

1 数字逻辑概论

1 数字信号描述方法

电信号 { 模拟电信号：连续
 数字电信号：离散

数字信号的描述方法

(1) 0, 1 数码 { 表示数量时称二进制数

 表示事物状态时称二值逻辑

(2) 逻辑电平 { 高电平：电压 $V_{H(\min)} \sim +V_{DD}$ 1
 低电平：电压 0 ~ $V_{L(\max)}$ 0

(3) 数字波形



一般采用二值数字逻辑

实际脉冲波形及主要参数

① 幅值

② 上升时间 t_r (ns) 10% V_m - 90% V_m 所经历的时间

③ 下降时间 t_f (ns)

④ 周期 T 频率 $f = \frac{1}{T}$

⑤ 脉冲宽度 t_w 上升 50% V_m - 下降 50% V_m

⑥ 占空比 $q = \frac{t_w}{T} \times 100\%$

2 数制

(1) 数制：计数规则（构成方法+进位规则）

十进制：采用 0~9 数码，“逢 10 进 1”

推广 ↓ $55.316 = 5 \times 10^1 + 5 \times 10^0 + 3 \times 10^{-1} + 1 \times 10^{-2} + 6 \times 10^{-3}$

- 一般表达式 $(N)_r = \sum_{i=-m}^{\infty} \underline{k_i} \times \underline{10^i}$
 系数 位权

R 进制：以 R 为基数的计数体制

采用 $0 \sim R-1$ 数码，“逢 R 进 1”

$$N_R = \sum_{i=0}^{\infty} (k_i \times R^i)$$

二进制：多位二进制小节中每一个数码称为 1 位 (1 bit)

Binary 8 位二进制数称为一个字节 (Byte)

e.g.

$$\begin{array}{cccccccccc} & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ & 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \end{array}$$

MSB (Most Significant Bit)

LSB

$$(N)_B = \sum_{i=0}^{\infty} (k_i \times 2^i)$$

Octal 八进制

$$(N)_D = \sum_{i=0}^{\infty} (k_i \times 8^i)$$

十六进制

0 1 2 3 4 5 6 7 8 9 A B C D E F

Hexadecimal

$$(N)_{16} = \sum_{i=0}^{\infty} (k_i \times 16^i)$$

(2) 二十进制数转换

$$= \rightarrow + \quad \sum_{i=0}^{\infty} (k_i \times 2^i)$$

$$+ \rightarrow =$$

① 整数部分

加权求和法

确定一组二进制权使它们的和等于已知的十进制数

在计算机中 $2^{10} - 1K$ $2^{20} - 1M$

$2^{30} - 1G$ $2^{40} - 1T$

$$\text{e.g. } (9)_D = 8 + 1 = 1 \times 2^3 + 1 \times 2^0 = 1001$$

重复除以 2 取余法

$$\begin{array}{r} 2 | 45 \\ 2 | 22 \\ 2 | 11 \\ 2 | 5 \\ 2 | 2 \\ 2 | 1 \\ \hline 0 \end{array}$$

$$\begin{array}{r} \text{余 } 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ \hline (45)_D = (101101)_B \end{array}$$

② 小数部分

加权求和法

$$\text{e.g. } (0.625)_{10} = 0.5 + 0.125 = 2^{-1} + 2^{-3} = (0.101)_B$$

重複乘以2取整法

$$(0.3125)_D = (0.0101)_B$$

e.g.

$$0.3125 \times 2 = 0.625$$

取 0

$$0.625 \times 2 = 1.25$$

1

$$0.25 \times 2 = 0.5$$

0

$$0.5 \times 2 = 1.0$$

1

e.g. 要求转换误差小于 1%

$$\text{即 } 2^{-m} \leq 1\% \Rightarrow m \geq \frac{2}{\lg 2} = 6.64$$

即转换到小数点后第 7 位

(3) 其他不同数制之间的转换

$\text{二} \longrightarrow \text{十六}$ ：将 4 位二进制看作一个整体

以二进制的小数点为基准，

将小数点左边的整数从右到左每 4 个分为一组；

… 右 … 从左到右 …

将每组数以一个十六位数代替

$$\text{e.g. } \text{二} = \underline{1101} \underline{1001} \underline{1011} \underline{0011} . \underline{0100}$$

十六 D 9 B 3 . 4

十 \longrightarrow 十六

方法①：十 \rightarrow 二 \longrightarrow 十六

方法②：仿照“十 \rightarrow 二”的方法

$$\begin{array}{r} 16 \longdiv{650} \\ 16 \longdiv{40} \\ 16 \longdiv{2} \\ \hline 0 \end{array}$$

$$\begin{array}{r} \text{余 A} \\ 8 \\ 2 \end{array}$$

$$(650)_D = (28A)_H$$

3 二进制数的算术运算

(1) 二进制数的算术运算

① 加法运算

和十进制类似

“逢二进一”

② 减法运算

③ 乘法运算

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1$$

④ 除法运算

$$0 \div 1 = 0$$

$$1 \div 1 = 1$$

(2) 有符号数的表示

计算机无法识别 (+) / (-) \Rightarrow 将 + / - 也用 0 / 1 表示

机器数 / 机器码：将数的符号和数值部分均用 0 / 1 进行
编码所表示出来的二进制数

真值：用 + / - 表示出来以十进制数或二进制数。

机器数 { 无符号数：所有二进制位均用来表示数值
有符号数：数的符号和编码均用二进制编码表示

正 0

负 1

{ 原码：最高位符号，其余相同
反码：最高位符号，正数与原码相同，
负数按位取反。

补码：“模”指一个系统的量程

e.g. 时钟模为 12，4 位二进制数模为 16

补码 \rightarrow 原码

8 位二进制数模为 256

减去某数可用加上其补码代替

最高位符号，正数与原码相同，
负数按位取反后，在最低位 + 1

(3) 补码的加减运算

$$[X]_{\text{补}} + [Y]_{\text{补}} = [X + Y]_{\text{补}}$$

$$[X]_{\text{补}} - [Y]_{\text{补}} = [X]_{\text{补}} + [-Y]_{\text{补}} = [X - Y]_{\text{补}}$$

注意：

① 参与运算的操作均为补码

② 符号位和数值位按同样的规则参加运算，结果的符号位由运算得出。

③ 补码总是对确定的模而言，如果运算结果超过了模，则去掉模（即进位）才能得到正确结果。

“溢出” \Rightarrow 位扩展

△ 九位有符号的二进制数的原码、反码和补码的数值范围

原码： $-(2^{n-1}-1) \sim +(2^{n-1}-1)$

反码： $-(2^{n-1}-1) \sim +(2^{n-1}-1)$

补码： $-2^{n-1} \sim +(2^{n-1}-1)$

4 二进制代码

码制：编制代码所要遵循的规则

二进制代码：将若干个二进制数码(0和1)按一定规律排列起来表示某种特定的信息，称为二进制代码

编码：用n位二进制数表示2^n个不同的信息，给每个信息规定一个具体的二进制数编码，这个过程为编码。

(1) 二-十进制码 (Binary-Coded Decimal)

用二进制编码表示的十进制数，简称BCD码

$2^3 < 10 < 2^4 \Rightarrow$ 需要4位二进制数

16种组合 \Rightarrow 选择10种表示 \Rightarrow 每种方案表示一种BCD码

e.g. 8421 BCD 码 (最常用)

十进制数 \Leftrightarrow BCD 码

$$(97)_D \Leftrightarrow (1001\ 0111)_{BCD}$$

注意：“十进制与二进制转换”与“用BCD码表示十进制数”概念不同！！

无权码

0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

几种常用的BCD代码

十进制 数码	“权” 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

(2) 格雷码

无权码、循环码、反射码

特点：① 两个相邻代码（包括首和尾）之间仅有一位取值不同。

② 最高位的0和1只改变一次；若以最高位的0和1交界为轴，其他位的代码是上下对称的。

用途：主要用于角度编码
（不能直接进行算术运算）

二进制码转换为格雷码

① 格雷码的最高位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二位相加（舍去进位）。

作为格雷码的下一位

e.g.	二进制码	1 0 1 1 0 1
	格雷码	1 1 1 0 1 1

(3) ASCII 码和奇偶校验码

ASCII 码：七位二进制编码 共 128 个

奇偶校验码：有效信息 (k 位) + 检验位 (1 位)

{ 奇检验码：信息位和检验位中“1”个数之和为奇数
偶检验码：信息位和检验位中“1”个数之和为偶数



由编码器根据信息位
编码产生奇偶检验位

通过检测器检查
含“1”个数的奇偶

注意 ① 只有检错能力，无纠错能力
② 只能发现单个错误，不能发现双错

§2 逻辑代数

1 逻辑代数简介

George Boole



Claude E. Shannon

逻辑代数 / 布尔代数

按一定逻辑规律进行运算的代数

布尔代数是分析继电器的有效方法

“二值逻辑”

在数字电路中

条件 → 输入信号
结果 → 输出信号

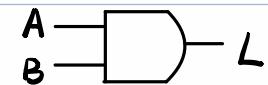
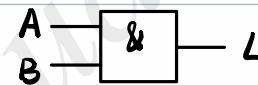
条件和结果的状态用逻辑“1”和“0”表示

2 逻辑运算和集成逻辑门简介

(1) 基本逻辑运算

与

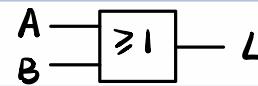
与门电路



$$L = A \cdot B$$

或

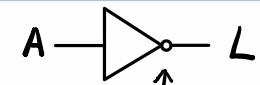
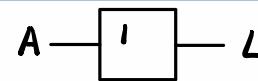
或门电路



$$L = A + B$$

非

非门电路



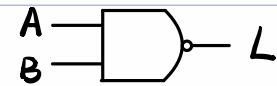
$$L = \bar{A}$$

表示反相

(2) 复合逻辑运算

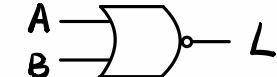
与非逻辑运算

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



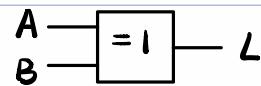
或非逻辑运算

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



异或逻辑运算

A, B 相同时, 输出 $L = 0$



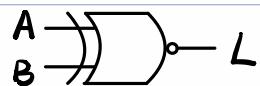
A, B 不同时, 输出 $L = 1$



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

同或逻辑运算

A、B 相同时，输出 $L = 1$



A、B 不同时，输出 $L = 0$

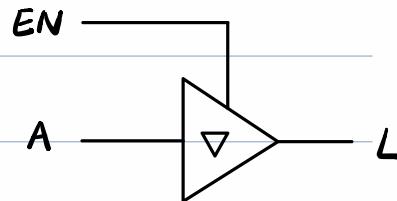
$$L = A \oplus B = AB + \bar{A}\bar{B}$$

(3) 三态门

有三种可能的输出：0、1、Z

Z 指输出为高阻态

Z 意味着输入与输出之间是断开的
真值表



使能 EN	输入 A	输出 L
1	0	0
1	1	1
0	X	Z

(4) 集成电路简介

集成电路 { 双极型
单极型
混合型

传输延迟：输出滞后于输入

高 \rightarrow 低 延迟 t_{PHL}

低 \rightarrow 高 延迟 t_{PLH}

平均传输延迟时间 $t_{pd} = \frac{1}{2}(t_{PHL} + t_{PLH})$

3 逻辑代数的基本定理和规则

(1) 逻辑代数基本定律

0、1 律

$$A \cdot 0 = 0$$

$$A + 0 = A$$

$$A \cdot 1 = A$$

$$A + 1 = 1$$

重叠律

$$A \cdot A = A$$

$$A + A = A$$

互补律

$$A \cdot \bar{A} = 0$$

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

还原律

$$\bar{\bar{A}} = A$$

交换律

$$A \cdot B = B \cdot A$$

$$A + B = B + A$$

结合律

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

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

可以以任何方式对变量分组

分配律

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

$$A + BC = (A + B)(A + C)$$

反演律 / 摩根定理

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

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

可用完全归纳法证明

(2) 逻辑代数常用公式

吸收律

$$A + A \cdot B = A$$

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

补吸收律

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

重要公式 $AB + \bar{A}C + BC = AB + \bar{A}C$

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

(3) 逻辑代数基本规则

代入规则

在任何一个包含 A 的式子，用另一个逻辑式代替 A ，式子仍然成立。

反演规则

对于任一逻辑表达式 L ，将其中与 (\cdot) 和或 ($+$) 交换、原变量和反变量交换、1 和 0 交换，得到的表达式是原函数的反函数。

注意：①保持原有优先级

② 对反变量以外的非号保持不变

e.g. $L = \bar{A} \cdot \bar{B} + C \cdot D$

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

对偶规则 “或”、“与”互换，0、1互换

得到 L 的对偶式 L'

e.g. $L = (A + \bar{B})(A + C)$

$$L' = A \cdot \bar{B} + A \cdot C$$

4 逻辑函数及其表示方法

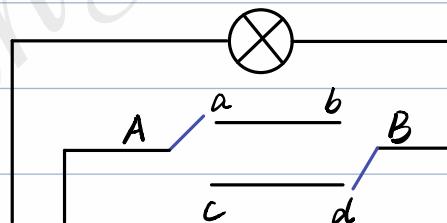
输入(条件) $\xrightarrow{\text{逻辑函数 (Logic Function)}}$ 输出(结果)

example.

