

数字逻辑概论

数制

二-十进制的转换

整数的转换

辗转相除法：将十进制数连续不断地除以2，直至商为零，所得余数由低位到高位排列，即为所求二进制数

当十进制数较大时可采用：

例1.2.3 将(133)D转换为二进制数

解：由于 2^7 为128，而 $133 - 128 = 5 = 2^2 + 2^0$ ，

所以对应二进制数 $b_7=1$, $b_2=1$, $b_0=1$, 其余各系数均为0, 所以得

$$(133)D = (10000101)B$$

小数的转换

b. 小数的转换:

对于二进制的小数部分可写成

$$(N)_D = b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + \cdots + b_{-(n-1)} \times 2^{-(n-1)} + b_{-n} \times 2^{-n}$$

将上式两边分别乘以2，得

$$2 \times (N)_D = b_{-1} \times 2^0 + b_{-2} \times 2^{-1} + \cdots + b_{-(n-1)} \times 2^{-(n-2)} + b_{-n} \times 2^{-(n-1)}$$

由此可见，将十进制小数乘以2，所得乘积的整数即为 b_{-1}

不难推知，将十进制小数每次除去上次所得积中的整数再乘以2，直到满足误差要求进行“四舍五入”为止，就可完成由十进制小数转换成二进制小数。

例 将十进制小数 $(0.39)_D$ 转换成二进制数，要求精度达到1%

解 由于精度要求达到1%，需要精确到二进制小数7位，即 $1/2^7=1/128$ 。

| | | | |
|------------------------|--------------|------------------------|--------------|
| $0.39 \times 2 = 0.78$ | $b_{-1} = 0$ | $0.24 \times 2 = 0.48$ | $b_{-5} = 0$ |
| $0.78 \times 2 = 1.56$ | $b_{-2} = 1$ | $0.48 \times 2 = 0.96$ | $b_{-6} = 0$ |
| $0.56 \times 2 = 1.12$ | $b_{-3} = 1$ | $0.96 \times 2 = 1.92$ | $b_{-7} = 1$ |
| $0.12 \times 2 = 0.24$ | $b_{-4} = 0$ | $0.92 \times 2 = 1.84$ | $b_{-8} = 1$ |

计算时要多算1位，然后考虑“4舍5入”。 $b_8 = 1$ 产生进位。

所以 $(0.39)_D = (0.0110010)_B$

二-十六进制的转换

二进制转换成十六进制：

因为16进制的基数 $16=2^4$ ，所以，可将四位二进制数表示一位16进制数，即 0000~1111 表示 0-F。

例 $(111\boxed{1000}\boxed{1010}\boxed{1110})_B = (78AE)_H$

十六进制转换成二进制：

将每位16进制数展开成四位二进制数，排列顺序不变即可。

例 $(BEEF)_H = (1011\boxed{1110}\boxed{1110}\boxed{1111})_B$

二-八进制的转换

4、二-八进制之间的转换

•因为八进制的基数 $8=2^3$ ，所以，可将三位二进制数表示一位八进制数，即 000~111 表示 0~7

•转换时，由小数点开始，整数部分自右向左，小数部分自左向右，三位一组，不够三位的添零补齐，则每三位二进制数表示一位八进制数。

例 $(10110.011)_B = (26.3)_O$

将每位八进制数展开成三位二进制数，排列顺序不变即可。

例 $(752.1)_O = (111\ 101\ 010.001)_B$

二进制算术运算

注 补码运算的结果仍为补码，需根据符号位还原

二进制代码

二-十进制码 (BCD码)

(1) 几种常用的BCD代码

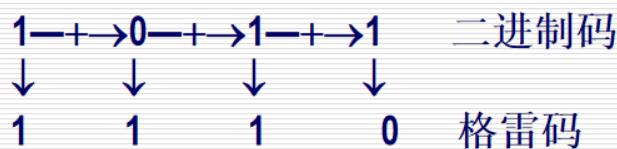
| BCD码十进制数码 | 8421码 | 2421码 | 5421码 | 余3码 | 余3循环码 |
|-----------|-------|-------|-------|------|-------|
| 0 | 0000 | 0000 | 0000 | 0011 | 0010 |
| 1 | 0001 | 0001 | 0001 | 0100 | 0110 |
| 2 | 0010 | 0010 | 0010 | 0101 | 0111 |
| 3 | 0011 | 0011 | 0011 | 0110 | 0101 |
| 4 | 0100 | 0100 | 0100 | 0111 | 0100 |
| 5 | 0101 | 1011 | 1000 | 1000 | 1100 |
| 6 | 0110 | 1100 | 1001 | 1001 | 1101 |
| 7 | 0111 | 1101 | 1010 | 1010 | 1111 |
| 8 | 1000 | 1110 | 1011 | 1011 | 1110 |
| 9 | 1001 | 1111 | 1100 | 1100 | 1010 |

注意 BCD码前后的0不可省略

格雷码

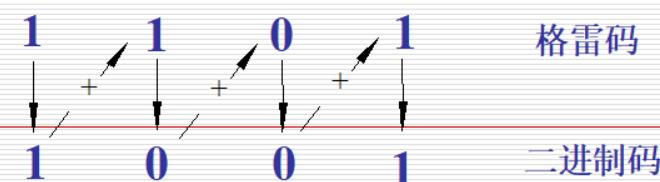
二进制码到格雷码的转换

- (1) 格雷码的最高位（最左边）与二进制码的最高位相同。
- (2) 从左到右，逐一将二进制码相邻的两位相加（舍去进位），作为格雷码的下一位。



格雷码到二进制码的转换

- (1) 二进制码的最高位（最左边）与格雷码的最高位相同。
- (2) 将产生的每一位二进制码，与下一位相邻的格雷码相加（舍去进位），作为二进制码的下一位。



基本逻辑运算

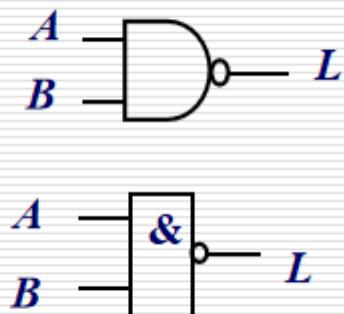
与非

1) 与非运算

与非逻辑符号

两输入变量与非
逻辑真值表

| A | B | L |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |



与非逻辑表达式

$$L = \overline{A \cdot B}$$

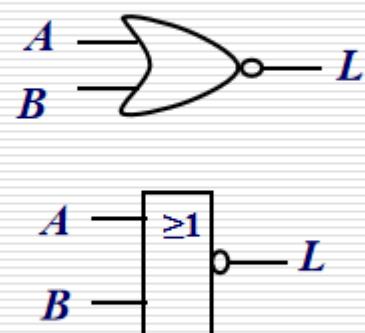
或非

或非运算

或非逻辑符号

两输入变量或非
逻辑真值表

| A | B | L |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |



或非逻辑表达式

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

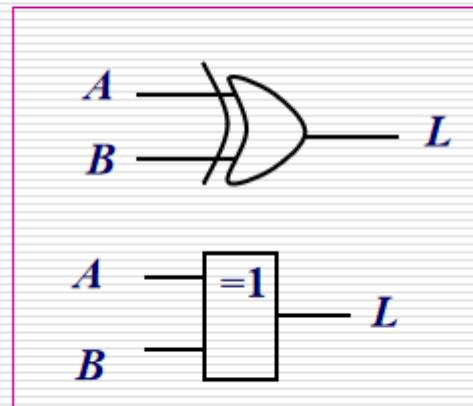
异或

若两个输入变量的值相异，输出为1，否则为0。

异或逻辑真值表

| A | B | L |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

异或逻辑符号



$$\text{异或逻辑表达式} \quad L = A \oplus B = \overline{AB} + A\overline{B}$$

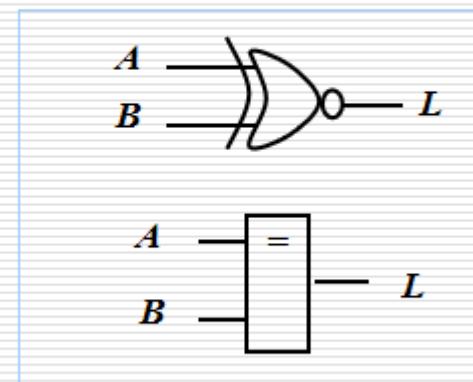
同或

若两个输入变量的值相同，输出为1，否则为0。

同或逻辑真值表

| A | B | L |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

同或逻辑逻辑符号



$$\text{同或逻辑表达式} \quad L = AB + \overline{A}\overline{B} = A \odot B$$

逻辑代数

$$\text{异或: } A \oplus B = \overline{AB} + A\overline{B}$$

注 多个变量异或时，奇数个1结果为1，偶数个1结果为0

$$\text{同或: } A \odot B = A \cdot B + \overline{A} \cdot \overline{B}$$

基本公式与恒等式

1、基本公式

0、1律: $A + 0 = A$ $A \cdot 1 = A$ $A \cdot 0 = 0$

互补律: $A + \bar{A} = 1$ $A \cdot \bar{A} = 0$

交换律: $A + B = B + A$ $A \cdot B = B \cdot A$

结合律: $A + B + C = (A + B) + C$ $A \cdot B \cdot C = (A \cdot B) \cdot C$

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

重叠律: $A + A = A$ $A \cdot A = A$

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

吸收律 $A + A \cdot B = A$ $A \cdot (A + B) = A$

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

其它常用恒等式

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

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

逻辑代数的基本规则

代入规则

1. 代入规则： 在包含变量A逻辑等式中，如果用另一个函数式代入式中所有A的位置，则等式仍然成立。这一规则称为代入规则。

例： $B(A + C) = BA + BC$,

用 $A + D$ 代替A，得

$$B[(A + D) + C] = B(A + D) + BC = BA + BD + BC$$

代入规则可以扩展所有基本公式或定律的应用范围

反演规则

2. 反演规则：

□ 对于任意一个逻辑表达式L，若将其中所有的与（•）换成或（+），或（+）换成与（•）；将1换成0，0换成1；原变量换为反变量，反变量换为原变量；则得到的结果就是原函数的反函数。非号以外的非号保持不变

