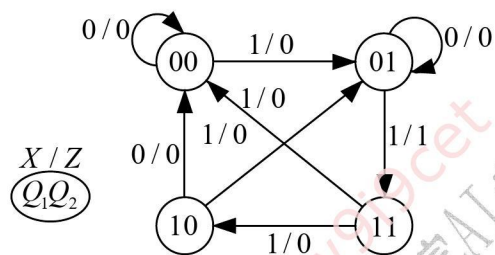
穆尔型：输出信号仅取决于存储电路状态，可表述为  $Y = F(Q)$

Moore 比 Mealy 的输出要晚一个时钟周期

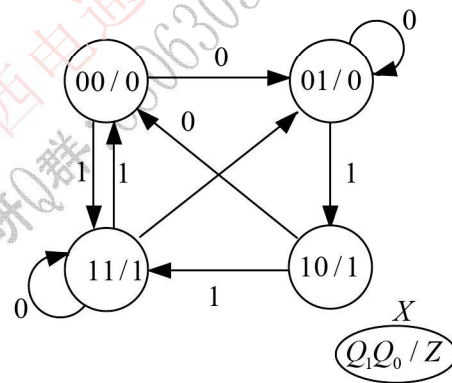
| Mealy 型                                    |                           |      |
|--------------------------------------------|---------------------------|------|
| $\begin{matrix} X \\ Q_1 Q_0 \end{matrix}$ | $Q_1^{n+1} Q_0^{n+1} / Z$ |      |
|                                            | 0                         | 1    |
| 0 0                                        | 01/0                      | 11/1 |
| 0 1                                        | 10/0                      | 00/0 |
| 1 1                                        | 00/0                      | 10/0 |
| 1 0                                        | 11/0                      | 01/0 |

| More 型                                     |                       |    |   |
|--------------------------------------------|-----------------------|----|---|
| $\begin{matrix} X \\ Q_1 Q_0 \end{matrix}$ | $Q_1^{n+1} Q_0^{n+1}$ |    | Z |
|                                            | 0                     | 1  |   |
| 0 0                                        | 01                    | 11 | 0 |
| 0 1                                        | 10                    | 00 | 0 |
| 1 1                                        | 00                    | 10 | 1 |
| 1 0                                        | 11                    | 01 | 0 |

状态图：外部输出在转移条件中给出



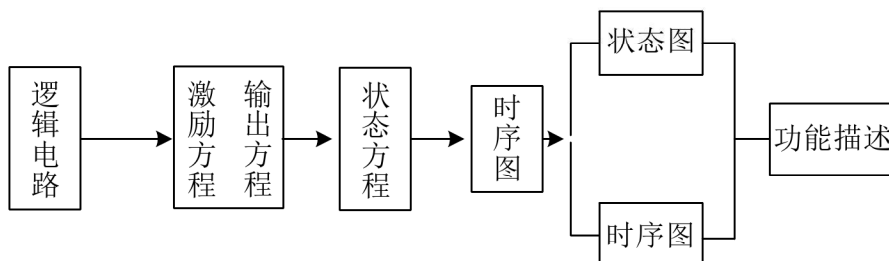
状态图：外部输出在圆圈内指明



外部信号  $\begin{cases} X: \text{外部输入} \\ Z: \text{外部输出} \end{cases}$

内部信号  $\begin{cases} Q: \text{存储状态输出, 组合电路的内部输入} \\ Y: \text{存储状态的激励, 组合电路的内部输出} \end{cases}$

#### 4. 同步时序电路分析



注意：

① 逻辑表达式：右边一定是现态

② 有效状态：一个闭合回路，在电路正常工作时，状态总是按照这个闭合回路里的序列循环变化，通常称处于这个序列循环的状态为有效状态。  
该循环为有效循环或主循环

③ 无效状态：不处于主循的其他状态，即使也是一个循环，也不是有效状态

④ 如果一个时序电路中所有的无效状态都能通向有效序列，则该电路有自启动能力

## 5.同步时序电路设计

根据设计要求 → 写出状态表 → 用卡诺图化简状态，得到状态方程 → 根据状态方程选择触发器 → 画出电路图

计数器的设计

计数器是一个周期性的时序电路，其状态有一个闭合环，闭合环循环一次需要的时钟脉冲个数称为计数器的模值  $M$

由几个触发器构成的计数器，其模值  $M$  一般应满足  $2^{n-1} < M \leq 2^n$

Eg: 设计一个模 5 计数器，状态变化为  $000 \rightarrow 001 \rightarrow 011 \rightarrow 101 \rightarrow 110 \rightarrow 000$  循环往复