(1) 逻辑真值表

开关 A、B 上(1) 下(0)

灯 L 灭(0) 亮(1)



A	B	L
0	0	1
0	1	0
1	0	0
1	1	1

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

$$AB$$

(2) 逻辑函数表达式

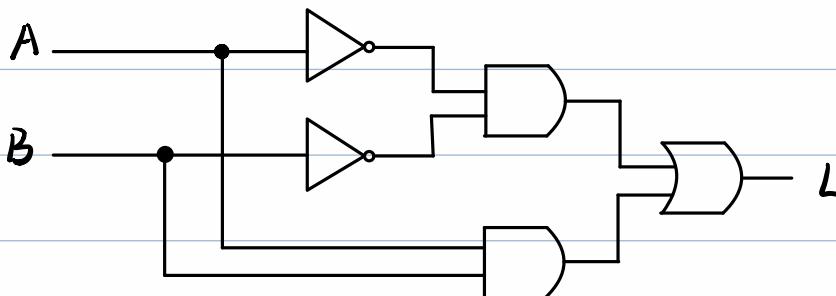
$$L = \bar{A} \cdot \bar{B} + A \cdot B$$

步骤：

- ① 把每个输出为1的一组输入变量组合写成乘积项
- ② 乘积项中，逻辑值1用原变量，0用反变量表示
- ③ 将乘积项进行逻辑或

注意：① 输入变量向量是“与”
输出状态的组合是“或”
② 对 A、B、L，取值 1 用原变量，0 用反变量表示

(3) 逻辑图



(4) 波形图

(5) 卡诺图

(6) 硬件描述语言 (HDL)

5 逻辑函数的代数化简法

(1) 逻辑函数表达式的形式

与 - 或表达式 积之和

或 - 与表达式 和之积

(2) 逻辑函数的代数化简法

如何判断“与 - 或”最高？

① 包含“与”项的个数最少

② 每个“与”项中的变量数最少

化简的主要方法

① 公式法 (代数法)

{ 并项法 $A + \bar{A} = 1$

吸收法 $A + AB = A$

消去法 $A + \bar{A}B = A + B$

配项法 $A = A(B + \bar{B})$

② 图解法 (卡诺图法)

(之后介绍)

(3) 逻辑函数表达式的变换

一种集成电路芯片内部通常只有一种类型的逻辑门。
为了减少门的种类，适应芯片的情况，需要
变换逻辑表达式

e.g. 与非

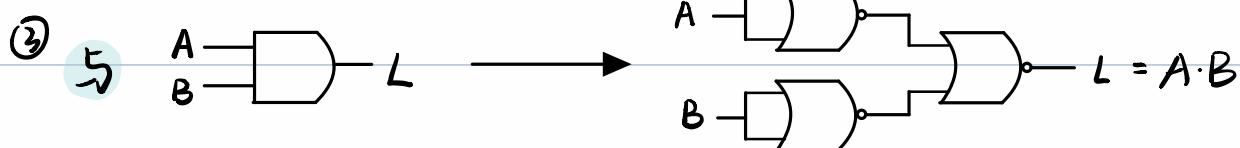
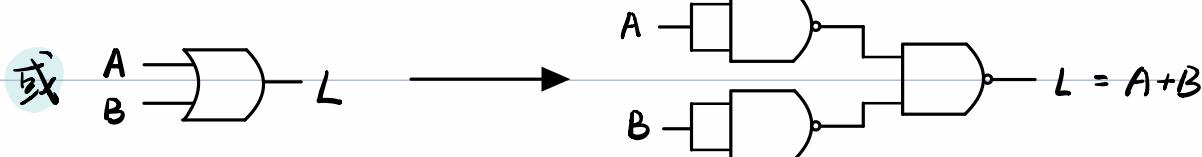
$$\begin{aligned} L &= A(B+C) = AB + AC \\ &= \overline{\overline{AB} + \overline{AC}} \\ &= \overline{\overline{AB} \cdot \overline{AC}} \end{aligned}$$

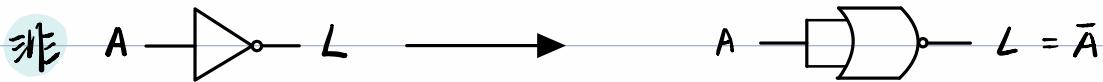
① 取非2次
② 幂根整理

或非

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

Note ① 所有的逻辑电路都可以只用与非门实现，
也可以只用或非门实现。





6 逻辑函数的卡诺图化简法

逻辑函数表达式 { 与-或表达式 → 最小项表达式
的基本形式 或-与表达式 → 最大项表达式 }

(1) 逻辑函数的最小项表达式

最小项：在 n 变量逻辑函数中，若一个乘积项包含全部的 n 个变量，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一次。

n 个变量的最小项有 2^n 个

最小项的编号

e.g. $\bar{A}\bar{B}\bar{C}$ = 二进制 000 → 十进制 0 记作 m_0
 $\bar{A}\bar{B}C$ = 二进制 001 → 十进制 1 记作 m_1

最小项的性质：

① \forall 最小项，输入变量只有一组取值使其值为 1，其它各组取值均使它为 0。

② \forall 取值，任意两个最小项乘积为 0

③ \forall 一组取值，全体最小项之和为 1

最小项表达式 / 标准与-或式

每个“与”项都是最小项

e.g. $L(A, B, C) = AB + \bar{A}C$

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

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

$$= m_7 + m_6 + m_3 + m_1$$

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

逻辑函数变换为最小项表达式

① 变换为乘积项之和

② 用 $A + \bar{A} = 1$ 配项

③ 逻辑函数的 **最大项表达式** (与最小项表达式类似)

最大项: 每一个或项都包含全部n个变量

最大项的编号 M_i :

对于一个最大项，输入变量只有一组二进制数使其取值为0，与该组二进制数对应的十进制数是该最大项的下标编号。

e.g. $\bar{A} + \bar{B} + C$

$$A=1 \quad B=1 \quad C=0 \rightarrow \bar{A} + \bar{B} + C = 0$$

$$110 \rightarrow 6 \rightarrow M_6$$

最大项的性质

① \vee 最大项，只有一组变量的取值使其值为0

② \vee 二最大项之和为1

③ \vee 一组变量取值，全体最大项之积为0

最大项表达式 / 标准或一与式

△ **最小项与最大项的关系** $M_i = \bar{m}_i$

e.g. 某电路真值表如下，试写出最小项和最大项表达式

① **最小项表达式**: 将L=1各最小项相加

$$L(A, B, C) = m_3 + m_5 + m_6$$

$$= \sum m(3, 5, 6)$$

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

A	B	C	L
0	0	0	0 m_0
0	0	1	0 m_1
0	1	0	0 m_2
0	1	1	1 $\rightarrow m_3$
1	0	0	0 m_4
1	0	1	1 $\rightarrow m_5$
1	1	0	1 $\rightarrow m_6$
1	1	1	0 m_7

② **最大项表达式**: 将L=0各最大项相乘

$$L(A, B, C) = M_0 \cdot M_1 \cdot M_2 \cdot M_4 \cdot M_7$$

$$= \prod M(0, 1, 2, 4, 7)$$

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

(3) 卡诺图的引入

① 相邻最小项：只有一个变量互为反变量

② 填写规则：逻辑相邻和几何位置相邻一致

③ “折叠展开”

i) 一变量卡诺图

0	1
---	---

ii) 两变量卡诺图

0	1	3	2
---	---	---	---

iii) 三变量卡诺图

0	1	3	2
4	5	7	6

iv) 四变量卡诺图

AB	CD	00	01	11	10
00					
01					
11					
10					

0	1	3	2
4	5	7	6
12	13	15	14
8	9	11	10

(4) 逻辑函数的卡诺图表示法

方法 ① 逻辑函数 \rightarrow 最小项表达式

② 填写卡诺图

③ L中列出的最小项，填1

L中不存在的最小项，填0

(4) 逻辑函数的卡诺图化简法

化简依据 $\left\{ \begin{array}{l} \text{相邻有一项互为相反项} \\ A + \bar{A} = 1 \end{array} \right.$

化简步骤 ① 逻辑函数 \rightarrow 最小项之和

② 填写卡诺图

- ③ 画包围圈，每个包围圈含 2^n 个方格
 ④ 写出每个包围圈的乘积项，
 将所有包围圈的乘积项相加。

包围圈的一般规则

- ① 圈内方格数 2^n 个， $n=0, 1, 2, \dots$
- ② 相邻方格包括上下底、左右边、四个角
- ③ 一个方格可被重复包围
- ④ 圈内方格尽可能多，包围圈数目尽可能少。

(5) 具有无关项的逻辑函数化简

无关项

无关项的处理 化简时可以取0或1，使函数尽可能简化。用“d”、“x”表示。

单输出逻辑函数的化简

- 方法 ① 列真值表
 ② 画卡诺图
 ③ 画包围圈

$$e.g. L(A, B, C, D) = \sum m(5, 6, 7, 8, 9) + \sum d(10, 11, 12, 13, 14, 15)$$

		CD	00	01	11	10
		AB	00	01	11	10
00	01	0	0	0	0	0
		0	1	1	1	1
11	10	x	x	x	x	
		1	1	x	x	

2^n

1, 2, 4, 8

$$L(A, B, C, D) = A + BC + BD$$

多输出逻辑函数的化简

tips: 找共同部分, 使整体最简 (成本降低)

e.g. $L_1(A, B, C) = \sum m(0, 4, 5, 6, 7)$

$L_2(A, B, C) = \sum m(0, 6, 7)$

(L₁)

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

考虑共享乘积项 $\bar{A}\bar{B}\bar{C}$

$$L_1(A, B, C) = A + \bar{A}\bar{B}\bar{C}$$

$$L_2(A, B, C) = AB + \bar{A}\bar{B}\bar{C}$$

(L₂)

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

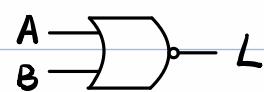
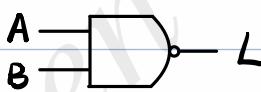
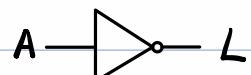
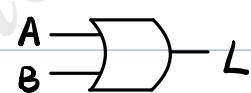
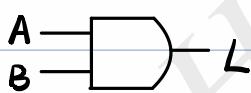
不考虑共享乘积项 $\bar{A}\bar{B}\bar{C}$

$$L_1(A, B, C) = A + \bar{B}\bar{C}$$

$$L_2(A, B, C) = AB + \bar{A}\bar{B}\bar{C}$$

7 逻辑门的替代符号

标准符号:

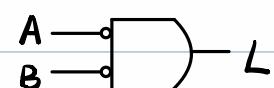
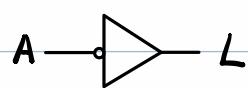
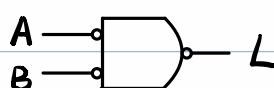
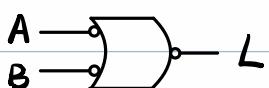


如何将标准符号变为替代符号

① 把标准符号的输出和每一个输入反相

② 运算符号“与” \leftrightarrow “或”

替代符号



注意 ① 等效逻辑符号可以推广到具有更多输入端的门电路。

② 标准符号的输入端无小圆圈，

所有替代符号的输入端都有小圆圈。

③ 每一种门的标准符号和替代符号都代表相同
的实际电路。

有效逻辑电平

输入或输出线上无小圆圈 → 高电平有效

输入或输出线上有小圆圈 → 低电平有效

§3. 组合逻辑电路

逻辑电路 { 组合逻辑电路
时序逻辑电路

1. 组合逻辑电路的分析

(1) 组合逻辑电路的特点

结构特征

① 输入、输出间无反馈延迟通路

② 电路中不含记忆单元

工作特征

电路的输出状态只取决于同一时刻的输入状态

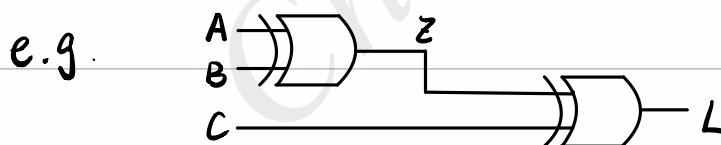
(2) 分析方法

① 由逻辑图写出各输出端的逻辑表达式

② 化简和变换

③ 列出真值表

④ 分析确定电路功能 (文字表述)



① 逻辑表达式

$$L = Z \oplus C = A \oplus B \oplus C$$

② 写真值表

③ 逻辑功能

奇数个 1, $L = 1$

偶数个 1, $L = 0$

“奇校验电路”

A	B	C	$Z = A \oplus B$	$L = (A \oplus B \oplus C)$
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	0	1

2 组合逻辑电路的设计

1) 设计过程

- ① 逻辑抽象
- ② 列真值表
- ③ 写逻辑表达式
- ④ 简化
- ⑤ 画逻辑图

2) 优化实现 用指定芯片实现逻辑电路

单输出电路

相同输入端的与非门比与门或者或门所用晶体管少

多输出电路

考虑“共享乘积项”

多级逻辑电路

- ① 提取公因子

- ② 函数分解

3 组合逻辑电路中的竞争冒险

1) 产生原因

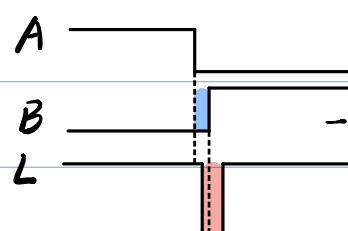
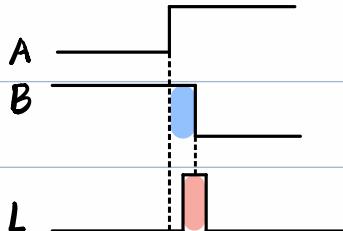
信号通过逻辑门有延迟时间



不考虑门的延迟时间，且 $B = \bar{A}$

$$L = AB = 0 \quad L = A + B = 1$$

考虑门的延迟时间，且用非门实现 $B = \bar{A}$



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相反对，由竞争而可能产生输出干扰脉冲的现象。

(2) 消除方法

① 发现并消除互补项

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

当 $B=C=0$ 时, $L=A\bar{A}$ 可能出现竞争冒险

变换 L 为 $L = AC + \bar{A}B + BC$

② 增加乘积项避免互补项相加

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

当 $A=B=1$ 时, $L=C+\bar{C}$ 可能出现竞争冒险

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

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

③ 输出端并联电容

4 编码器

编码: 赋予二进制码含义的过程

编码器 { **普通编码器**: 任何时刻只允许输入一个有效的编码信号

优先编码器: 允许多个输入信号同时有效，仅对优先级高的输入进行编码。

4线 —— 2线普通编码器

输入

I_0

I_1

I_2

I_3

输出

J_1

J_0

1	0	0	0	0	0	0
0	1	0	0	0	0	1
0	0	1	0	1	1	0
0	0	0	1	1	1	1

4线 —— 2线优先编码器

输入				输出	
I ₀	I ₁	I ₂	I ₃	Y ₁	Y ₀
1	0	0	0	0	0
x	1	0	0	0	1
x	x	1	0	1	0
x	x	x	1	1	1

典型编码器电路

8线 —— 3线优先编码器 CD4532

输入信号高电平有效，输出为二进制代码

CD4532	I ₀	EO	EI	GS	EO	状态
	I ₁	GS	0	0	0	不工作
	I ₂		1	0	1	工作，但无输入
	I ₃			1	0	工作且有输入
	I ₄				1	不可能出现
	I ₅					
	I ₆					
	I ₇					