例2.1.1 试求 $L = \overline{AB} + CD + 0$ 的非函数

解：按照反演规则，得

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

对偶规则

3. 对偶规则:

□ 对于任何逻辑函数式，若将其中的与（•）换成或（+），或（+）换成与（•）；并将1换成0，0换成1；那么，所得的新的函数式就是 L 的对偶式，记作 L' 。

例：逻辑函数 $L = (A + \bar{B})(A + C)$ 的对偶式为

$$L' = A\bar{B} + AC$$

□ 当某个逻辑恒等式成立时，则该恒等式两侧的对偶式也相等。这就是对偶规则。利用对偶规则，可从已知公式中得到更多的运算公式，例如，吸收律 $A + \bar{A} \cdot B = A + B$
 $A(\bar{A} + B) = AB$

逻辑函数表达式的形式

最小项与最小项表达式

1. 最小项的定义和性质

n 个变量 X_1, X_2, \dots, X_n 的最小项是 n 个因子的乘积，每个变量都以它的原变量或非变量的形式在乘积项中出现，且仅出现一次。一般 n 个变量的最小项应有 2^n 个。

例如， A, B, C 三个逻辑变量的最小项有（ $2^3 =$ ）8个，即

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

$\bar{A}B, A\bar{B}\bar{C}, A(B+C)$ 等则不是最小项。

最大项与最大项表达式

1. 最大项的定义和性质

n 个变量 X_1, X_2, \dots, X_n 的最大项是 n 个因子相或，每个变量都以它的原变量或非变量的形式在或项中出现，且仅出现一次。一般 n 个变量的最大项应有 2^n 个。

例如， A, B, C 三个逻辑变量的最大项有（ $2^3 =$ ）8个，即

$(\bar{A} + \bar{B} + \bar{C})$ 、 $(\bar{A} + \bar{B} + C)$ 、 $(\bar{A} + B + \bar{C})$ 、 $(\bar{A} + B + C)$ 、 $(A + \bar{B} + \bar{C})$ 、 $(A + \bar{B} + C)$ 、 $(A + B + \bar{C})$ 、 $(A + B + C)$

卡诺图

卡诺图的画法

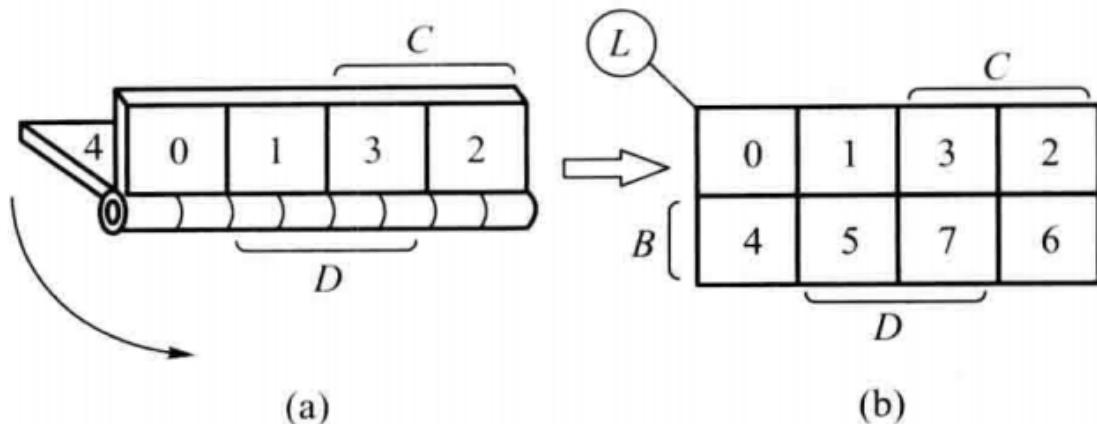


图 2.4.3 三变量卡诺图

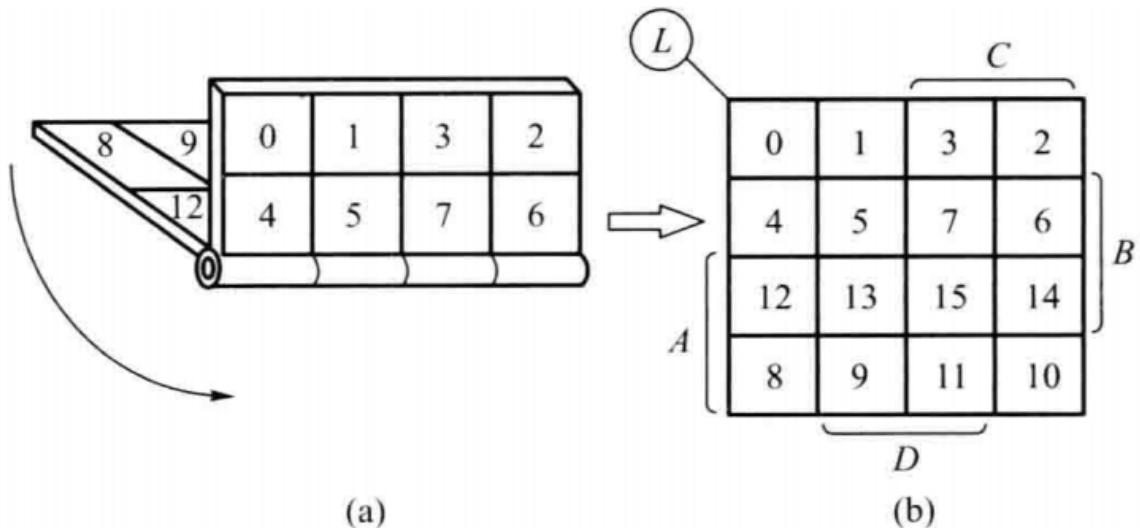


图 2.4.4 四变量卡诺图

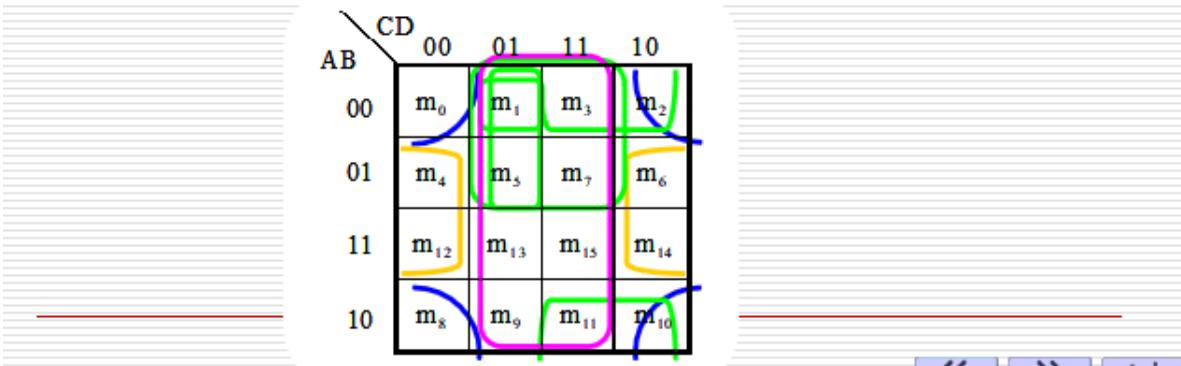
卡诺图化简的步骤

口用卡诺图化简逻辑函数的步骤如下：

- (1) 将逻辑函数写成最小项表达式
 - (2) 按最小项表达式填卡诺图，凡式中包含了的最小项，其对应方格填1，其余方格填0。
 - (3) 合并最小项，即将相邻的1方格圈成一组(包围圈)，每一组含 2^n 个方格，对应每个包围圈写成一个新的乘积项。本书中包围圈用虚线框表示。
 - (4) 将所有包围圈对应的乘积项相加。