分析：有至少 5 个状态，因此需要 3 个触发器

依据上述状态，构建状态图（不在上述存在的状态，标记为 X）

⇒ 凭借卡诺图化简，得到状态方程

⇒ 自启动检查

⇒ 依据状态方程选择触发器

⇒ 构建电路

详见《例 6.4.4》

序列信号发生器的设计（一个能输出周期序列的电路）

设计思路：根据要设计的序列中的 01 个数  $N$ ，先设计一个模值为  $N$  的计数器

计数器中的每个状态对应一个输出的序列值

例：设计能产生 01011 的序列发生器

分析：设计计数器的状态变化为： $000 \rightarrow 001 \rightarrow 011 \rightarrow 101 \rightarrow 110 \rightarrow 000$  循环往复

要发生一个周期序列包含 5 个不同的状态；

先设计一个模 5 的计数器，让每个有效状态对应一个序列中的状态数，  
故需要 3 个触发器

⇒ 根据序列，把  $z$  对应与有效状态的表现，规划好（即  $z$  就是每个状态下对应的输出的序列中的每个 0 或 1 值）

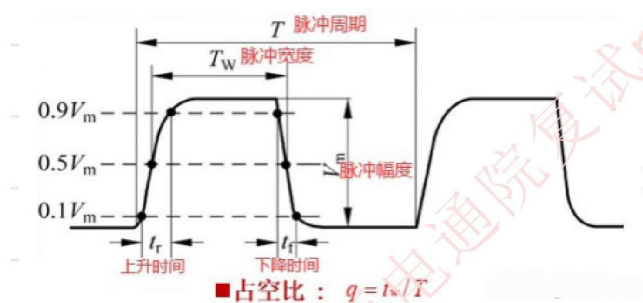
⇒ 根据上述列出状态表，再利用卡诺图进行化简，得到状态方程，并选择触发器，得出电路图

## 第七章 脉冲波形的产生与整形

1. 获得矩形脉冲的方法：

① 脉冲波形发生电路（多谐振荡器）

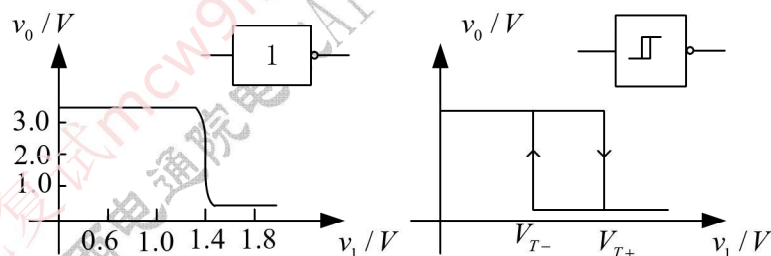
② 脉冲波形整形电路：利用整形电路把周期变化的波形变换为符合要求的矩形脉冲



2. 施密特触发器（一种特殊的数组传输门）

(1) 工作特点：

有两个阈值电压——滞回特性

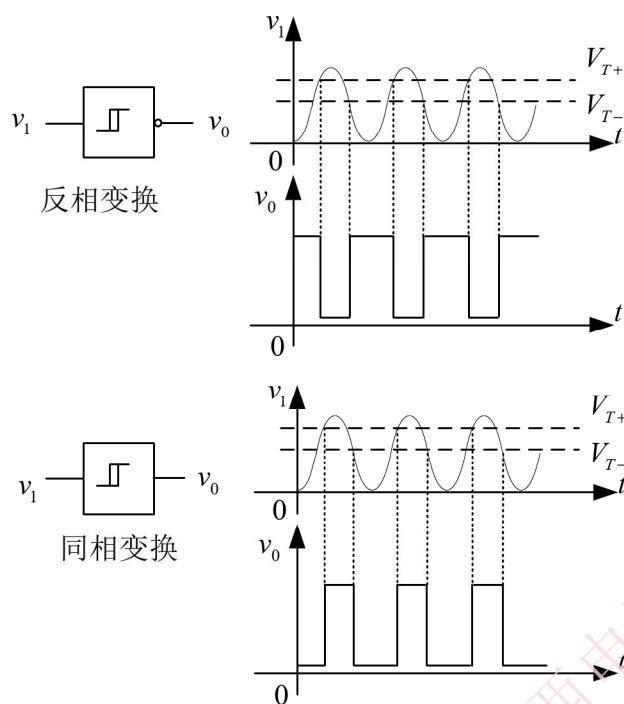


△ 输入从低电平上升的过程，输出发生状态转换时对应的输入电平称为正向阈值电压 ( $V_{T+}$ )