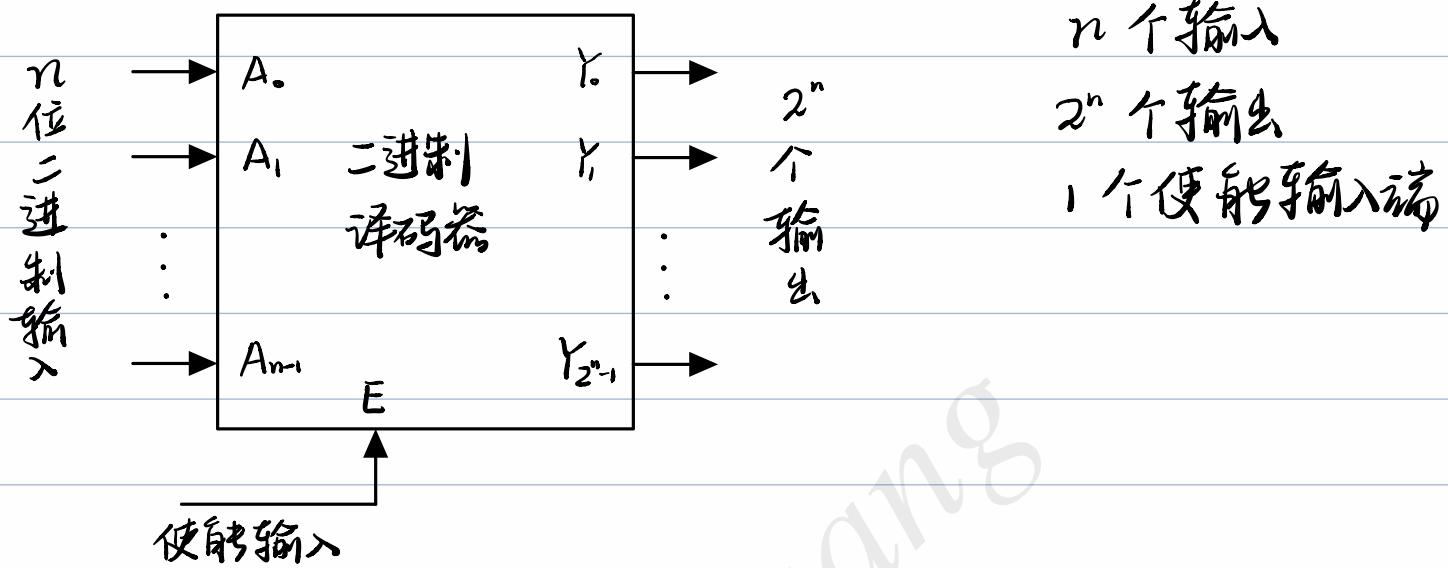
输入										输出				
EI	I ₇	I ₆	I ₅	I ₄	I ₃	I ₂	I ₁	I ₀	Y ₂	Y ₁	Y ₀	GS	EO	
0	x	x	x	x	x	x	x	x	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	1	x	x	x	x	x	x	x	1	1	1	1	0	
1	0	1	x	x	x	x	x	x	1	1	0	1	0	
1	0	0	1	x	x	x	x	x	1	0	1	1	0	
1	0	0	0	1	x	x	x	x	1	0	0	1	0	
1	0	0	0	0	1	x	x	x	0	1	1	1	0	
1	0	0	0	0	0	1	x	x	0	1	0	1	0	
1	0	0	0	0	0	0	1	x	0	0	1	1	0	
1	0	0	0	0	0	0	0	1	0	0	0	1	0	

5 译码器

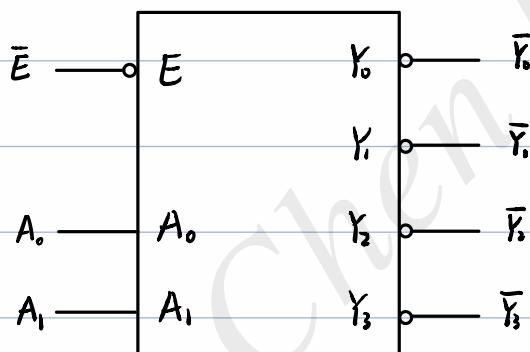
二进制代码 译码 → 高低电平信号 / 其他代码

(1) 二进制译码器

“多输入” “多输出”



2线 - 4线译码器



$$\begin{aligned}\bar{Y}_0 &= \overline{E \cdot \bar{A}_1 \bar{A}_0} = \overline{\bar{E} \cdot m_0} \\ \bar{Y}_1 &= \overline{E \cdot \bar{A}_1 A_0} = \overline{\bar{E} \cdot m_1} \\ \bar{Y}_2 &= \overline{E \cdot A_1 \bar{A}_0} = \overline{E \cdot m_2} \\ \bar{Y}_3 &= \overline{E \cdot A_1 A_0} = \overline{E \cdot m_3}\end{aligned}$$

输入

\bar{E} A_1 A_0

输出

\bar{Y}_0 \bar{Y}_1 \bar{Y}_2 \bar{Y}_3

1	X	X	1	1	1	1
---	---	---	---	---	---	---

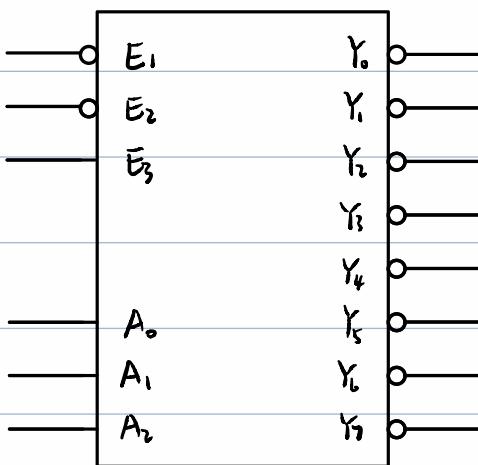
0	0	0	0	1	1	1
---	---	---	---	---	---	---

0	0	1	1	0	1	1
---	---	---	---	---	---	---

0	1	0	1	1	0	1
---	---	---	---	---	---	---

0	1	1	1	1	1	0
---	---	---	---	---	---	---

3线—8线译码器



$$\begin{aligned}Y_0 &= \overline{E} \overline{A}_2 \overline{A}_1 \overline{A}_0 \\Y_1 &= \overline{E} \overline{A}_2 \overline{A}_1 A_0 \\Y_2 &= \overline{E} \overline{A}_2 A_1 \overline{A}_0 \\Y_3 &= \overline{E} A_2 \overline{A}_1 A_0\end{aligned}$$

...

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	1	0	1	1	1
1	0	0	1	0	0	1	1	1	1	1	0	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

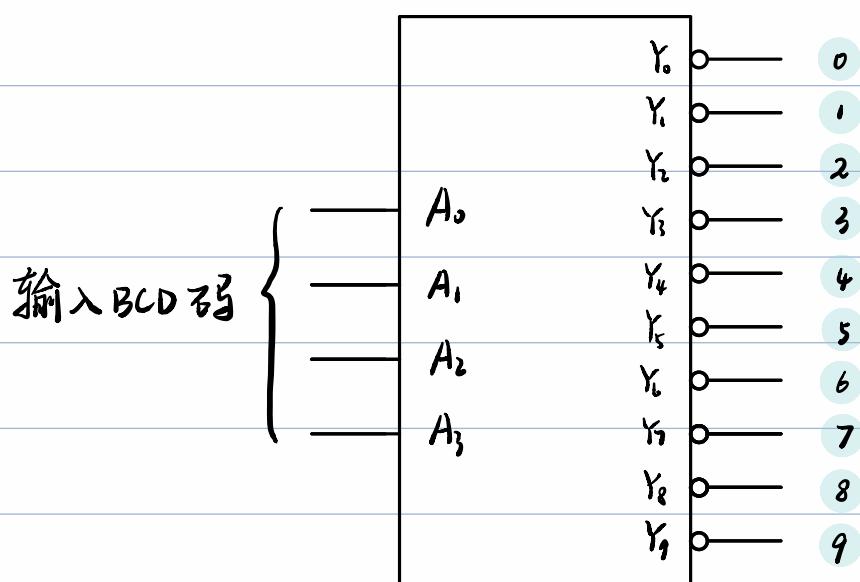
(2) 二—十进制译码器

功能：将 8421 BCD 码译成 10 个状态输出

⇒ 4 个输入端，10 个输出端，输出低电平有效

当输出超过 8421 码的范围时 (1010~1111)

输出均为高电平，此时没有有效译码输出。



BCD 输入				输出									
A ₃	A ₂	A ₁	A ₀	Y ₀	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇	Y ₈	Y ₉
0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1	1	1
0	1	1	0	1	1	1	1	1	1	0	1	1	1
1	0	1	1	1	1	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0

e.g. 若二十一进制译码器输出为 111101111
可以表示为 16进制 (3EF)_H

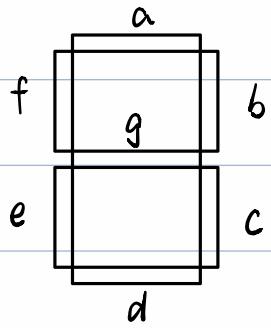
(3) 七段显示译码器

数字显示电路 { 译码驱动器

{ 数码显示器 { 七段数字显示器

液晶显示器

七段数码管



发光二极管的连接

{ 花阴极显示器
花阳极显示器

显示译码器 / 七段译码器

功能：将输入的 4 位 BCD 码翻译成“七段码”
送给数码管。

分类 { 输出高电平有效：用来驱动花阴极显示器
输出低电平有效：用来驱动花阳极显示器

e.g. 74HC4511 输出高电平有效

(4) 数字显示电路

显示电路

{ 静态显示电路：一个译码器与一个显示器相连接

{ 动态扫描显示电路：一个译码器与多个显示器相
↓ (视觉暂留) 连接，再增加显示控制电路。

注意 ① f_c 太高 \Rightarrow 看不清显示的数码

f_c 太低 \Rightarrow 看到 4 个数字轮流显示

② 通常 $25 \text{ Hz} < f_c < 100 \text{ Hz}$

6 数据分配器与数据选择器

(1) 数据分配器 (相当于单刀多掷开关)

数据分配 将公共数据线上的数据根据需要送到不同的通道上去。

实现 ① 3线-8线译码器 74H138

改变 $A_2 A_1 A_0$ 的值

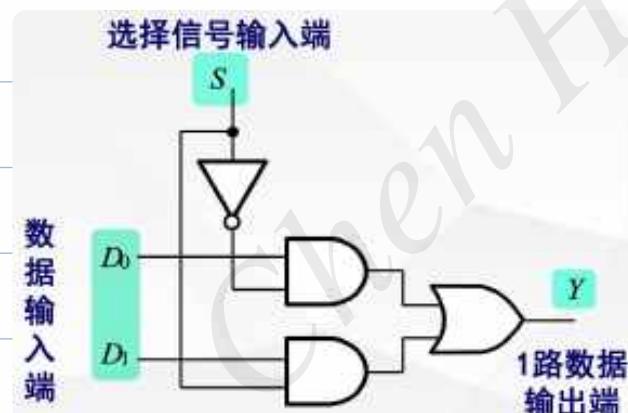
将数据分配到不同输出端

② 三态门

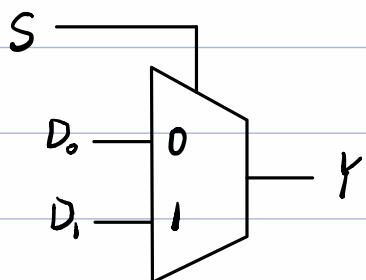
(2) 数据选择器

从多个数据源中选择一路送到公共线路上

① 2选1数据选择器



逻辑符号



逻辑表达式 $Y = \bar{S}D_0 + SD_1$

② 4选1数据选择器

$$Y = D_0\bar{S}_1\bar{S}_0 + D_1\bar{S}_1S_0 + D_2S_1\bar{S}_0 + D_3S_1S_0$$

③ 8选1数据选择器

2个4选1 + 1个2选1

数据选择器的应用

△ 变量数 = 选通端数，

用数据选择器实现函数的步骤：

① 将函数变换为最小项表达式

② 让数据选择器处于使能状态，并将函数的输入变量与地址信号 S_2, S_1, S_0 相连接

③ 将数据输入 $D_0 \sim D_3$ 接相应逻辑电平，即在逻辑表达式中有 m_i ，则相应 $D_i = 1$ ，其余为 0。

7 数值比较器与加法运算电路

(1) 数值比较器

一位数值比较器

真值表	A	B	$F_{A>B}$	$F_{A < B}$	$F_{A=B}$
	0	0	0	0	1
	0	1	0	1	0
	1	0	1	0	0
	1	1	0	0	1

$$\text{逻辑表达式 } F_{A>B} = A\bar{B}$$

$$F_{A < B} = \bar{A}B$$

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

二位數值比較器

A_1, B_1	A_0, B_0	$F_{A>B}$	$F_{A<B}$	$F_{A=B}$
$A_1 > B_1$	x	1	0	0
$A_1 < B_1$	x	0	1	0
$A_1 = B_1$	$A_0 > B_0$	1	0	0
$A_1 = B_1$	$A_0 < B_0$	0	1	0
$A_1 = B_1$	$A_0 = B_0$	0	0	1

$$F_{A>B} = A_1 \bar{B}_1 + (A_1 B_1 + \bar{A}_1 \bar{B}_1) A_0 \bar{B}_0$$

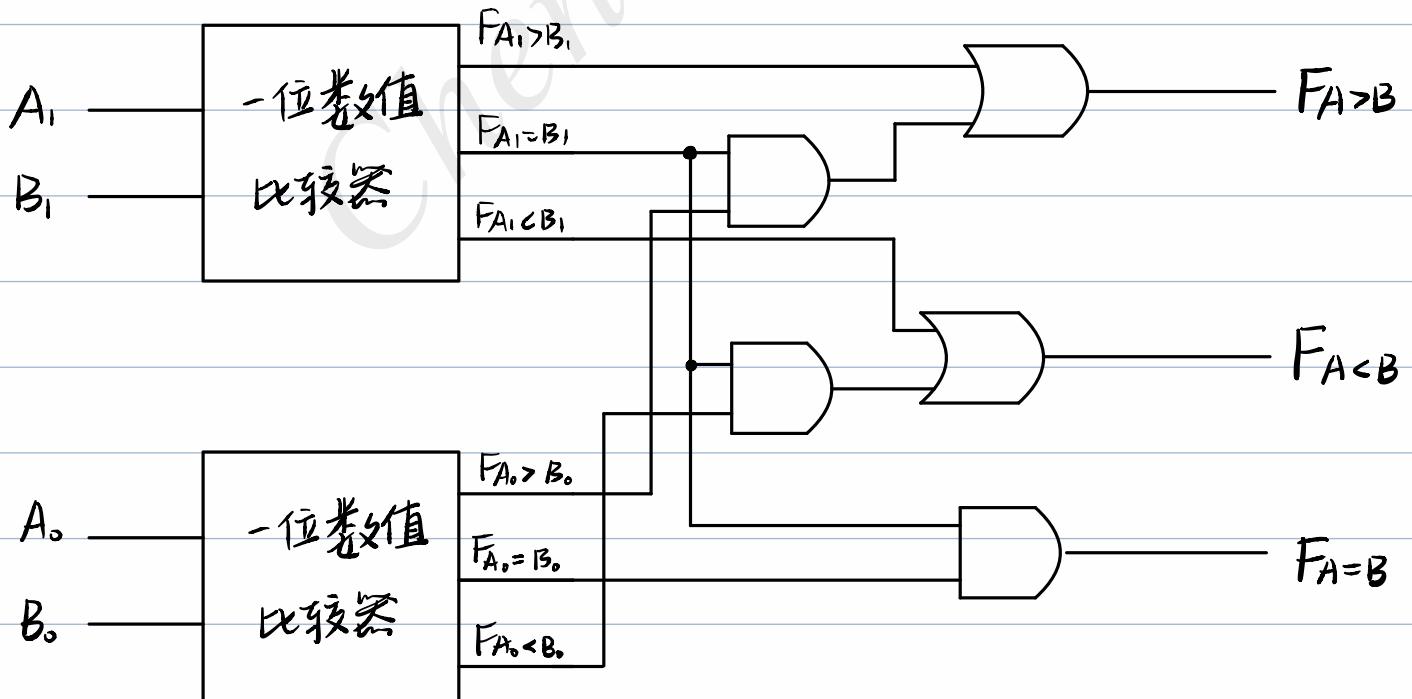
$$= F_{A_1 > B_1} + F_{A_1 = B_1} \cdot F_{A_0 > B_0}$$

$$F_{A<B} = \bar{A}_1 B_1 + (A_1 B_1 + \bar{A}_1 \bar{B}_1) \bar{A}_0 B_0$$

$$= F_{A_1 < B_1} + F_{A_1 = B_1} \cdot F_{A_0 < B_0}$$

$$F_{A=B} = (A_1 B_1 + \bar{A}_1 \bar{B}_1)(A_0 B_0 + \bar{A}_0 \bar{B}_0)$$

$$= F_{A_1 = B_1} \cdot F_{A_0 = B_0}$$



(2) 典型数值比较器

四位数值比较器 74HC85

数值比较器的位数扩展

{ 串联
 | 并联 (快)

(3) 加法器

{ 半加器 两个1位二进制数相加时，不考虑低位的进位。
 | 全加器 考虑 ...