画包围圈时应遵循的原则：

- (1) 包围圈内的方格数一定是 2^n 个，且包围圈必须呈矩形。
- (2) 循环相邻特性包括上下底相邻，左右边相邻和四角相邻。
- (3) 同一方格可以被不同的包围圈重复包围多次，但新增的包围圈中一定要有原有包围圈未曾包围的方格。
- (4) 一个包围圈的方格数要尽可能多，包围圈的数目要可能少



注 卡诺图化简的结果一般为与或式，若要得到或与式，可以圈出0然后将结果取非

具有无关项的化简

在真值表内对应于变量的某些取值下，函数的值可以是任意的，或者这些变量的取值根本不会出现，这些变量取值所对应的最小项称为无关项或任意项。

在含有无关项逻辑函数的卡诺图化简中，它的值可以取0或取1，具体取什么值，可以根据使函数尽量得到简化而定。

逻辑门电路

逻辑门电路简介

开关电路

基本CMOS逻辑门电路

CMOS反相器

其他基本CMOS逻辑门电路

CMOS与非门

CMOS或非门

CMOS传输门

CMOS逻辑门的不同输出结构及参数

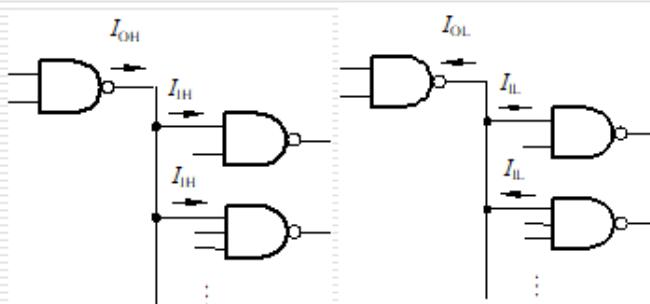
CMOS漏极开路和三态门电路

CMOS逻辑门电路的重要参数

6. 扇入数：输入端的个数；扇出数：在正常工作情况下，所能带同类门电路的最大数目

口 拉电流和灌电流

- ✓ 都是针对驱动门而言的，
- ✓ 逻辑门输出为高电平时的电流，此时电流由驱动门流向负载门，称为为拉电流
- ✓ 逻辑门输出为低电平时的负载电流，此时电流由驱动门流向负载门，称为灌电流
- ✓ 由于驱动门有内阻，拉电流和灌电流越大，驱动门的内电压越高，进而影响其输出电压



组合逻辑电路

组合逻辑电路的分析

组合逻辑电路的分析方法

一. 组合逻辑电路分析

根据已知逻辑电路，经分析确定电路的逻辑功能。

二. 组合逻辑电路的分析步骤：

- 1、由逻辑图写出各输出端的逻辑表达式；
- 2、化简和变换逻辑表达式；
- 3、列出真值表；
- 4、根据真值表或逻辑表达式，经分析最后确定其功能。

组合逻辑电路的设计

组合逻辑电路的设计过程

4.2.1 组合逻辑电路的设计过程

一、组合逻辑电路的设计：根据实际逻辑问题，求出所要求逻辑功能的最简单逻辑电路。

二、组合逻辑电路的设计步骤

- 1、**逻辑抽象**：根据实际逻辑问题的因果关系**确定输入、输出变量**，并定义逻辑状态的含义；
- 2、根据逻辑描述列出真值表；
- 3、由真值表写出逻辑表达式；
- 4、简化和变换逻辑表达式
- 5、画出逻辑图。

组合逻辑电路中的竞争与冒险

产生原因

竞争:当一个逻辑门的两个输入端的信号同时向相反方向变化，而变化的时间有差异的现象。

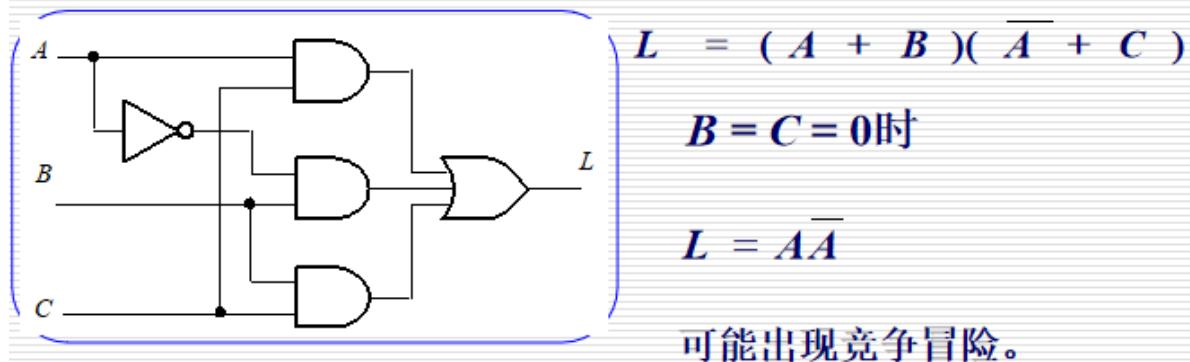
冒险:两个输入端的信号取值的变化方向是相反时，如门电路输出端的逻辑表达式简化成两个互补信号相乘或者相加，由竞争而可能产生输出干扰脉冲的现象。

消去竞争冒险的方法

注意 在卡诺图中，可能会产生竞争冒险的表现是圈出的圆出现“相切”的情况，此时只需在相切的位置添加一个圈即可消除竞争冒险

发现并消除互补变量

1. 发现并消除互补变量

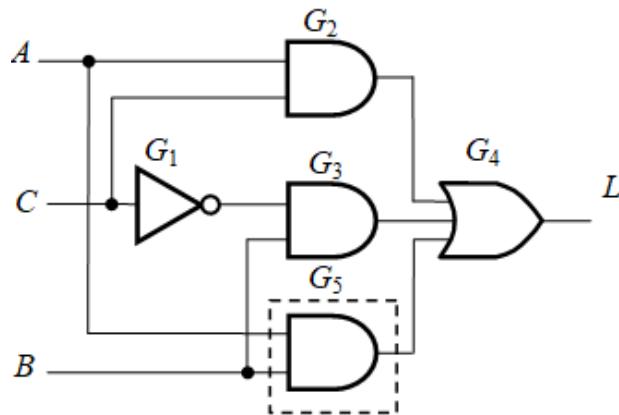


为消掉 $A\bar{A}$ ，变换逻辑函数式为

$$L = AC + \bar{A}B + BC$$

增加乘积项，避免互补项增加

2. 增加乘积项，避免互补项相加



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

当 $A=B=1$ 时

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

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

$$L = AC + B\bar{C} + AB$$

| | | 00 | 01 | 11 | 10 |
|-----|-----|----|----|----|----|
| | | 0 | 0 | 0 | 1 |
| | | 1 | 0 | 1 | 1 |
| L | B | 00 | 01 | 11 | 10 |