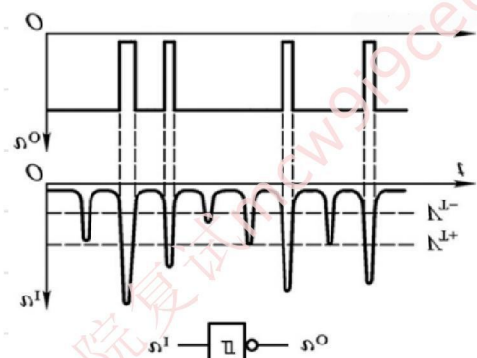
△ 输入从高电平下降的过程中，输出发生状态转换时对应的输入电平，称为负向阈值电压 ( $V_{T-}$ )

回差电压： $\Delta V_T = V_{T+} - V_{T-}$

波形变换：将周期性变化的波形转换为矩形脉冲



鉴幅：将幅度大于  $V_{T+}$  的脉冲信号选出



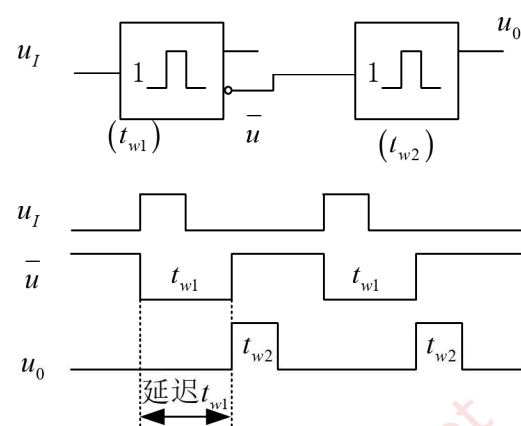
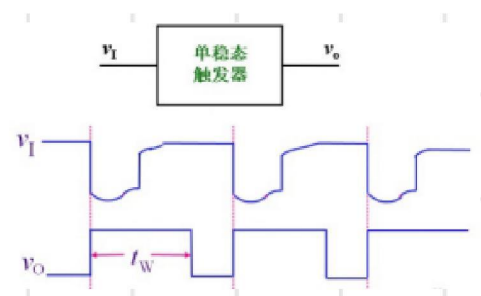
### 3. 单稳态触发器

电路具有稳态和暂稳态两个工作状态，在外界触发脉冲作用下，能由稳态暂时反转到暂稳态维持一段时间后自动返回稳态，暂稳态持续时间与电路自身参数有关，与外界触发脉冲无关

脉冲整形：能够把输入的不规则脉冲信号  $V_1$  整形为具有一定幅度和一定宽度的标准矩形脉冲  $V_0$ 。  $V_0$  的幅度取决于单稳态电路输出的高、低电平宽度  $t_w$  决定于电路元件的参数

脉冲延时：脉冲延时电路一般要用 2 个单稳态触发器完成。

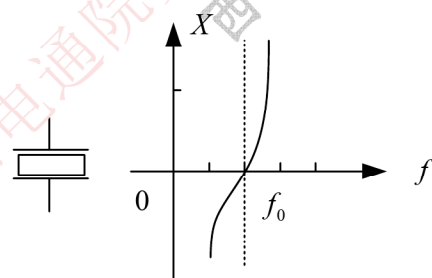
延时时间为 $t_w$ ，它决定与第一级单稳态触发器的定时元件。



#### 4.多谐振荡器

- ①无稳态，有2个暂稳态
- ②自激振荡器——在接通电源后，不需外加触发信号，便能自动产生矩形脉冲
- ③矩形波中除基波外，还含有丰富的谐波——称为多谐振荡器

#### Δ石英晶体多谐振荡器



多谐振荡器可以由多种电路结构实现，但往往存在频率稳定性差的弱点。在振荡电路中加入石英晶体，可以稳频，故称为石英晶体振荡器

晶体的选频特性极好：当频率为谐振频率 $f_0$ 时，石英晶体的等效阻抗最小，信号最容易通过，而其他频率信号均被衰减掉

输出频率稳态性高:  $f_0$  只与晶体管的材料, 几何形状, 尺寸大小有关, 而与电路中 RC 无关

#### 5.555 定时器

- (1) 多用途的数模混合集成电路, 内部有 3 个  $5k\Omega$  电阻
- (2)  $\left[ \begin{array}{l} \text{双极型产品: 最后数码为555, Eg } 5G555 \\ \text{CMOS型产品: 最后数码为7555, Eg } CC7555 \end{array} \right.$
- (3) 应用: 构成施密特触发器, 单稳态触发器, 多谐振荡器