半加器

A	B	进位 C	和 S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

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

$$C = AB$$

全加器

上一级的进位

A	B	<u>C_i</u>	S	C _o
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

$$S = \bar{A}\bar{B}C_1 + \bar{A}B\bar{C}_1 + A\bar{B}\bar{C}_1 + ABC_1 \\ = A \oplus B \oplus C$$

$$C_0 = AB + A\bar{B}C_1 + \bar{A}BC_1 \\ = AB + (A \oplus B)C_1$$

(4) 多位数加法器

串行进位加法器：逻辑电路简单

级前进位加法器：运算速度快

$$\begin{array}{r} \text{e.g. } \quad A_3 \quad A_2 \quad A_1 \quad A_0 \\ + \quad B_3 \quad B_2 \quad B_1 \quad B_0 \\ \hline C_3 \quad S_3 \quad S_2 \quad S_1 \quad S_0 \end{array}$$

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

$$C_i = A_i B_i + (A_i \oplus B_i) C_{i-1}$$

$$\text{设 } G_i = A_i B_i \quad P_i = A_i \oplus B_i$$

$$S_i = P_i \oplus C_{i-1}$$

$$C_i = G_i + P_i C_{i-1}$$

当 $A_i = B_i = 1$ 时， $G_i = 1$ ， $P_i = 0 \Rightarrow C_i = 1$

G_i 产生变量

当 $A_i = 0$, $B_i = 1$ 或 $A_i = 1$, $B_i = 0$ 时, $P_i = 1$, $G_i = 0$

$\Rightarrow C_i = C_{i-1}$ P_i 传输变量

8 组合可编程逻辑器件

PLD : Programmable Logic Device

(1) 性能特点

① 逻辑功能强

② 集成度高

③ 设计方法灵活

④ 具有完善先进的开发工具

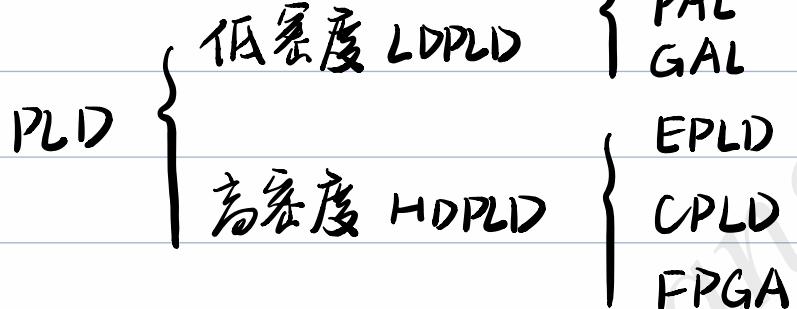
⑤ 系统处理速度高

⑥ 系统具有加密功能

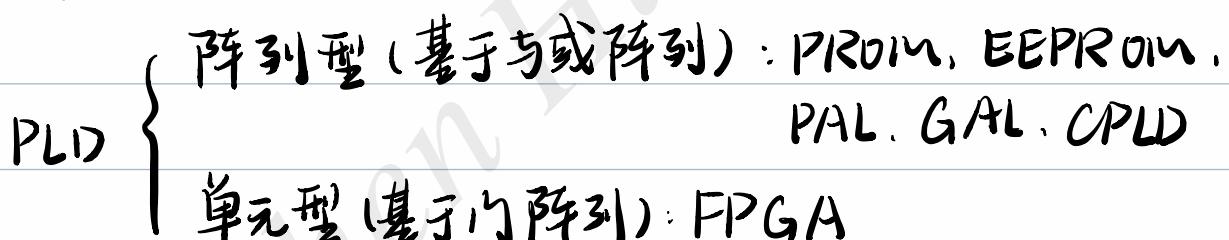
⑦ 使用方便

(iv) 分类

① 按密度分



② 按结构特点分

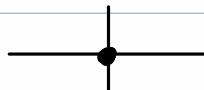


(3) PLD 的逻辑符号表示方法

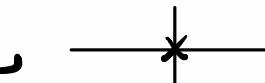
门阵列的每个交叉点都是一个“单元”

连接方式

硬线连接方式



可编程接通单元



被编程擦除单元



(4) 组合逻辑电路的PLD实现

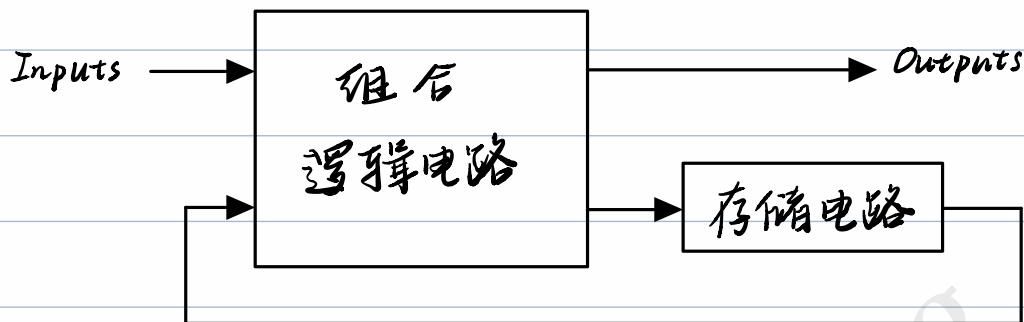
§4 锁存器和触发器

1 概述

时序逻辑电路的基本结构

存储电路：保存电路当前的状态

组合逻辑电路：逻辑运算



基本双稳态电路

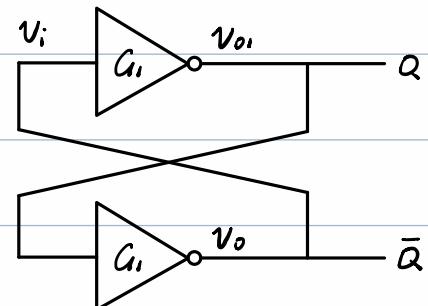
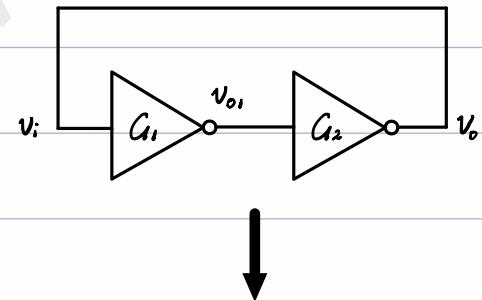
电路一旦进入某一种逻辑状态，

就能长期保持该状态不变。

Q 常态输出

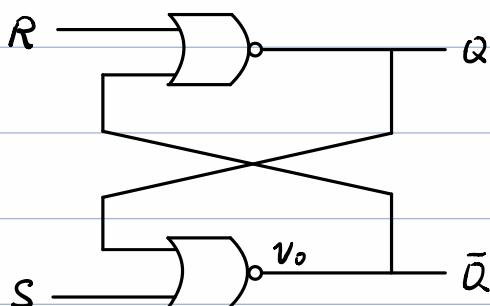
\bar{Q} 反态输出

定义
 $\begin{cases} 0 \text{ 状态} & Q = 0 \quad \bar{Q} = 1 \\ 1 \text{ 状态} & Q = 1 \quad \bar{Q} = 0 \end{cases}$



2 SR锁存器

(1) 用或非门组成的SR锁存器 / 基本SR锁存器

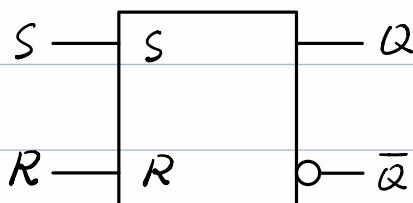


功能表

S	R	Q	\bar{Q}	功能
0	0	不变	不变	保持
0	1	0	1	复位 / 置 0
1	0	1	0	置 1
1	1	0	0	非定义状态 →

逻辑符号

约束条件 $S \cdot R = 0$

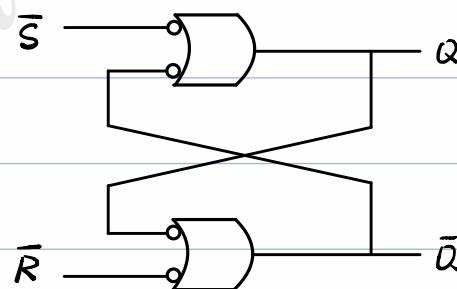


动态特性

传输延迟时间 t_{PLH} 和 t_{PHL}

脉冲宽度 t_w ：基本 SR 锁存器工作时，必须保证 S 和 R 的高电平脉冲宽度不小于某一最小值 t_w

(2) 用与非门构成的 SR 锁存器 / 基本 SR 锁存器

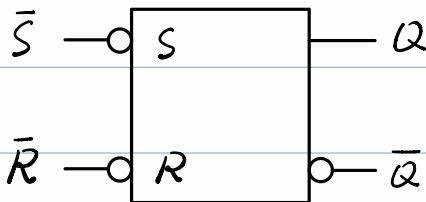


功能表

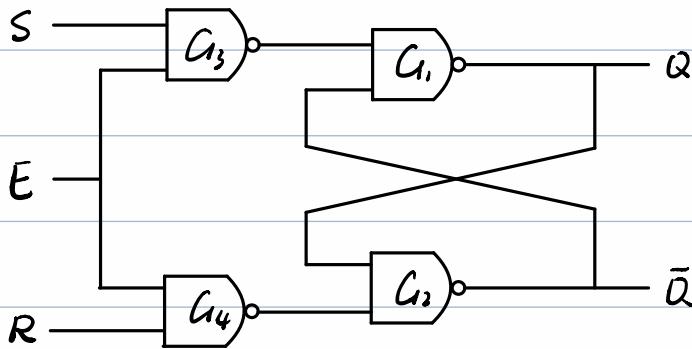
\bar{S}	\bar{R}	Q	\bar{Q}	功能
1	1	不变	不变	电路输出不变
1	0	0	1	置 0 / 复位 (reset)
0	1	1	0	置 1 / 置位 (set)
0	0	1	1	非定义状态 →

约束条件 $\bar{S} + \bar{R} = 1$

逻辑符号



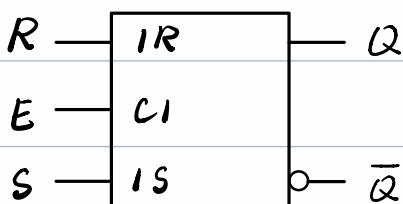
(3) 门控SR锁存器



功能表

E	S	R	Q	\bar{Q}	功能
0	X	X	不变	不变	保持
1	0	0	不变	不变	保持
1	0	1	0	1	Reset
1	1	0	1	0	Set
1	1	1	1	1	非定义状态

逻辑符号



CI 和 IR, IS 表示内部逻辑之间的关系

$C \Rightarrow \text{控制} \quad "1" \xrightarrow{\text{控制}} "1" \quad "$

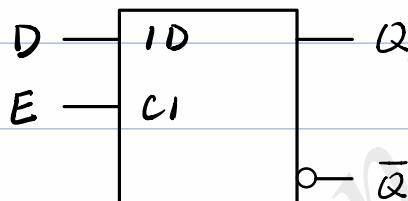
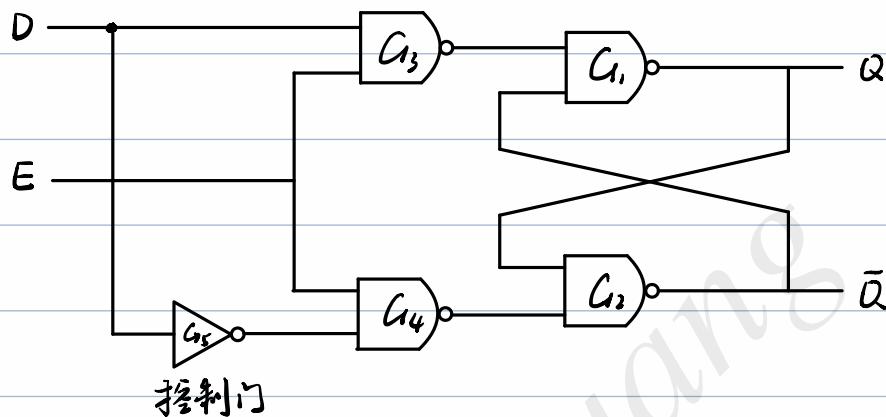
3 D锁存器

*不存在非定义状态

D锁存器 { 逻辑门控D锁存器
传输门控D锁存器

(1) D锁存器的电路结构

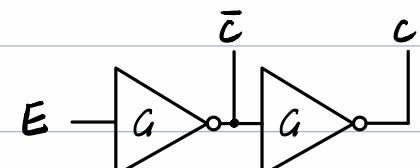
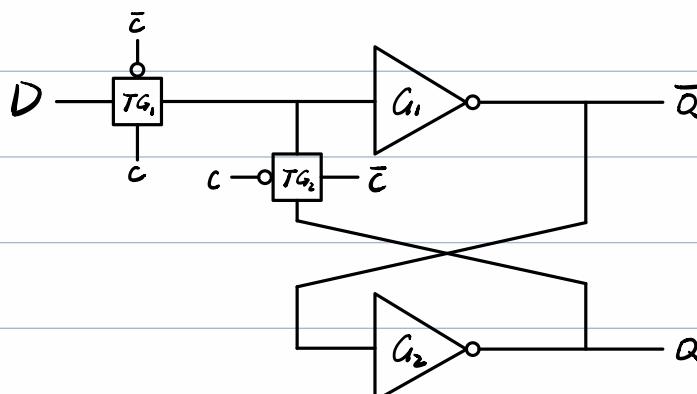
逻辑门控D锁存器