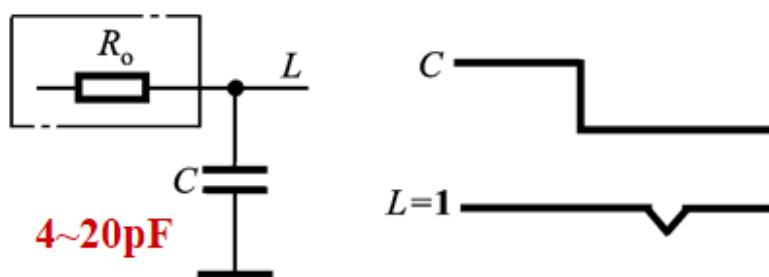
当 $A=B=1$ 时，根据逻辑表达式有

$$L = C + \bar{C} + 1$$

输出端并联电容器

3. 输出端并联电容器

如果逻辑电路在较慢速度下工作，为了消去竞争冒险，可以在输出端并联一电容器，致使输出波形上升沿和下降沿变化比较缓慢，可对于很窄的负跳变脉冲起到平波的作用。



若干典型的组合逻辑电路

编码器

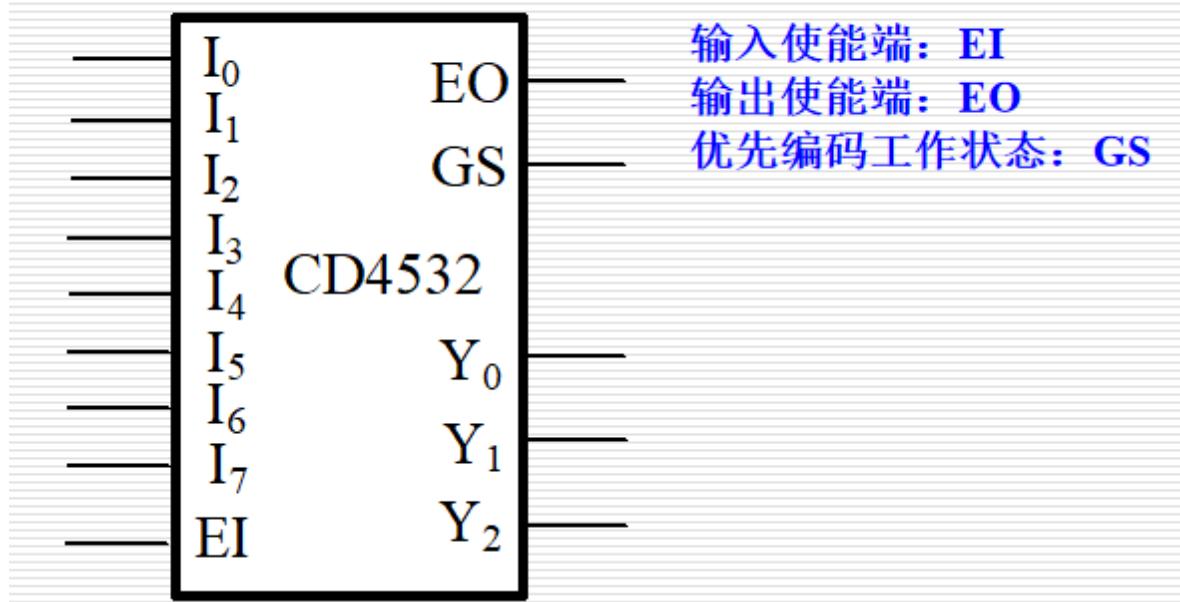
普通编码器

优先编码器

典型编码器电路

CD4532

8线3线—优先编码器CD4532的示意框图



优先编码器CD4532功能表

| 输入 | | | | | | | | | | 输出 | | | | |
|----|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----|----|--|
| EI | I ₇ | I ₆ | I ₅ | I ₄ | I ₃ | I ₂ | I ₁ | I ₀ | Y ₂ | Y ₁ | Y ₀ | GS | EO | |
| 0 | × | × | × | × | × | × | × | × | 0 | 0 | 0 | 0 | 0 | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | |
| 1 | 1 | × | × | × | × | × | × | × | 1 | 1 | 1 | 1 | 0 | |
| 1 | 0 | 1 | × | × | × | × | × | × | 1 | 1 | 0 | 1 | 0 | |
| 1 | 0 | 0 | 1 | × | × | × | × | × | 1 | 0 | 1 | 1 | 0 | |
| 1 | 0 | 0 | 0 | 1 | × | × | × | × | 1 | 0 | 0 | 1 | 0 | |
| 1 | 0 | 0 | 0 | 0 | 1 | × | × | × | 0 | 1 | 1 | 1 | 0 | |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | × | × | 0 | 1 | 0 | 1 | 0 | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | × | 0 | 0 | 1 | 1 | 0 | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | |

为什么要设计GS、EO输出信号？

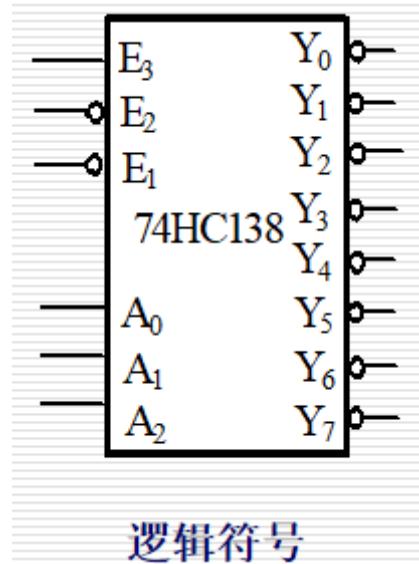
编码器禁止工作：EI=0；此时GS=0；EO=0

编码器允许工作：EI=1；

若所有输入端=0，则GS=0，EO=1>用于下一个编码器的输入EI；
若至少有一个输入端=1，GS=1，EO =0，

译码器/数据分配器

3线-8线译码器74HC138



3线-8线译码器（74HC138）功能表

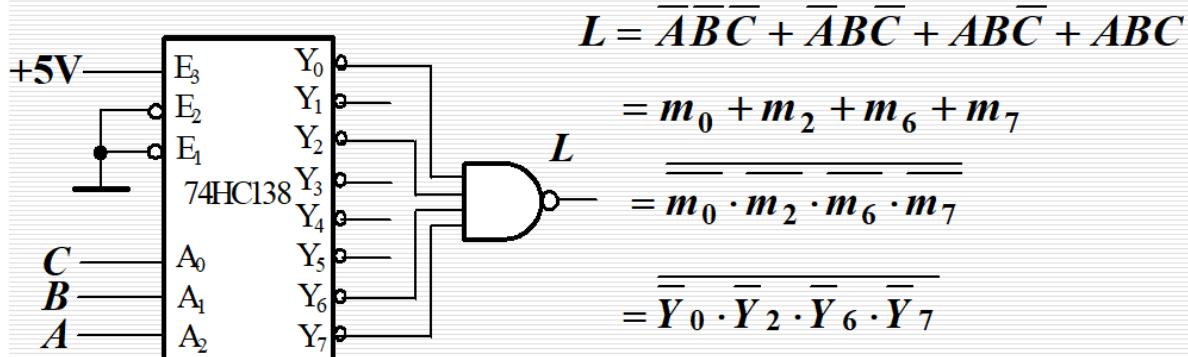
$E = E_3 \cdot \bar{E}_2 \cdot \bar{E}_1$ **E=0时禁止译码，即要求：** $E_3 = 1, , E_2 = 1, E_1 = 1$

| 输入 | | | | | | 输出 | | | | | | | |
|-------|-------------|-------------|-------|-------|-------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| E_3 | \bar{E}_2 | \bar{E}_1 | A_2 | A_1 | A_0 | \bar{Y}_0 | \bar{Y}_1 | \bar{Y}_2 | \bar{Y}_3 | \bar{Y}_4 | \bar{Y}_5 | \bar{Y}_6 | \bar{Y}_7 |
| x | 1 | x | x | x | x | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| x | x | 1 | x | x | x | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | x | x | x | x | x | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

用译码器实现逻辑函数

用一片74HC138实现函数 $L = \overline{A}\overline{C} + AB$

首先将函数式变换为最小项之和的形式

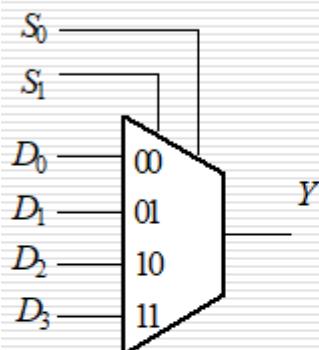


在译码器的输出端加一个与非门，即可实现给定的组合逻辑函数。

数据选择器

工作原理及逻辑功能

真值表



| 选择输入 | | 输出 |
|----------------|----------------|----------------|
| S ₁ | S ₀ | Y |
| 0 | 0 | D ₀ |
| 0 | 1 | D ₁ |
| 1 | 0 | D ₂ |
| 1 | 1 | D ₃ |

$$Y = \overline{S_1}\overline{S_0}D_0 + \overline{S_1}S_0D_1 + S_1\overline{S_0}D_2 + S_1S_0D_3$$

$$Y = D_0m_0 + D_1m_1 + D_2m_2 + D_3m_3$$

用数据选择器实现逻辑函数

(3) 数据选择器实现逻辑函数

例4.4.8 试用数据选择器实现下列逻辑函数

1、用4选1数据选择器实现 $L_0 = \overline{AB} + A\overline{B}$

2、用2选1数据选择器和必要的逻辑门实现 $L_1 = AB + A\overline{C} + BC$

3、用4选1数据选择器和必要的逻辑门实现 $L_1 = AB + AC + BC$

4选1数据选择器的输出与输入关系为：

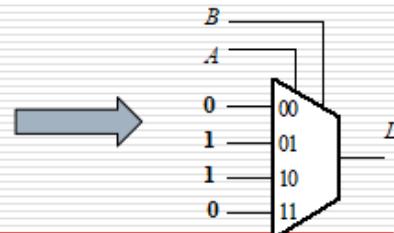
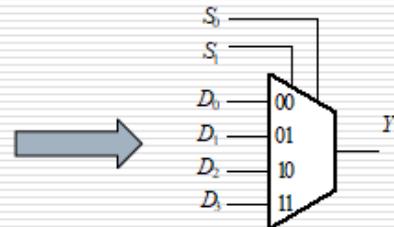
$$Y = \overline{S_1 S_0} D_0 + \overline{S_1} S_0 D_1 + S_1 \overline{S_0} D_2 + S_1 S_0 D_3$$