## 第八章 可编程逻辑器件

任何组合逻辑电路都可以用“与门-或门”两级电路实现

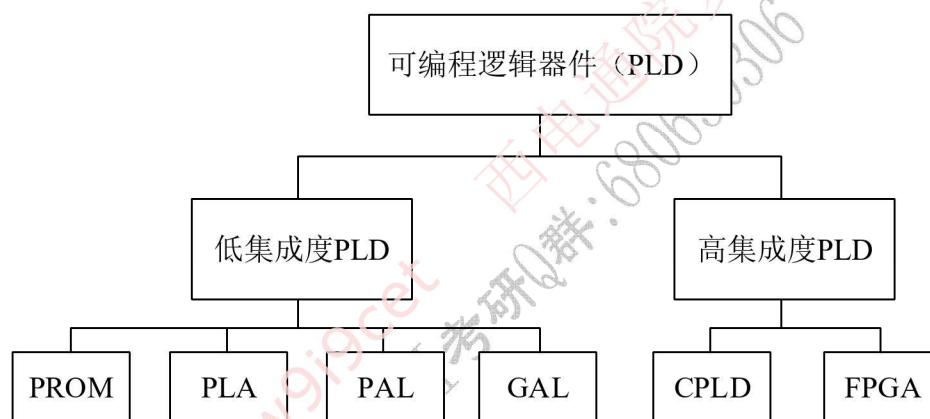
任何时序逻辑电路都可以用组合逻辑电路的存储电路实现

PLD (Programmable Logic Device)

发展历程:

PROM (可编程 ROM) → PLA (可编程逻辑阵列) → PAL (可编程逻辑阵列) → GAL

(通用阵列逻辑) → <sup>高密度</sup>EPLD (可擦除可编程) → CPLD (复杂可编程) → FPGA (现场可编程门阵列) → SPOC (片上可编程系统)



PLD 按集成度分类

低密度 PLD (简单 PLD/低集成度): 集成度小于每片 700 等效门, 包括 PROM, PLA, PAL, GAL。

高密度 PLD (复杂 PLD/高集成度): 集成度大于每片 1000 等效门。包括 CPLD 和 FPGA。

PLD 连接方式:   

断开      编程连接      固定连接 (硬连接)

四种低密度 PLD 结构特点:

| 类型   | 阵列 |     | 输出方式 |
|------|----|-----|------|
|      | 与  | 或   |      |
| PROM | 固定 | 可编程 | 固定   |

|     |     |     |     |
|-----|-----|-----|-----|
| PLA | 可编程 | 可编程 | 固定  |
| PAL | 可编程 | 固定  | 固定  |
| GAL | 可编程 | 固定  | 可编程 |

**MROM**（掩膜只读存储器）

- ①只读，不写
- ②非易失：存储的数据不因断电而消失

**PROM**（可编程只读存储器）

- ①实现组合逻辑函数，能写成最小项之和形式
- ②容量=与门数×或门数= $2^n \times m$

③缺点：逻辑门利用效率低

**EPROM**（紫外线可擦除形 PROM）

- ①通过紫外线照射擦除内容可重复擦除上万次
- ②离线擦除和写入需要专用设备

**E<sup>2</sup>PROM**（电可擦除形 PROM）

- ①电擦除可重复擦写 100 次、擦除速度快
- ②既具有 ROM 的非易失性又具备 RAM 的功能

**Flash Memory**（快闪存储器）

- ①数据的擦除和写入分开进行
- ②写入方式与 EPROM 相同，可擦除/写入 100W 次以上

**PLA**（可编程阵列逻辑）

- ①与/或阵列都可编程
- ②实现组合逻辑函数，用最简与或形式表示
- ③容量=与门数×或门数
- ④缺点：制造工艺复杂

**PAL**（可编程阵列逻辑）

- ①与阵列可编程、或阵列固定
- ②实现组合逻辑函数，用最简与或形式表示
- ③容量=门输入端数\*或门数

④缺点：输出包含的乘积项受限

GAL（通用阵列逻辑）

①基于 PAL 和  $E^2$ PROM 与阵列可编程或阵列固定

②输出端没有可编程的 OLMC（输出逻辑宏单元）

## 免责声明

晋程考研辅导班所建的 QQ 群和钉钉群仅供讨论考研、学习相关的内容，为营造良好的学习氛围，群成员不得发布或传播反动、暴力、恐怖言论等违法信息，不得有辱骂、诽谤、言语攻击他人以及其他妨碍他人学习交流的行为。请各位成员认真遵守上述纪律，群管理员有权将违反纪律者移除相应群并无条件终止服务。

该材料为晋程考研辅导班的内部赠送电子材料，仅供个人登录晋程考研提供的账号和密码线上观看学习或打印使用，该电子材料为非卖品，禁止盗版，翻印，传播等侵权行为。

以上声明最终解释权归晋程考研所有。