功能表

E	D	Q	\bar{Q}	功能
0	x	不变	不变	保持
1	0	0	1	Reset
1	1	1	0	Set

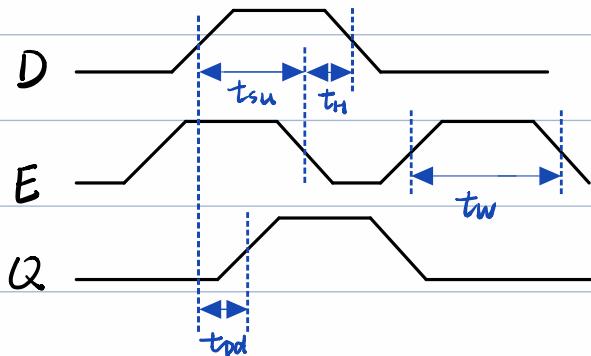
传输门控D锁存器



动能表

E	D	Q	\bar{Q}	功能
0	X	不变	不变	保持
1	0	0	1	置0
1	1	1	0	置1

(2) D锁存器的动态特性



传输延迟时间 t_{pd} ($D \rightarrow Q$)：输出信号对输入信号的响应延迟时间。

建立时间 t_{su} ：D信号对下降沿的最小提前量

保持时间 t_h ：D信号电平在E电平下降后需要继续保持的最长时间

脉冲宽度 t_w ：E信号的宽度不小于 t_w

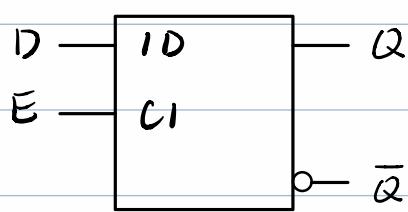
4 逐从D触发器

(1) 逐从D触发器的结构及工作原理

锁存器 v.s. 触发器：触发方式不同

锁：电平敏感

触：边沿敏感

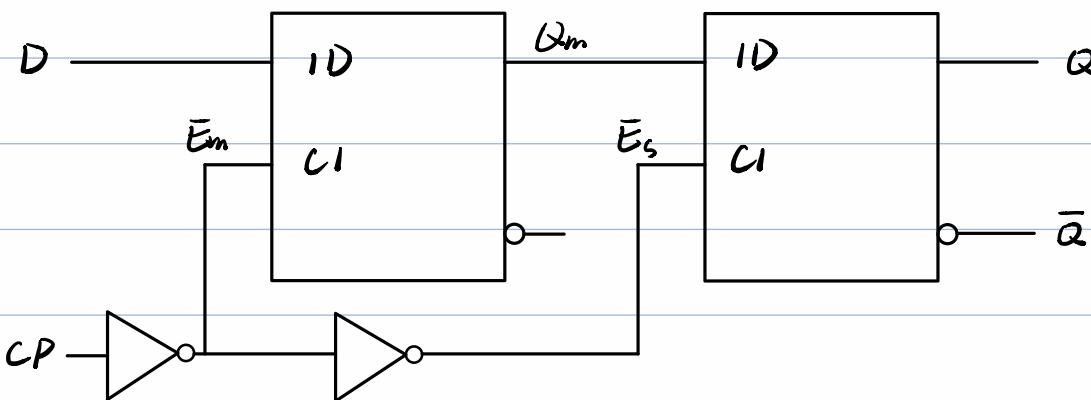


E	D	Q	\bar{Q}	功能
0	X	不变	不变	保持
1	0	0	1	Reset
1	1	1	0	Set

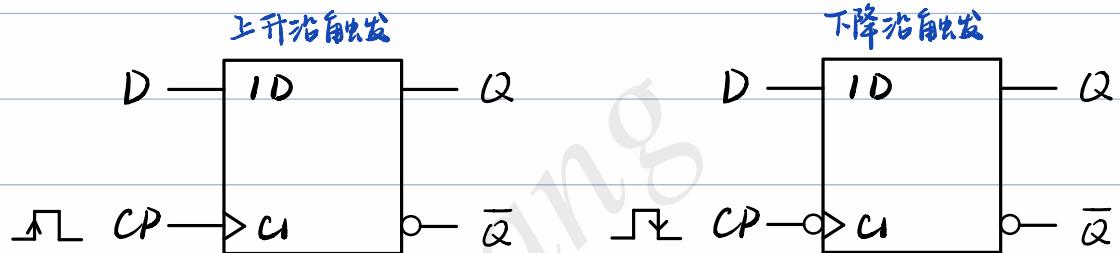
主H-D 触发器

主锁存器

从锁存器



逻辑符号



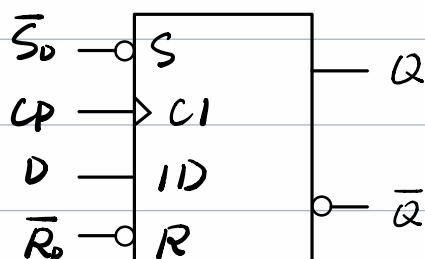
特性表

CP	D	Q^n	Q^{n+1}
x	x	x	Q^n
↑	0	x	0
↑	1	x	1
↓	0	x	0
↓	1	x	1

D 触发器特性方程 $Q^{n+1} = D$

(2) 有其他控制端的D触发器

逻辑符号



特性表

\bar{S}_D	\bar{R}_D	CP	D	Q^{n+1}	\bar{Q}^{n+1}	功能
0	1	x	x	1	0	异步置1
1	0	x	x	0	1	异步置0
0	0	x	x	1	1	输出不确定
1	1	↑	0	0	1	同步置0
1	1	↑	1	1	0	同步置1

① 异步输入端

异步：与 CP 信号无关

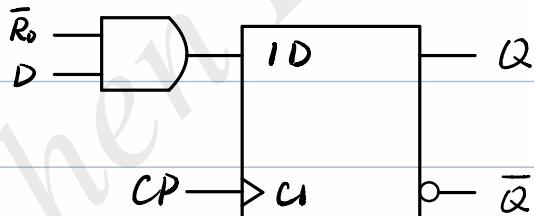
{ 直接置1端 S_D
直接置0端 R_D

② 同步输入端

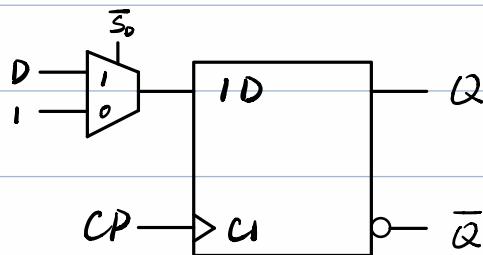
{ 同步清零端

同步置1端

同步清零端：清零输入有效，且 CP 为有效跳变沿到来时，才能将触发器清零。

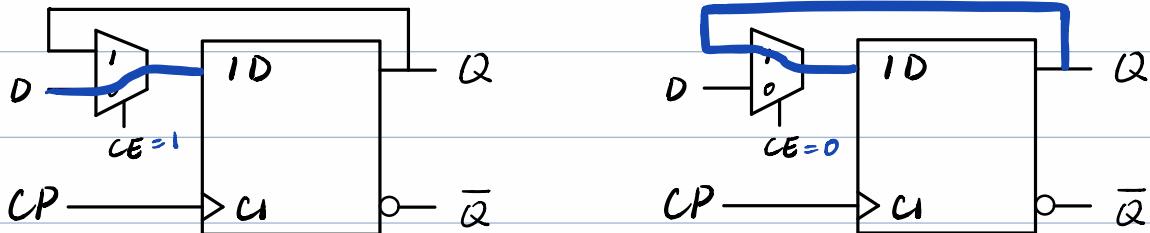


同步置1端：置1输入有效，且 CP 为有效跳变沿到来时，才能将触发器置1。



③ 带有使能端的 D 触发器

功能 { $E_n = 0$ Q 保持不变
 $E_n = 1$ 在 CP 作用下， $Q = D$



(3) 动态特性

传输延迟时间 t_{pd} ($Q \rightarrow Q$)

建立时间 t_{su} ($D \rightarrow Q$)

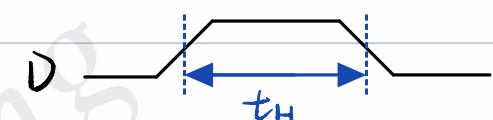
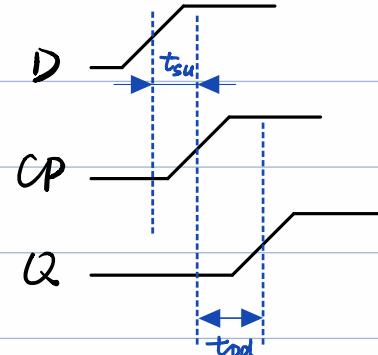
D 提前于 CP 的上升沿

保持时间 t_H (D)

D 在 CP 上升沿到来后

还应保持一段时间，

该时间的最小值为 t_H

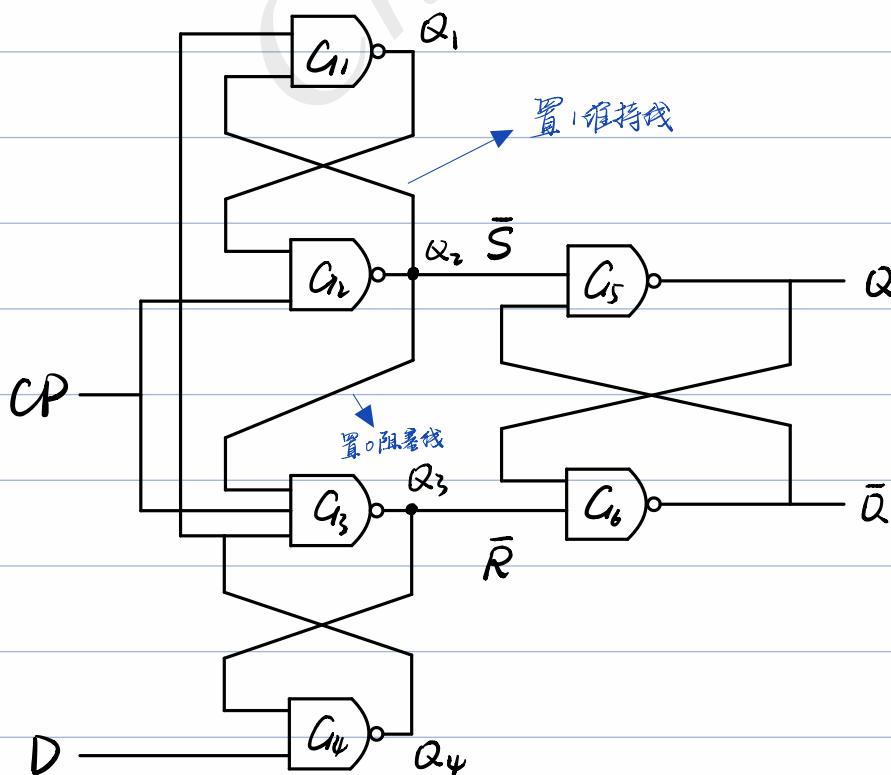


触发脉冲宽度 t_w (CP)

最高时钟频率 f_{cmax}

$$f_{cmax} = \frac{1}{T_{cmin}}$$

5 维持阻塞 D 触发器



6 触发器的逻辑功能

表明其状态与输入信号之间的逻辑关系

描述方法

特性表

特性方程

状态图

时序波形图

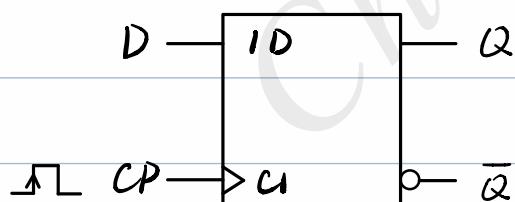
触发器按逻辑功能分类

{ D 触发器
JK 触发器
T 触发器
SR 触发器

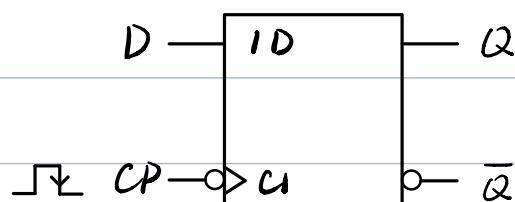
(1) D 触发器

逻辑符号

上升沿触发



下降沿触发



特性表

D Q^n Q^{n+1}

0 0 0

0 1 0

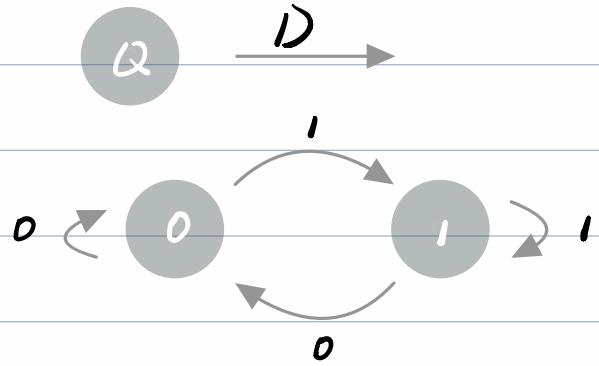
1 0 1

1 1 1

特性方程

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

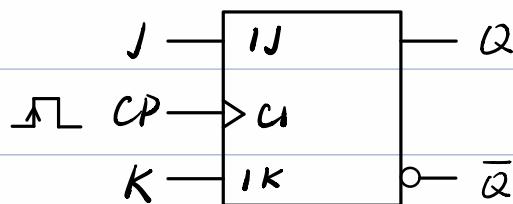
状态图



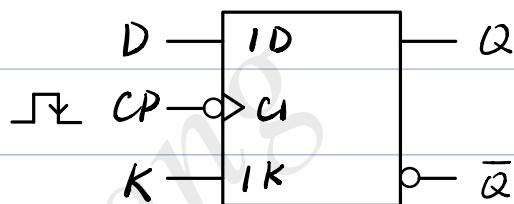
(2) JK 触发器

逻辑符号

上升沿触发



下降沿触发



特性表

J	K	Q^n	Q^{n+1}	说明
0	0	0	0	状态不变
0	0	1	1	
0	1	0	0	置 0
0	1	1	0	
1	0	0	1	置 1
1	0	1	1	
1	1	0	1	翻转
1	1	1	0	