$$Y = D_0 m_0 + D_1 m_1 + D_2 m_2 + D_3 m_3$$

1、 $L_0 = \overline{AB} + A\overline{B}$

当 $S_1 = A$, $S_0 = B$, 有 $L_0 = \overline{S_1 S_0} + S_1 \overline{S_0}$

$$D_0 = D_3 = 0, \quad D_1 = D_2 = 1$$



利用数据选择器实现函数的一般步骤：

□ 变量数=选通端数

a、将函数变换成最小项表达式

b、函数的输入变量作为选通端 S_1 、 S_0

c、处理数据输入 $D_0 \sim D_7$ 信号电平。逻辑表达式中有 m_i , 则相应 $D_i = 1$, 其他的数据输入端均为 0。

□ 变量数>选通端数

✓ 多个数据选择器扩展使用

✓ 多余变量接输入端，并附加门电路实现

8选1数据选择器74HC151

8选1数据选择器74HC151的功能表

• 当 $\overline{E}=1$ 时， $Y=0$ 。

• 当 $\overline{E}=0$ 时

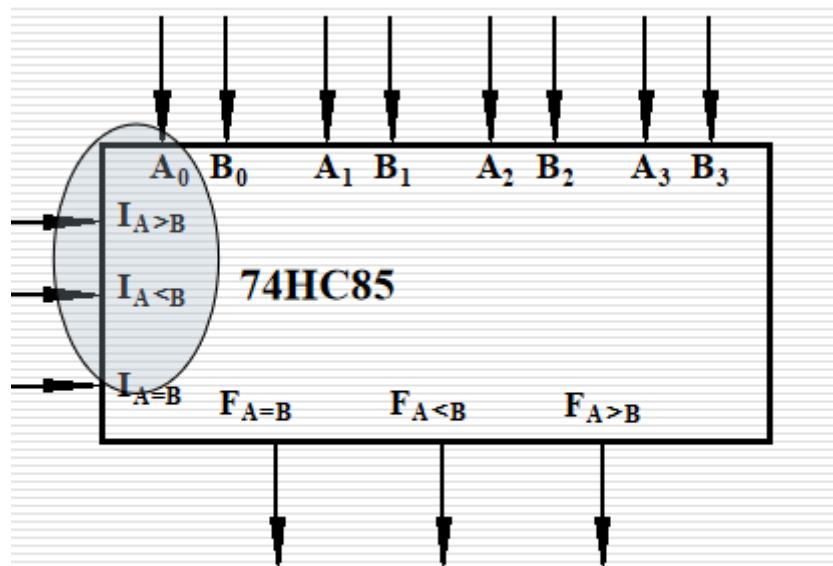
$$Y = \overline{S_2} \overline{S_1} S_0 D_0 + \overline{S_2} S_1 \overline{S_0} D_1 + \overline{S_2} S_1 \overline{S_0} D_2 \\ + \overline{S_2} S_1 S_0 D_3 + S_2 \overline{S_1} \overline{S_0} D_4 + S_2 \overline{S_1} S_0 D_5 \\ + S_2 S_1 \overline{S_0} D_6 + S_2 S_1 S_0 D_7$$

$$Y = \sum_{i=0}^7 D_i m_i$$

| 输入 | | | 输出 | | |
|----------------------|-------|-------|-------|-------|------------------|
| 使能 \overline{E} | 选择 | | | Y | \overline{Y} |
| | S_2 | S_1 | S_0 | | |
| 1 | X | X | X | L | H |
| 0 | 0 | 0 | 0 | D_0 | \overline{D}_0 |
| 0 | 0 | 0 | 1 | D_1 | \overline{D}_1 |
| 0 | 0 | 1 | 0 | D_2 | \overline{D}_2 |
| 0 | 0 | 1 | 1 | D_3 | \overline{D}_3 |
| 0 | 1 | 0 | 0 | D_4 | \overline{D}_4 |
| 0 | 1 | 0 | 1 | D_5 | \overline{D}_5 |
| 0 | 1 | 1 | 0 | D_6 | \overline{D}_6 |
| 0 | 1 | 1 | 1 | D_7 | \overline{D}_7 |

数值比较器

集成数值比较器74HC85



74HC85的示意框图

4位数值比较器74HC85的功能表

| 输入 | | | | | | | 输出 | | | | | | |
|-------------|-------|-------------|-------|-------------|-------|-------------|----------|-----------|-----------|-----------|-----------|-----------|-----------|
| A_3 | B_3 | A_2 | B_2 | A_1 | B_1 | A_0 | B_0 | $I_{A>B}$ | $I_{A<B}$ | $I_{A=B}$ | $F_{A>B}$ | $F_{A<B}$ | $F_{A=B}$ |
| $A_3 > B_3$ | | \times | | \times | | \times | | \times | \times | \times | 1 | 0 | 0 |
| $A_3 < B_3$ | | \times | | \times | | \times | | \times | \times | \times | 0 | 1 | 0 |
| $A_3 = B_3$ | | $A_2 > B_2$ | | \times | | \times | | \times | \times | \times | 1 | 0 | 0 |
| $A_3 = B_3$ | | $A_2 < B_2$ | | \times | | \times | | \times | \times | \times | 0 | 1 | 0 |
| $A_3 = B_3$ | | $A_2 = B_2$ | | $A_1 > B_1$ | | \times | | \times | \times | \times | 1 | 0 | 0 |
| $A_3 = B_3$ | | $A_2 = B_2$ | | $A_1 < B_1$ | | \times | | \times | \times | \times | 0 | 1 | 0 |
| $A_3 = B_3$ | | $A_2 = B_2$ | | $A_1 = B_1$ | | $A_0 > B_0$ | | \times | \times | \times | 1 | 0 | 0 |
| $A_3 = B_3$ | | $A_2 = B_2$ | | $A_1 = B_1$ | | $A_0 < B_0$ | | \times | \times | \times | 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 |
| $A_3 = B_3$ | | $A_2 = B_2$ | | $A_1 = B_1$ | | $A_0 = B_0$ | \times | \times | 1 | | 0 | 0 | 1 |
| $A_3 = B_3$ | | $A_2 = B_2$ | | $A_1 = B_1$ | | $A_0 = B_0$ | 1 | 1 | 0 | | 0 | 0 | 0 |
| $A_3 = B_3$ | | $A_2 = B_2$ | | $A_1 = B_1$ | | $A_0 = B_0$ | 0 | 0 | 0 | | 1 | 1 | 0 |

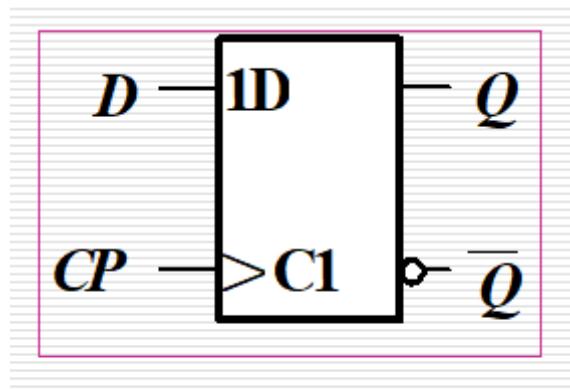
算数运算电路

锁存器和触发器

- 锁存器对脉冲电平敏感
- 触发器对时钟脉冲边沿敏感

触发器的逻辑功能

D触发器



D 触发器

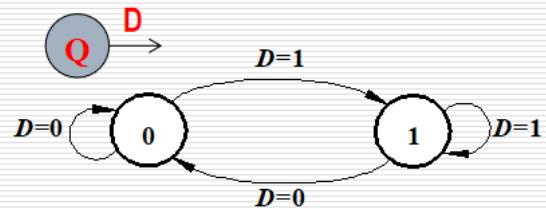
1. 特性表

| D | Q^n | Q^{n+1} |
|-----|-------|-----------|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

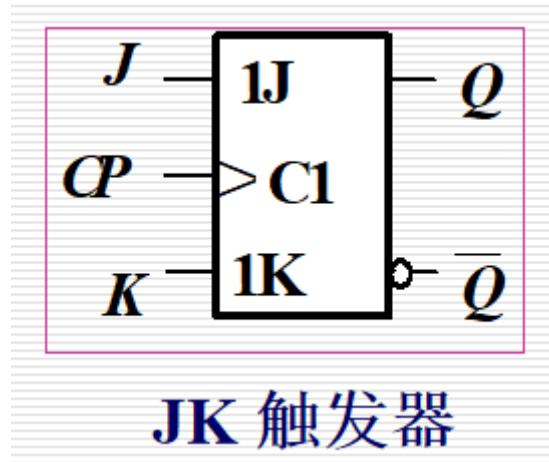
2. 特性方程

$$Q^{n+1} = D$$

3. 状态图



JK触发器

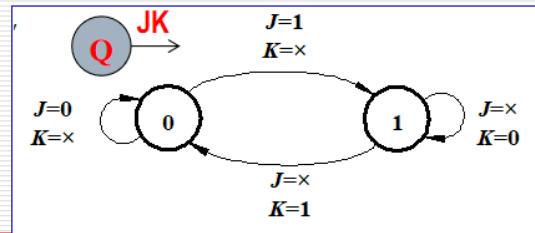


2. 特性方程

| J | K | Q^n | Q^{n+1} | |
|-----|-----|--------|-----------|------|
| | | KQ^n | | |
| | | 00 | 01 | 11 |
| 0 | 0 | 0 | 0 | |
| 0 | 0 | 1 | 1 | |
| 0 | 1 | 0 | 0 | 状态不变 |
| 0 | 1 | 1 | 0 | |
| 1 | 0 | 0 | 1 | 置 0 |
| 1 | 0 | 1 | 1 | |
| 1 | 1 | 0 | 1 | 置 1 |
| 1 | 1 | 1 | 0 | |
| 1 | 1 | 1 | 0 | 翻 转 |
| 1 | 1 | 0 | 1 | |

$$Q^{n+1} = J\bar{Q}^n + \bar{K}Q^n$$

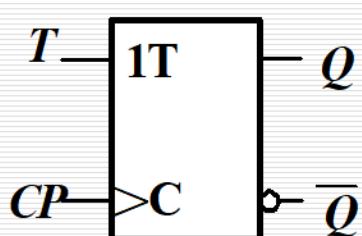
3. 状态转换图



JK触发器：00不变，11翻转，01、10同

T触发器

逻辑符号



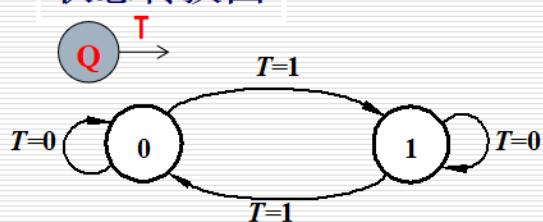
特性表

| T | Q^n | Q^{n+1} |
|-----|-------|-----------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