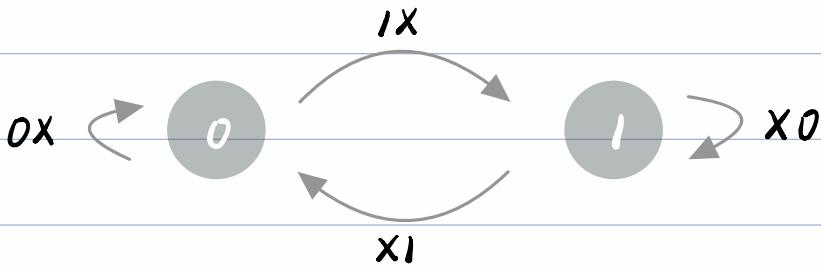
特性方程

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

状态图

Q

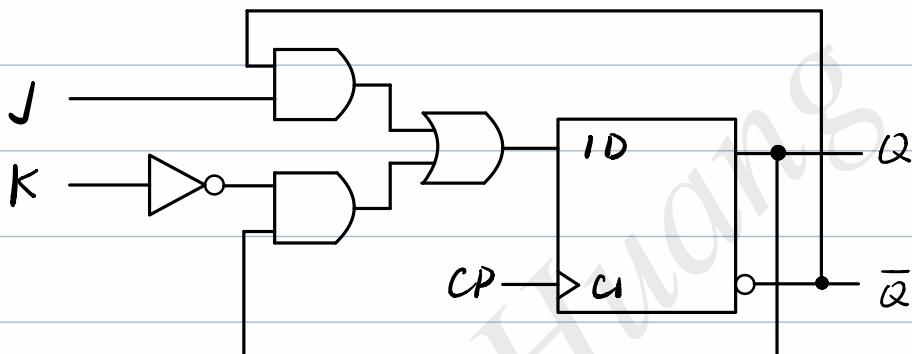
JK



用D触发器构造JK触发器

$$\begin{cases} Q^{n+1} = D \\ Q^{n+1} = J\bar{Q}^n + \bar{K}Q^n \end{cases}$$

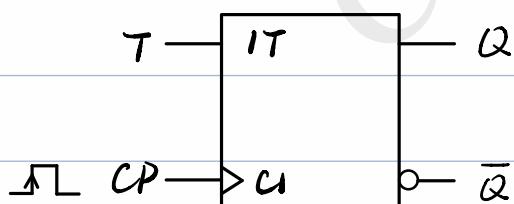
$$\Rightarrow D = J\bar{Q}^n + \bar{K}Q^n$$



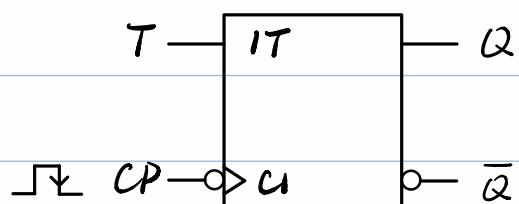
(3) T触发器

逻辑符号

上升沿触发



下降沿触发

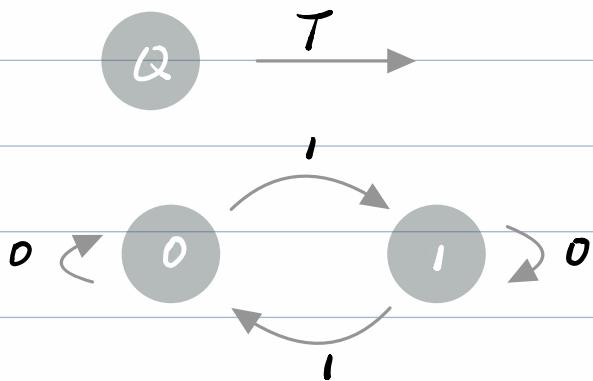


特性表

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

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

状态图

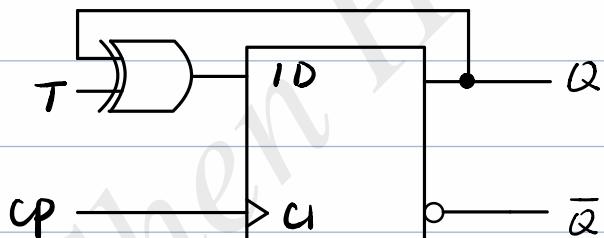


用D触发器构造T触发器

$$\begin{cases} Q^{n+1} = D \\ Q^{n+1} = T\bar{Q}^n + \bar{T}Q^n \end{cases}$$

$$\Rightarrow D = T\bar{Q}^n + \bar{T}Q^n$$

$$= T \oplus Q^n$$

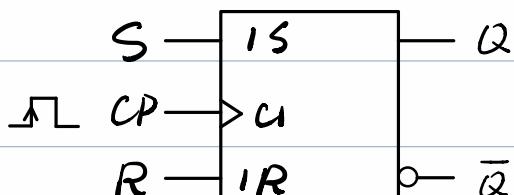


令 $T = 1$ ，得到 $Q^{n+1} = \bar{Q}^n \Rightarrow T' \text{ 触发器}$

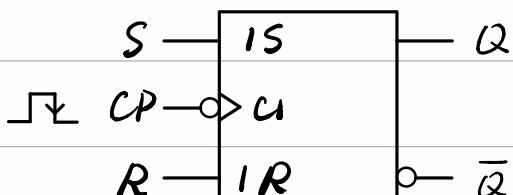
(4) SR触发器

逻辑符号

上升沿触发



下降沿触发



特性表

S	R	Q^n	Q^{n+1}	说明
0	0	0	0	状态不变
0	0	1	1	
0	1	0	0	置0
0	1	1	0	
1	0	0	1	置1
1	0	1	1	
1	1	0	不确定	不确定
1	1	1	不确定	

特性方程

$$\begin{cases} Q^{n+1} = S + \bar{R}Q^n \\ SR = 0 \text{ (约束条件)} \end{cases}$$

§5 时序逻辑电路

1 时序逻辑电路的基本概念

数字电路 { 组合逻辑电路
时序逻辑电路

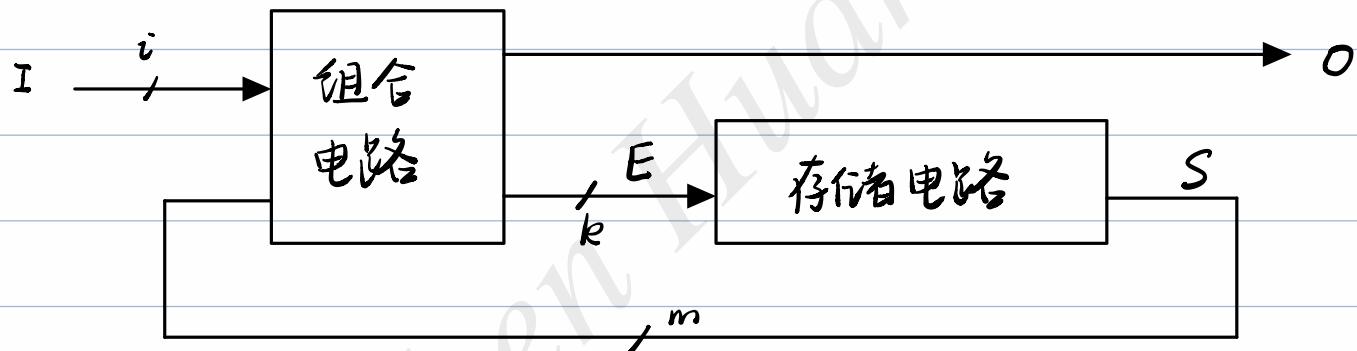
结构特点 { 含记忆元件
存在反馈通路

动作特点：输出与输入及电路状态有关

输入信号 $I = (I_1, I_2, \dots, I_i)$

输出信号 $O = (O_1, O_2, \dots, O_j)$

存储电路的输入 $E = (E_1, E_2, \dots, E_k)$



激励方程 $E = f_1(I, S)$

状态方程 $S^{n+1} = f_2(E, S^n)$

输出方程 $O = f_3(I, S)$

时序电路的分类

按动作特点

{ 同步时序电路：统一的时钟源，状态同时更新

异步时序电路：没有统一的时钟源，电路中只要有1个CP没有连在一起，就是异步……

按输出信号特点

米利型：输出与输入及触发器状态有关 $O = h(I, S)$

移存型：输出只取决于触发器状态 $O = h(s)$

时序逻辑功能的描述

方程描述：逻辑方程组

图表描述 { 状态转换表
 | 状态转换图
 | 时序图

Example

$$\text{激励方程 } D_0 = (Q_1 + Q_0) A$$

$$D_1 = \bar{Q}_0 A$$

$$\text{状态方程 } Q^{n+1} = D \Rightarrow \begin{cases} Q_0^{n+1} = (Q_1^n + Q_0^n) A \\ Q_1^{n+1} = \bar{Q}_0^n A \end{cases}$$

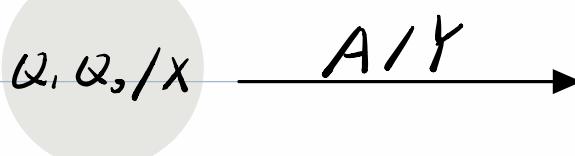
$$\text{输出方程 } X = \bar{Q}_1 Q_0$$

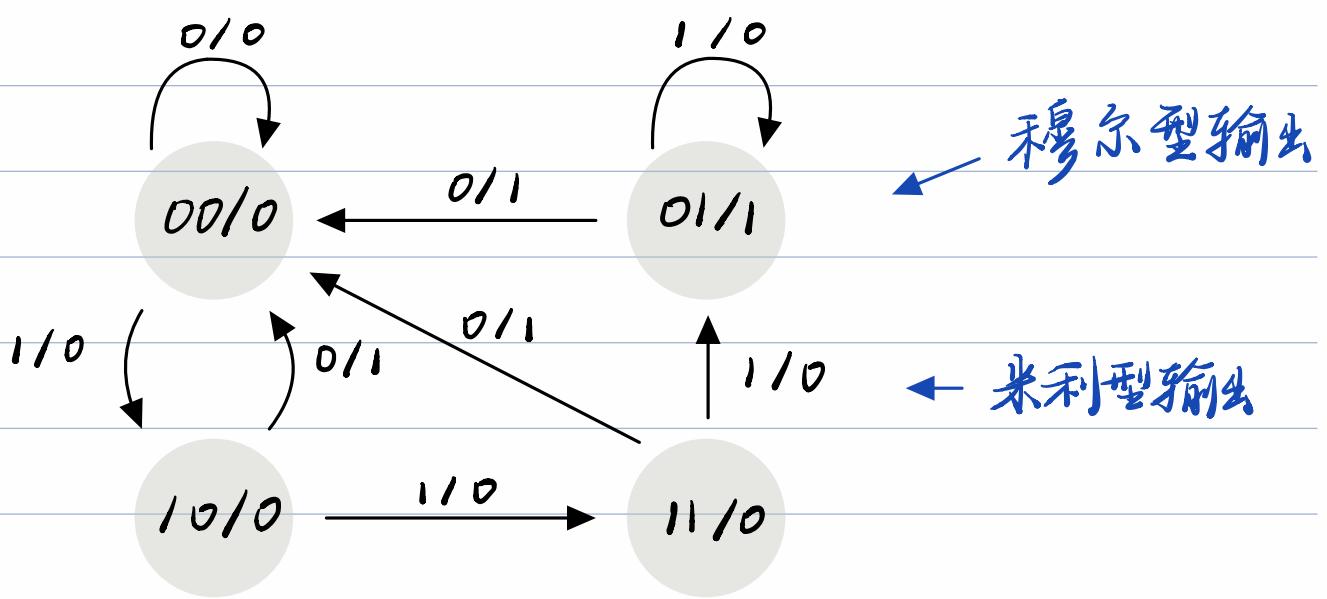
$$Y = (Q_1 + Q_0) \bar{A}$$

状态转换表

$Q_1^n Q_0^n$	Q_1^{n+1}	Q_0^{n+1}	/ Y	X
	$A=0$	$A=1$		
00	00/0	10/0		0
01	00/1	01/0		1
10	00/1	11/0		0
11	00/1	01/0		0

状态转换图





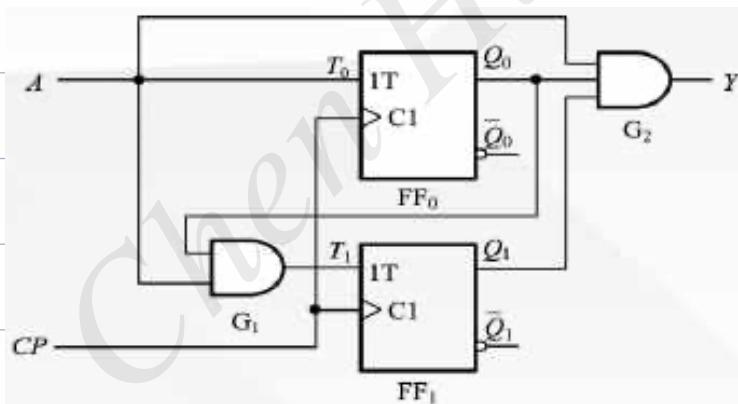
2 同步时序逻辑电路的分析

步骤 ① 列方程

② 列图表

③ 确定功能 {
逻辑功能
验证自启动}

Example



① 列方程组

激励方程 $T_1 = Q_0 \cdot A$

$T_0 = A$

状态方程 D 触发器 $Q^{n+1} = T \oplus Q^n$

$$Q_1^{n+1} = (Q_0^n \cdot A) \oplus Q_1^n$$

$$Q_0^{n+1} = A \oplus Q_0^n$$

输出方程 $Y = Q_1 \cdot Q_0 \cdot A$

② 列图表

状态转换表

$Q_1^{n+1} Q_0^{n+1} / Y$

$Q_1^n Q_0^n$

00

01

10

11

$A = 0$

0010

0110

1010

1110

$A = 1$

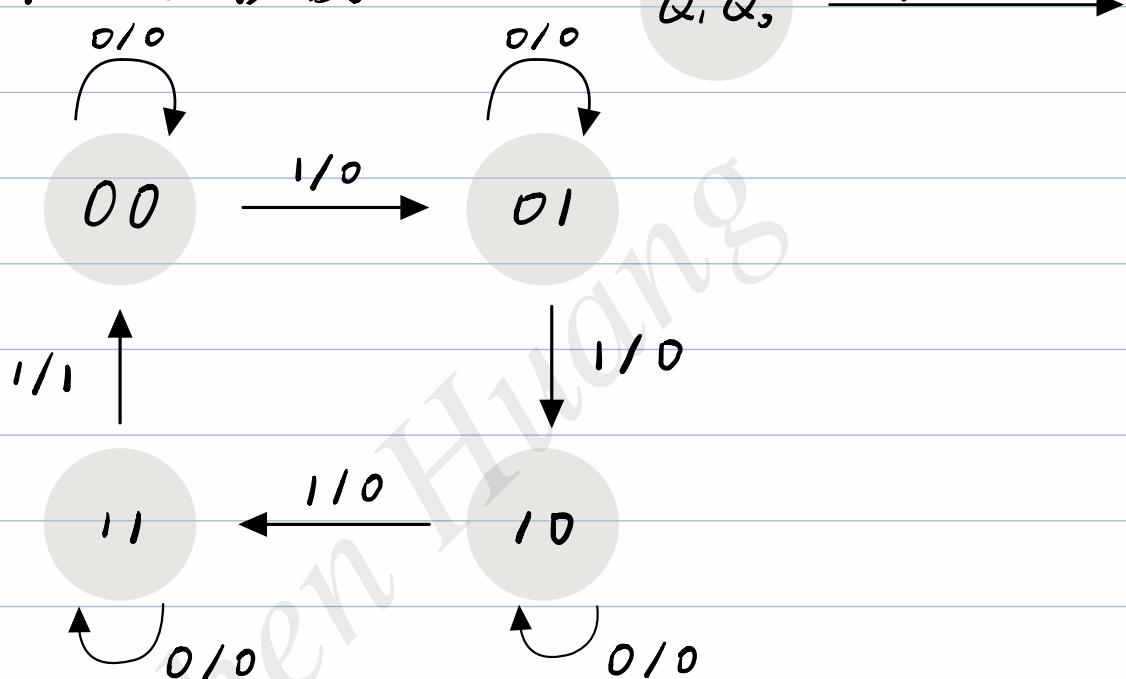
0110

1010

1110

0011

状态转换图



③ 功能分析

可擦二进制计数器

3

同步时序逻辑电路的设计

步骤聚 ① 逻辑抽象 \Rightarrow 转换图/表

② 状态化简

③ 状态分配：给每个状态赋予二进制编码

$$2^{n-1} < M \leq 2^n$$

M : 状态数 n : 触发器个数

④ 选择触发器类型

⑤ 确定激励方程和输出方程

⑥ 检查能否自启动，画逻辑图

Example 1

用D触发器设计一个8421BCD码同步十进制加计数器。

① 分析：对时钟脉冲进行计数，每个脉冲作用下+1十个状态 \Rightarrow 4个D触发器 $Q = (Q_3, Q_2, Q_1, Q_0)$
穆尔型电路

② 列转换表

现态				次态				激励信号			
Q_3^n	Q_2^n	Q_1^n	Q_0^n	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	1	0	0	0	1	0
0	0	1	0	0	0	1	1	0	0	1	1
0	0	1	1	0	1	0	0	0	1	0	0
0	1	0	0	0	1	0	1	0	1	0	0
0	1	0	1	0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	1	0	1	1	1
0	1	1	1	1	0	0	0	0	1	0	0
1	0	0	0	1	0	0	1	1	0	0	1
1	0	0	1	0	0	0	0	0	0	0	0

③

		$Q_1 Q_0$	00	01	11	10
$Q_3 Q_2$		00	0	0	0	0
00	01	00	0	0	1	0
01	01	01	X	X	X	X
11	11	11	1	0	X	X
10	10	10	1	0	X	X

$$D_3 = Q_2^n Q_1^n Q_0^n + Q_3^n \bar{Q}_0^n$$

D_2

Q_3, Q_2	Q_1, Q_0	00	01	11	10
00	0	0	1	0	
01	1	1	0	1	
11	X	X	X	X	
10	0	0	X	X	

$$D_2 = Q_2^n \bar{Q}_1^n + Q_2^n \bar{Q}_0^n + \bar{Q}_2^n Q_1^n Q_0^n$$

D_1

Q_3, Q_2	Q_1, Q_0	00	01	11	10
00	0	1	0	1	
01	0	1	0	1	
11	X	X	X	X	
10	0	0	X	X	

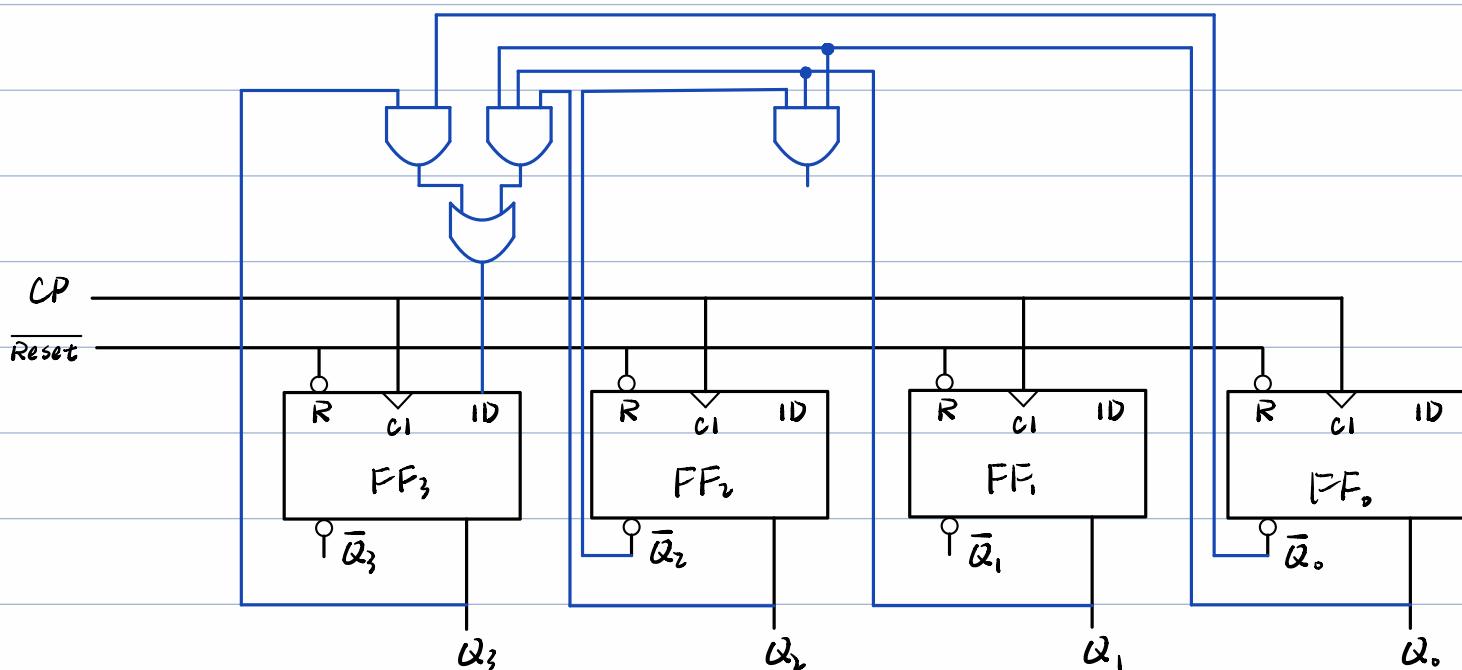
$$D_1 = \bar{Q}_3^n \bar{Q}_1^n Q_0^n + Q_1^n \bar{Q}_0^n$$

D_0

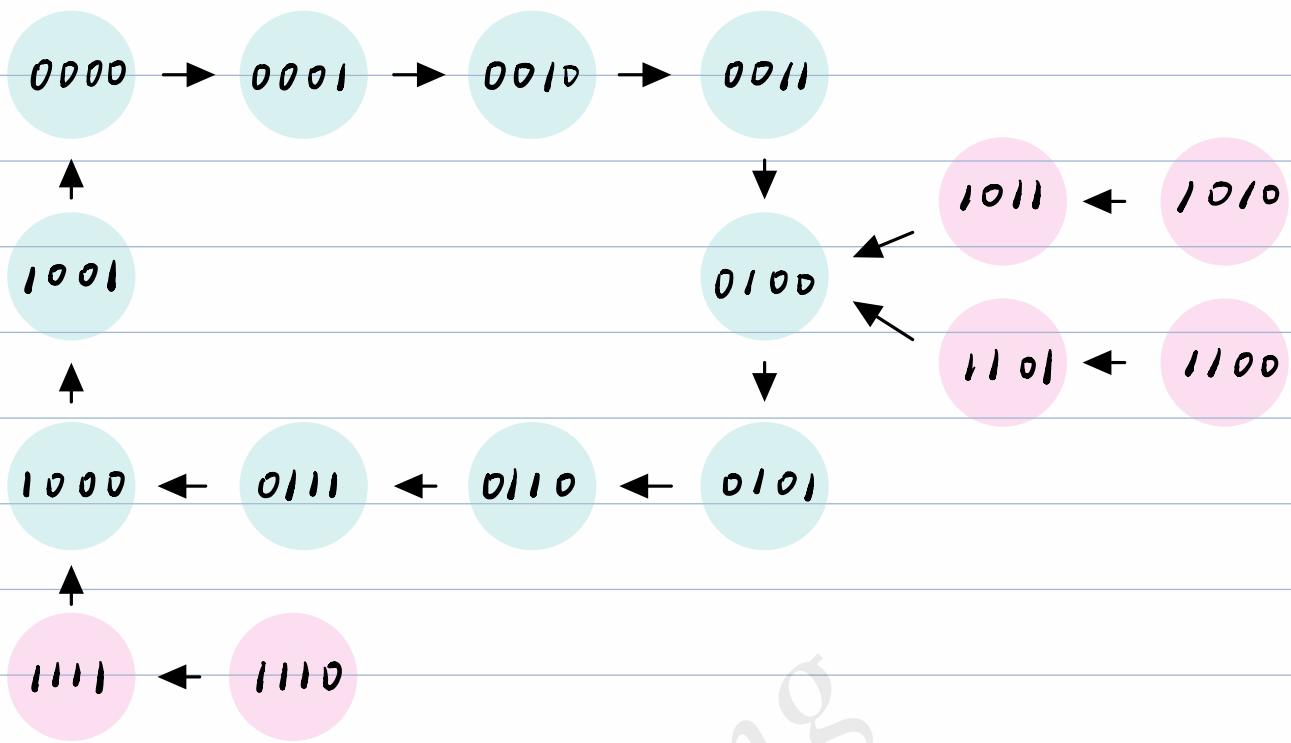
Q_3, Q_2	Q_1, Q_0	00	01	11	10
00	1	0	0	1	
01	1	0	0	1	
11	X	X	X	X	
10	1	0	X	X	

$$D_0 = \bar{Q}_0^n$$

④ 亂畫真摃圖 (略. 不想亂了)



⑤ 检查自启动功能



Example 2

设计一个串行数据检测器。电路的输入信号 A 是与时钟脉冲同步的串行数据。要求当输入 A 出现 110 序列时，输出信号 Y 为 1，否则为 0。

① 逻辑抽象

输入变量 A

输出变量 Y

初始状态 S_0

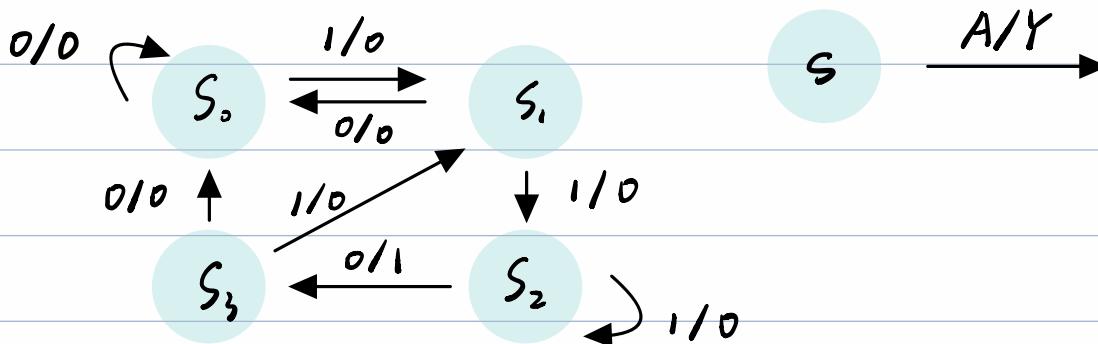
A 输入 1 后 S_1

A 输入 11 后 S_2

A 输入 110 后 S_3

状态数：4 个

② 画原始状态图，列原始转换表

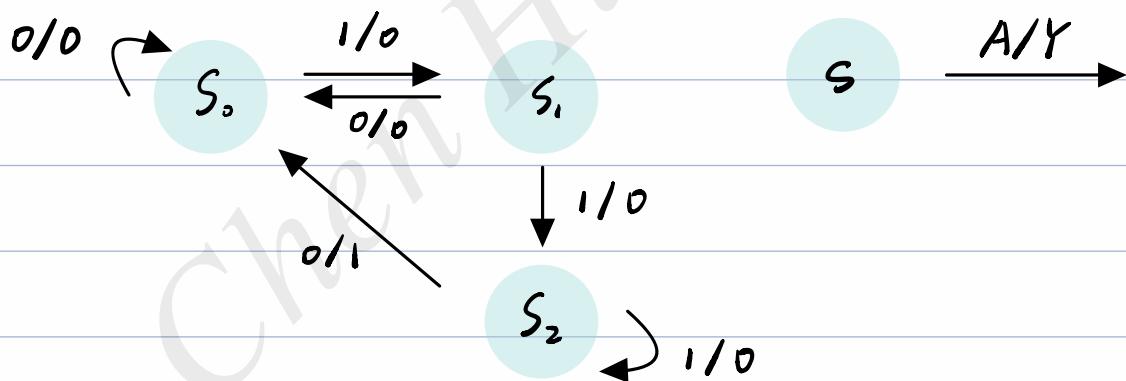


S^n	S^{n+1}/Y	
	$A=0$	$A=1$
s_0	$s_0/0$	$s_1/0$
s_1	$s_0/0$	$s_2/0$
s_2	$s_3/1$	$s_2/0$
s_3	$s_0/0$	$s_1/0$

合并

③ 状态简化

S^n	S^{n+1}/Y	
	$A=0$	$A=1$
s_0	$s_0/0$	$s_1/0$
s_1	$s_0/0$	$s_2/0$
s_2	$s_0/1$	$s_2/0$



④ 状态编码

$$s_0 = 00$$

$$s_1 = 01$$

$$s_2 = 11$$

S^n	S^{n+1}/Y	
	$A=0$	$A=1$
00	00/0	01/0
01	00/0	11/0
11	00/1	11/0

⑤ 状态数 3 触发器：2个

采用 CP 下降沿敏感的 JK 触发器

⑥ 表激励方程和输出方程
状态转换真值表

Q_1^n	Q_0^n	A	Q_1^{n+1}	Q_0^{n+1}	Y	J ₁	K ₁	J ₀	K ₀
0	0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	0	X	1	X
0	1	0	0	0	0	0	X	X	1
0	1	1	1	1	0	1	X	X	0
1	1	0	0	0	1	X	1	X	1
1	1	1	1	0	0	X	0	X	0