特性方程

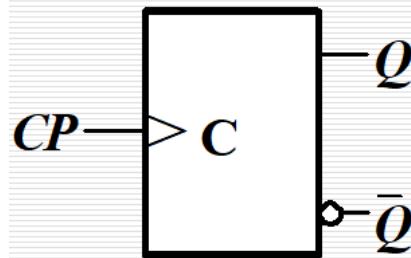
$$Q^{n+1} = T\bar{Q}^n + \bar{T}Q^n$$

状态转换图



T' 触发器

逻辑符号



特性方程

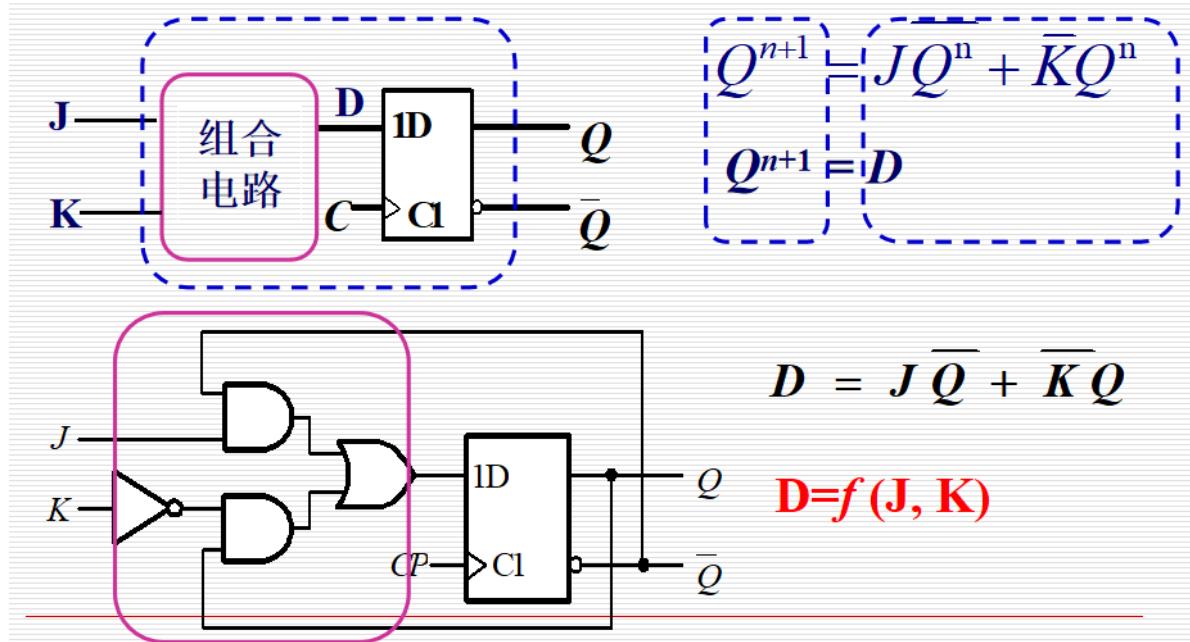
$$Q^{n+1} = \overline{Q^n}$$

时钟脉冲每作用一次，触发器翻转一次。

D触发器功能的转换

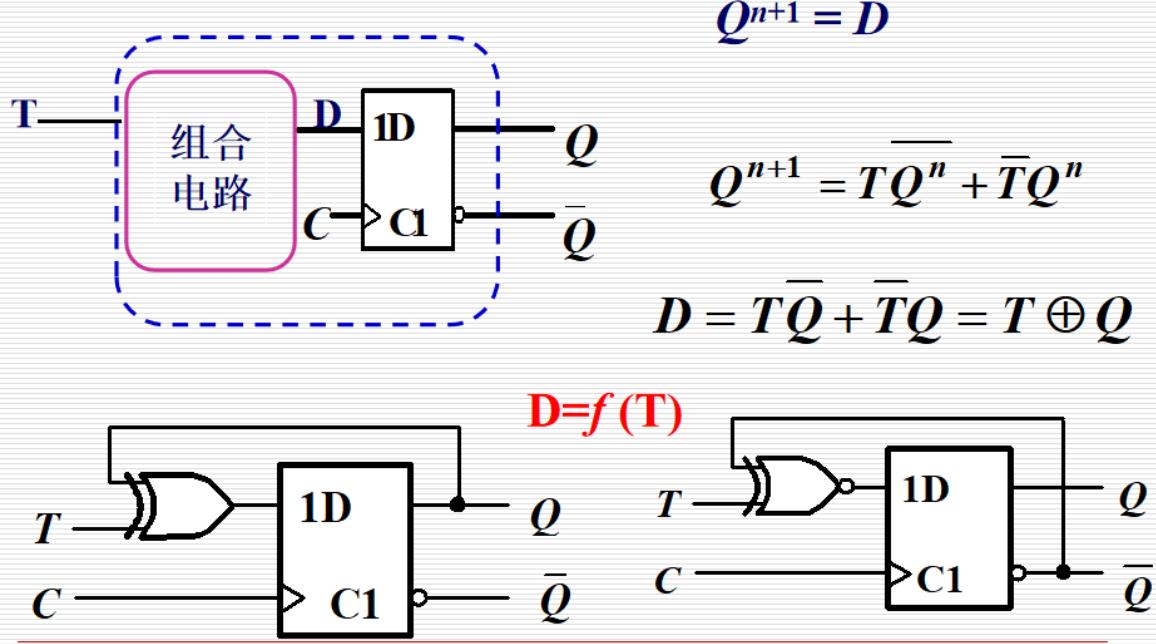
D触发器构成JK触发器

1. D 触发器构成 JK 触发器



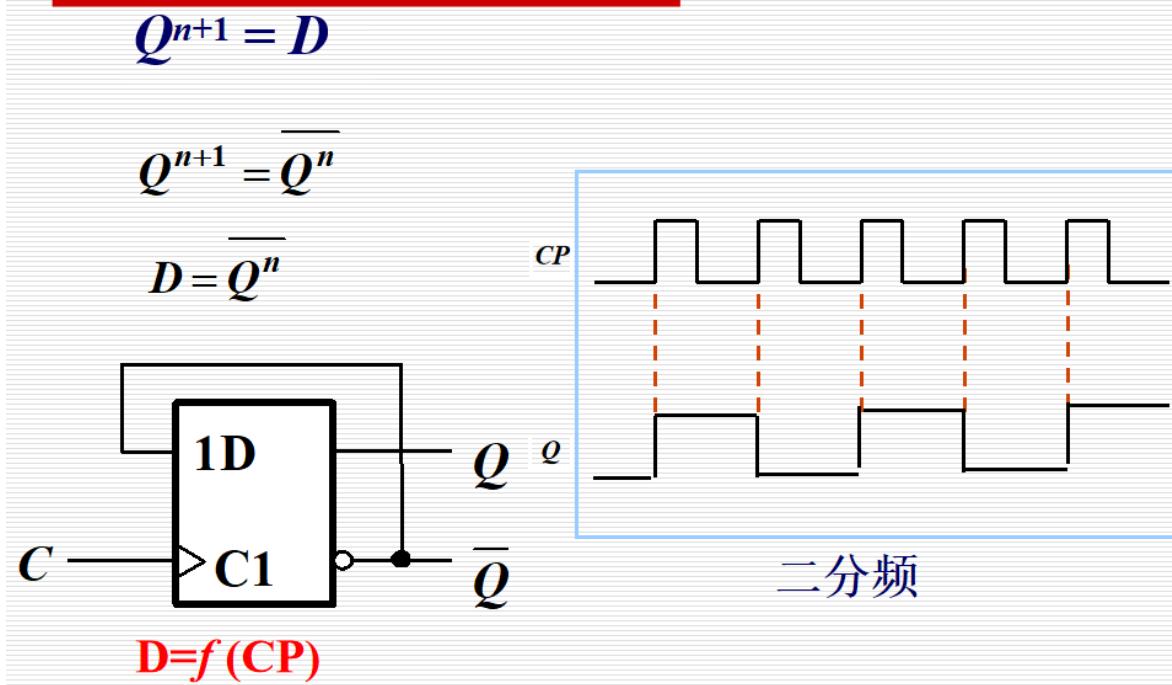
D触发器构成T触发器

2. D 触发器构成 T 触发器



D触发器构成T'触发器

3. D 触发器构成 T' 触发器



时序逻辑电路

时序逻辑电路的基本概念

时序逻辑电路的基本结构及分类

异步时序电路与同步时序电路

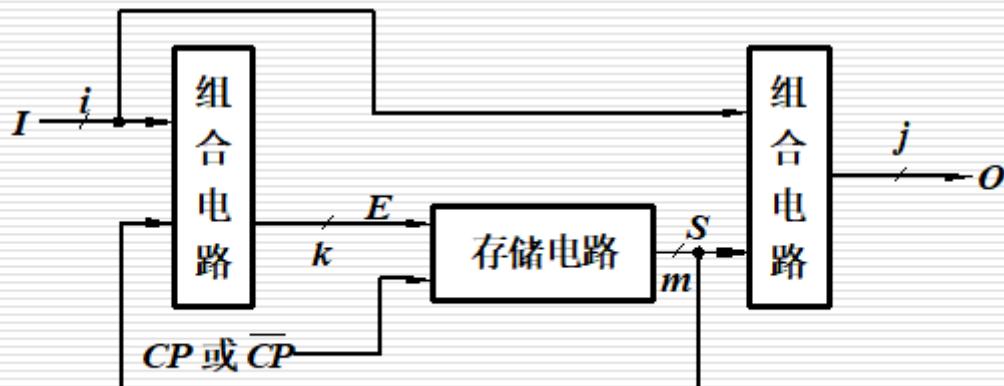
- 同步时序电路：存储电路里所有触发器有一个统一的时钟源，它们的状态在同一时刻更新
- 异步时序电路：没有统一的时钟脉冲或没有时钟脉冲，电路的状态更新不是同时发生的

米利型电路和穆尔型电路

米利型电路

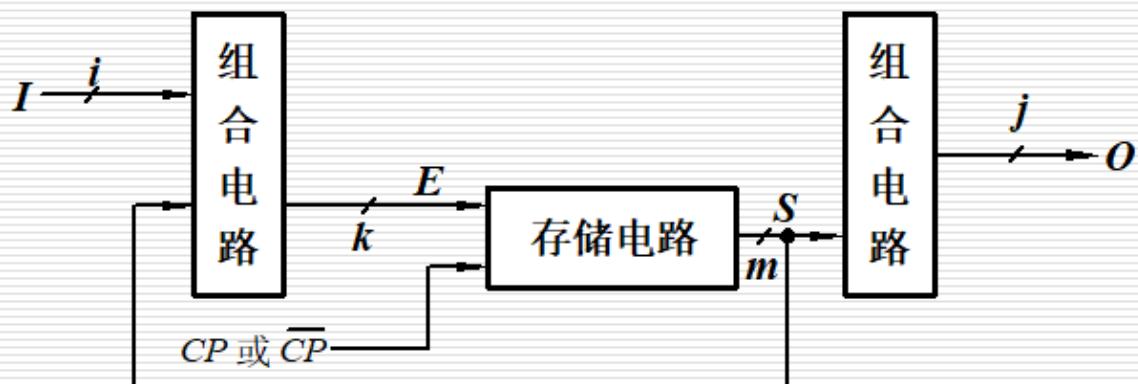
电路的输出是输入变量*A*及触发器输出 Q_1 、 Q_0 的函数，

这类时序电路亦称为米利型电路



穆尔型电路

电路输出仅仅取决于各触发器的状态，而不受电路当时的输入信号影响或没有输入变量，这类电路称为穆尔型电路



时序逻辑电路功能的表达

同步时序逻辑电路的分析

- 电路状态表
- 状态图
- 工作波形图

注意 米利型和穆尔型输出的状态图有所不同，[米利型输出标在方向线旁](#)，[穆尔型标在圆圈状态名旁](#)。

分析同步时序逻辑电路的一般步骤

1. 了解电路的组成:

电路的输入、输出信号、触发器的类型等

2. 根据给定的时序电路图,写出下列各逻辑方程式:

(1) 输出方程;

(2) 各触发器的激励方程;

(3) 状态方程: 将每个触发器的驱动方程代入其特性方程得状态方程.

3. 列出状态转换表或画出状态图和波形图;

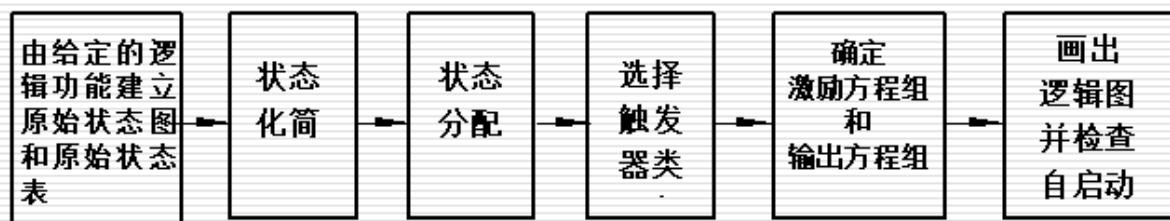
4. 确定电路的逻辑功能.

同步时序逻辑电路的分析一般还用到[状态转换真值表](#)。

同步时序逻辑电路的设计

设计同步时序逻辑电路的一般步骤

同步时序电路的设计过程



(1) 根据给定的逻辑功能建立原始状态图和原始状态表

- ① 明确电路的输入条件和相应的输出要求，分别确定输入变量和输出变量的数目和符号。
- ② 找出所有可能的状态和状态转换之间的关系。
- ③ 根据原始状态图建立原始状态表。

(2) 状态化简---求出最简状态图

合并等价状态，消去多余状态的过程称为状态化简

等价状态：在相同的输入下有相同的输出，并转换到同一个次态去的两个状态称为等价状态。

(3) 状态编码（状态分配）

给每个状态赋以二进制代码的过程。

根据状态数确定触发器的个数，

$$2^{n-1} < M \leq 2^n \quad (M: \text{状态数}; n: \text{触发器的个数})$$

(4) 选择触发器的类型

(5) 求出电路的激励方程和输出方程

(6) 画出逻辑图并检查自启动能力

JK激励表

| Q^n | Q^{n+1} | J | K |
|-------|-----------|-----|-----|
| 0 | 0 | 0 | d |
| 0 | 1 | 1 | d |
| 1 | 0 | d | 1 |
| 1 | 1 | d | 0 |

异步时序逻辑电路的分析

分析方法：

一. 异步时序逻辑电路的分析方法：

分析步骤：

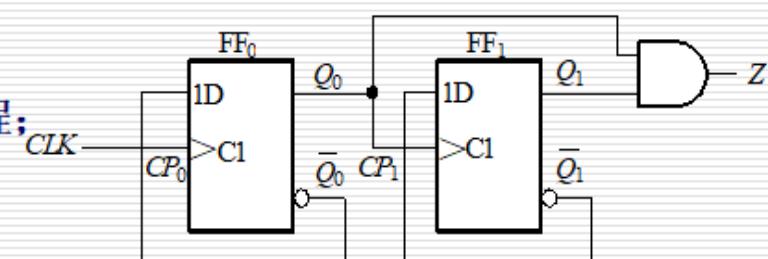
1. 写出下列各逻辑方程式：

a) 时钟方程

b) 触发器的激励方程；

c) 输出方程

d) 状态方程



2. 列出状态转换表或画出状态图和波形图；

3. 确定电路的逻辑功能。

注意：

- (1) 分析状态转换时必须考虑各触发器的时钟信号作用情况
有作用，则令 $cp_n=1$ ；否则 $cp_n=0$
根据激励信号确定那些 $cp_n=1$ 的触发器的次态， $cp_n=0$ 的触发器则保持原有状态不变。
- (2) 每一次状态转换必须从输入信号所能触发的第一个触发器开始逐级确定
- (3) 每一次状态转换都有一定的时间延迟
同步时序电路的所有触发器是同时转换状态的，与之不同，异步时序电路各个触发器之间的状态转换存在一定的延迟，也就是说，从现态 S^n 到次态 S^{n+1} 的转换过程中有一段“不稳定”的时间。在此期间，电路的状态是不确定的。只有当全部触发器状态转换完毕，电路才进入新的“稳定”状态，即次态 S^{n+1} 。

注 若D触发器有时钟脉冲的上升沿作用时，其状态变化；无时钟脉冲上升沿作用时，其状态不变，则有如下常用公式：

$$Q^{n+1} = D \cdot cp + Q^n \cdot \overline{cp}$$

若干典型的时序逻辑电路

寄存器和移位寄存器

寄存器

移位寄存器

双向移位寄存器74HC194

表 6.5.4 多功能 4 位双向移位寄存器 74HC/HCT194 的功能表

| 清零 | 控制信号 | 时钟 <i>CP</i> | 输入 | | | | 输出 | | | | 功 能 | | |
|----|------|-----------------|-------------|-------------|----------|------------|-------------|-------------|-------------|-------------|----------|----------|----------|
| | | | 串行输入 | | 并行输入 | | Q_0^{n+1} | Q_1^{n+1} | Q_2^{n+1} | Q_3^{n+1} | | | |
| | | | 右移 D_{SR} | 左移 D_{SL} | D_{10} | D_{11} | D_{12} | D_{13} | | | | | |
| L | × | × | × | × | × | × | × | × | L | L | L | 异步清零 | |
| H | L | L | × | × | × | × | × | × | Q_0^n | Q_1^n | Q_2^n | 保持 | |
| H | L | H | ↑ | L | × | × | × | × | L | Q_0^n | Q_1^n | 右移 | |
| H | L | H | ↑ | H | × | × | × | × | H | Q_0^n | Q_1^n | 右移 | |
| H | H | L | ↑ | × | L | × | × | × | Q_1^n | Q_2^n | Q_3^n | 左移 | |
| H | H | L | ↑ | × | H | × | × | × | Q_1^n | Q_2^n | Q_3^n | 左移 | |
| H | H | H | ↑ | × | × | D_{10}^* | D_{11}^* | D_{12}^* | D_{13}^* | D_{10} | D_{11} | D_{12} | D_{13} |

注： D_{1N}^* 表示 CP 脉冲上升沿之前瞬间 D_{1N} 的电平。

注意

- 右移：从高位向低位
- 左移：从低位

计数器

二进制计数器

74161

表 6.5.6 74LVC161 的功能表

| 清零 | 预置 <i>PE</i> | 输入 | | | | 输出 | | | | 进位 <i>TC</i> | | | |
|-----------|-----------------|------------|------------|-----------------|---------|---------|---------|---------|-------|-----------------|-------|-------|---|
| | | 使能 | | 时钟 <i>CP</i> | 预置数据输入 | | | | Q_3 | Q_2 | Q_1 | Q_0 | |
| | | <i>CEP</i> | <i>CET</i> | | D_3 | D_2 | D_1 | D_0 | | | | | |
| 异步清零 L | × | × | × | × | × | × | × | × | L | L | L | L | L |
| 同步置数 H | L | × | × | ↑ | D_3^* | D_2^* | D_1^* | D_0^* | D_3 | D_2 | D_1 | D_0 | # |
| H | H | L | × | × | × | × | × | × | 保 | 持 | | | # |
| H | H | × | L | × | × | × | × | × | 保 | 持 | | | L |
| H | H | H | H | ↑ | × | × | × | × | 计 | 数 | | | # |

注： D_N^* 表示 CP 脉冲上升沿之前瞬间 D_N 的电平。#表示只有当 $Q_3 Q_2 Q_1 Q_0 \cdot CET = 1$ (正逻辑体系) 时, TC 输出为高电平, 其余均为低电平。

说明

- \overline{CR} 的作用：异步清零（强制清零，与 CP 无关，低电平有效）
- \overline{PE} 的作用：同步置数（与 CP 有关，低电平有效）
- 进位信号 TC 只有在 $Q_3 Q_2 Q_1 Q_0 = 1111$ 且 $CET = 1$ 时输出为 1, 其余时间均为 0, 可简单记为：

$$TC = CET \cdot Q_3Q_2Q_1Q_0$$

(考点) 利用74161实现任意进制计数器

用集成计数器构成任意进制计数器的一般方法

1) $N < M$ 的情况：

(已有的集成计数器是 M 进制，需组成的是 N 进制计数器)

实现的方法：

反馈清零法

利用清零输入端，使电路计数到某状态时产生清零操作，清除 $M-N$ 个状态实现 N 进制计数器。

反馈置数法

利用计数器的置数功能，通过给计数器重复置入某个数码的方法减少 $(M-N)$ 个独立状态，实现 N 进制计数器。

2) $N > M$ 的情况

实现的方法：采用多片 M 进制计数器构成。

按芯片连接方式可分为：

(1) 串行进位方式：构成异步计数器

(2) 并行进位方式：构成同步计数器

实现方法（注意异步清零、同步置数）：

- 反馈清零
- 反馈置数

- 整体反馈法（多片连接）
- 级联法（+ 置数）（多片连接，当进制数可以拆成两个数的乘积时使用）

环形计数器

- 若一个计数器可被分为两个独立的子计数器 (m, n 进制)，则该计数器的进制为 $m \times n$

半导体存储器

小结

- ROM是一种非易失性存储器，存储固定数据，一般只读；根据写入方式不同，分为**固定ROM**和**可编程ROM**（PROM）
- PROM又可以分为：一次性PROM，光可擦除EPROM，电可擦除E²PROM，和快闪存储器；
- RAM是一种时序逻辑电路，具有记忆功能，存储的数据断电后消失，是一种易失性可读写存储器，包括：静态SRAM和动态DRAM两种；SRAM用锁存器记忆数据，DRAM用MOS管加栅极电容存储数据；
- SRAM和DRAM都有同步RAM：SSRAM和SDRAM；
- DDR SDRAM：双倍数据率同步动态RAM；DDR II

脉冲波形的变换与产生

单稳态触发器

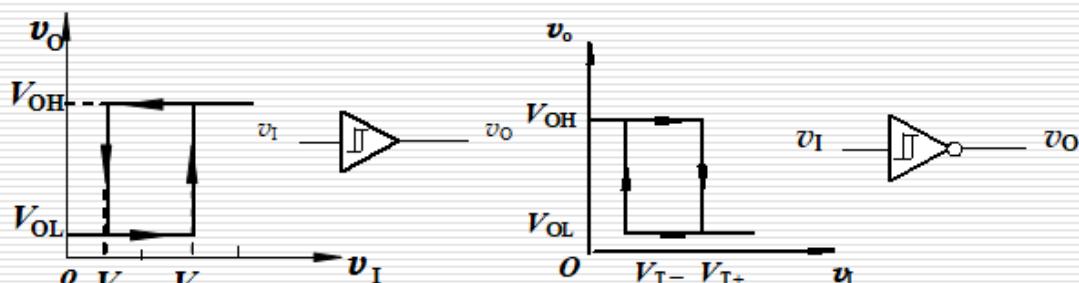
单稳态触发器的工作特点：

- ① 电路在没有触发信号作用时处于一种稳定状态。
- ② 在外来触发信号作用下，电路由稳态翻转到暂稳态；
- ③ 由于电路中 RC 延时环节的作用，暂稳态不能长保持，经过一段时间后，电路会自动返回到稳态。
暂稳态的持续时间仅取与 RC 参数值有关。

施密特触发器

施密特触发器电压传输特性及工作特点：

- ① 施密特触发器属于电平触发器件，当输入信号达到某一定电压值时，输出电压会发生突变。
- ② 电路有两个阈值电压。输入信号增加和减少时，电路的阈值电压分别是正向阈值电压 (V_{T+}) 和负阈值电压 (V_{T-})。



模数与数模转换器

D/A转换器

1、分辨率

- ✓ 分辨率：其定义为D/A转换器模拟输出电压可能被分离的等级数。n位DAC最多有 2^n 个模拟输出电压。位数越多D/A转换器的分辨率越高。
- ✓ 分辨率也可以用能分辨的最小输出电压与最大输出电压之比给出。
$$\text{分辨率} = V_{\text{LSB}}/V_m = \frac{1}{2^n - 1}$$

V_{LSB} 为输入数字量中最低有效位变化所引起的输出电压的变化值； V_m 为最大输出电压

2、转换精度：

- ✓ 转换精度是指对给定的数字量，D/A转换器实际值与理论值之间的最大偏差。
- ✓ 产生原因：由于D/A转换器中各元件参数值存在误差，如基准电压不够稳定或运算放大器的零漂等各种因素的影响。
- ✓ 几种转换误差：比例系数误差、失调误差和非线性误差等

A/D转换器

1. 转换精度

单片集成A/D转换器的转换精度是用分辨率和转换误差来描述的。

□ 分辨率：

- ✓ 说明A/D转换器对输入信号的分辨能力。通常以输出二进制(或十进制)数的位数表示。
- ✓ 输入电压一定时，输出位数越多，分辨率越高，例如：输入信号最大值为5V，则分辨率为8位二进制时，能区分出的输入信号最小值为 $5/2^8=19.53\text{mV}$

□ 转换误差：

- ✓ 表示A/D转换器实际输出的数字量和理论上的输出数字量之间的差别。

2. 转换时间

- ✓ 指A/D转换器从转换控制信号到来开始，到输出端得到稳定的数字信号所经过的时间。 A/D转换器的转换时间与转换电路的类型有关

| | |
|----------------|-------------|
| 并行比较A/D转换器(8位) | <50ns |
| 逐次比较型A/D转换器 | 10~50μs |
| 间接A/D转换器 | 10ms~1000ms |