卡诺图化简

J_1

		$Q_1 Q_0$	00	01	11	10
		A	0	0	X	X
		A	1	0	1	X
0	0	0	0	X	X	X
1	0	1	1	X	X	X

K_1

		$Q_1 Q_0$	00	01	11	10	
		A	0	X	X	1	X
		A	1	X	X	0	X
0	0	0	X	X	1	X	
1	0	1	X	X	0	X	

J_0

		$Q_1 Q_0$	00	01	11	10
		A	0	0	X	X
		A	1	1	X	X
0	0	0	0	X	X	X
1	0	1	1	X	X	X

K_0

		$Q_1 Q_0$	00	01	11	10	
		A	0	X	1	1	X
		A	1	X	0	0	X
0	0	0	X	1	1	X	
1	0	1	X	0	0	X	

Y

		$Q_1 Q_0$	00	01	11	10
		A	0	0	1	X
		A	1	0	0	X
0	0	0	0	0	1	X
1	0	1	0	0	0	X

$Y = Q_1 \bar{A}$

		$Q_1 Q_0$	00	01	11	10
		A	0	0	1	X
		A	1	0	0	X
0	0	0	0	0	1	X
1	0	1	0	0	0	X

激励方程 $J_1 = Q_0 \cdot A$

$$K_1 = \bar{A}$$

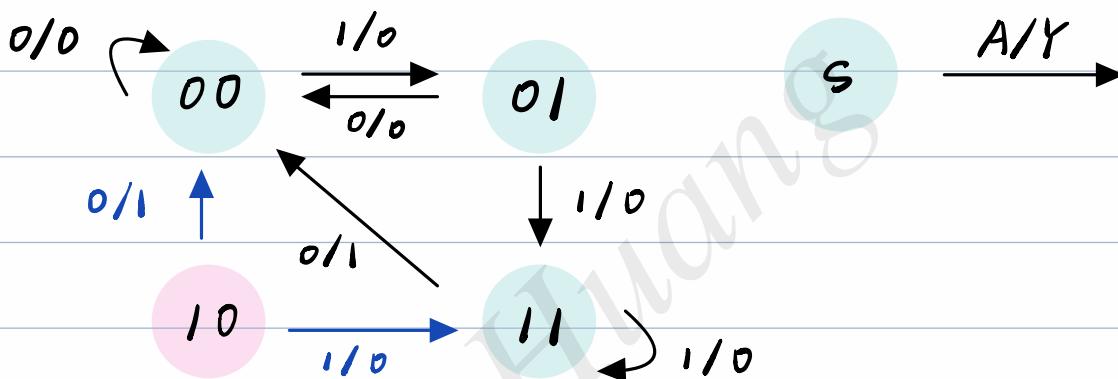
$$J_0 = A$$

$$K_0 = \bar{A}$$

输出方程 $Y = Q_1 \cdot \bar{A}$

⑦ 检查自启动能力 $Q^{n+1} = J \bar{Q}^n + E Q^n$

$$Q_1^n Q_0^n = 10 \quad \begin{cases} A=0 & Q_1^{n+1} Q_0^{n+1} = 00 \\ A=1 & Q_1^{n+1} Q_0^{n+1} = 11 \end{cases} \quad Y = 1 \quad Y = 0$$



无效状态能进入有效循环，故能自启动

但由于 需要修改卡诺图

Y

A $Q_1 Q_0$

		00	01	11	10
0	0	0	1	0	
1	0	0	0	X	

$$Y = Q_1 Q_0 \bar{A}$$

4 异步时序逻辑电路的分析

{ 没有统一的时钟脉冲，电路状态不是同时更新的
| 触发器有时钟脉冲沿作用时，状态发生变化，
反之，则状态不变。

步骤

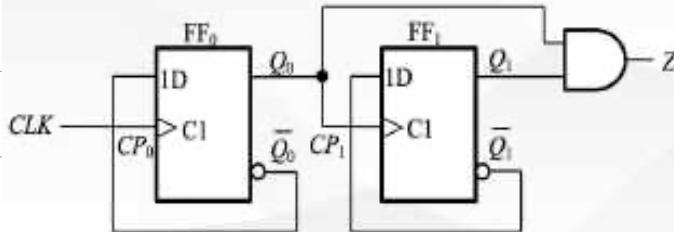
① 写方程

时钟方程、激励方程、状态方程、输出方程

② 列图表

③ 描述逻辑功能、检查自启动功能

Example



① 时钟方程

$$CP_0 = CLK$$

$$CP_1 = Q_0$$

激励方程

$$D_0 = \bar{Q}_0$$

$$D_1 = \bar{Q}_1$$

状态方程

$$Q^{n+1} = D_0 CP_n + Q^n \bar{CP}_n$$

$$Q_0^{n+1} = \bar{Q}_0^n CP_n + Q_0^n \bar{CP}_n$$

$$Q_1^{n+1} = \bar{Q}_1^n CP_n + Q_1^n \bar{CP}_n$$

输出方程

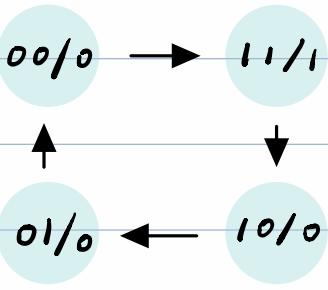
$$Z = Q_1 Q_0$$

② 状态转换表

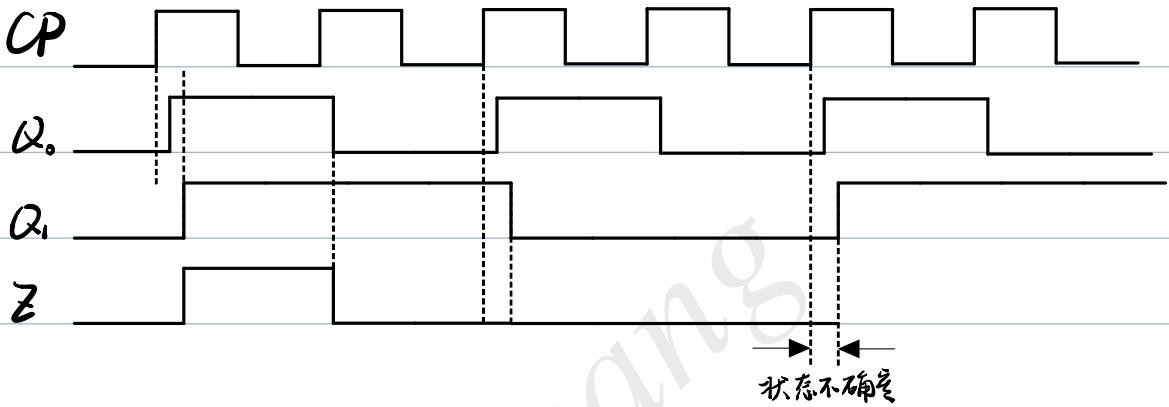
$Q_1^n Q_0^n$	CP_1	CP_0	$Q_1^{n+1} Q_0^{n+1}$	Z
0 0	1	1	1 1	0
0 1	0	1	0 0	0
1 0	1	1	0 1	0
1 1	0	1	1 0	1

状态图

$Q_1 Q_0 / Z \rightarrow$



时序图



② 逻辑功能分析 异步二进制减计数器

5 寄存器和移位寄存器

寄存器

数字系统中用来存储一组二值代码的逻辑部件

一个寄存器可存储一位二进制代码

存储n位二值代码需要n个触发器

移位寄存器 (同步时序电路)

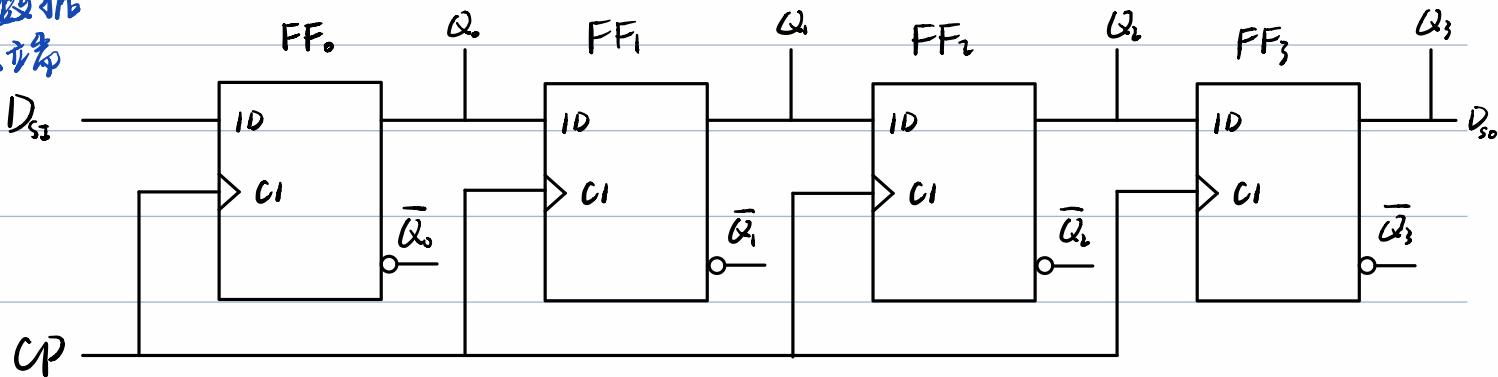
既能寄存数码，又能在时钟脉冲的作用下使数码向高位或低位移动。

{ 单向移位 { 左移
 双向移位 右移

基本串行移位寄存器 (只能用边沿敏感触发器)

并行数据输出端

串行数据
输入端



$$\text{激励方程 } D_0 = D_{SJ} \quad D_1 = Q_0^n \quad D_2 = Q_1^n \quad D_3 = Q_2^n$$

$$\text{状态方程 } Q_0^{n+1} = D_{SJ} \quad Q_1^{n+1} = Q_0^n \quad Q_2^{n+1} = Q_1^n \quad Q_3^{n+1} = Q_2^n$$

n 位二进制数需要 n 个触发器

多工能双向移位寄存器

6 计数器

计数器的功能

对输入的时钟脉冲进行计数

分类

同步
异步

} 二进制
十进制
任意进制

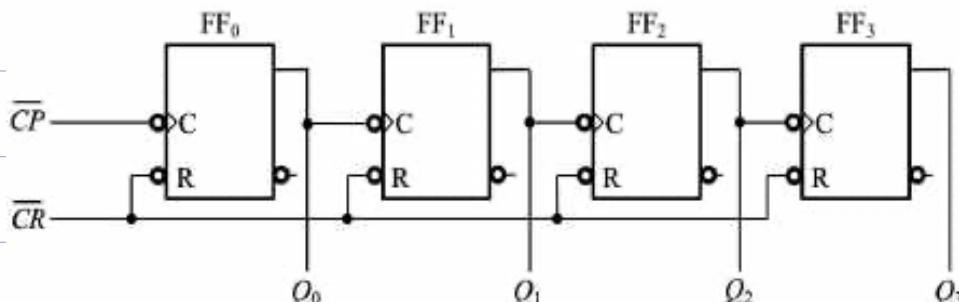
} 加法
减法
可逆

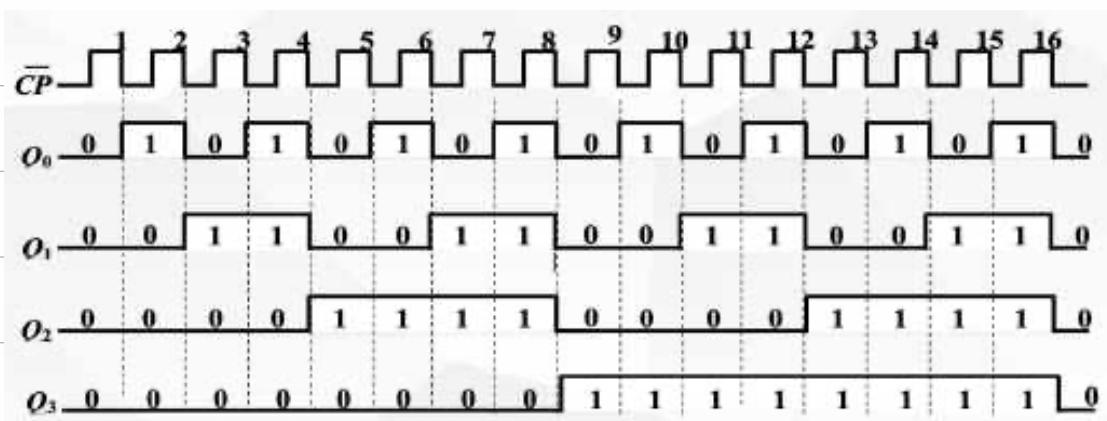
N位二进制计数器

按照二进制数自然递增或递减而工作的计数器

n 位二进制计数器的模为 2^n , 由 n 个触发器组成

4位异步二进制加计数器





4位同步二进制加计数器

T 触发器：T=1 翻转，T=0 保持

$$\left\{ \begin{array}{l} T_0 = 1 \\ T_1 = Q_0 \\ T_2 = Q_0 Q_1 \\ T_3 = Q_0 Q_1 Q_2 \end{array} \right.$$

8 集成计数器的应用

M进制 \longrightarrow N进制

$N < M$

反馈清零法

利用异步置零输入端，在M进制计数的计数过程中，跳过 $M-N$ 个状态，得到N进制计数器

反馈置数法

利用同步置数端，在M进制计数的计数过程中，跳过 $M-N$ 个状态，得到N进制计数器

$N > M$

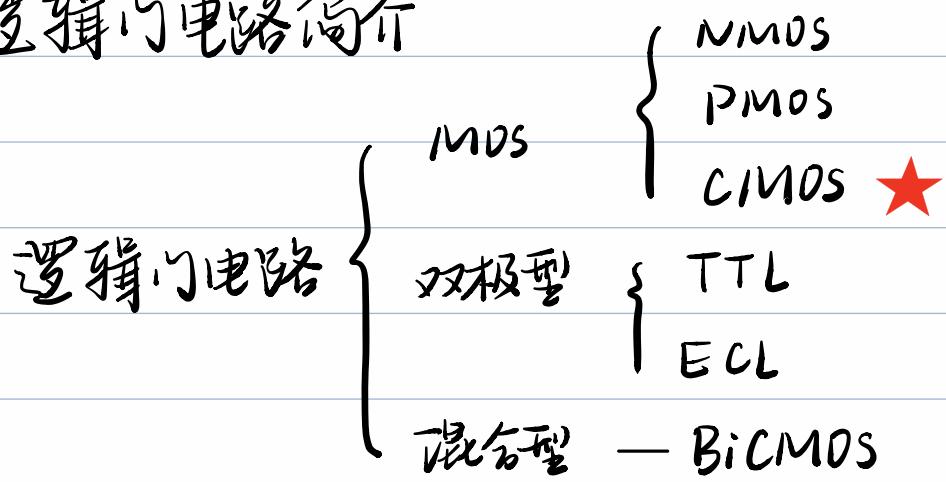
多个芯片连接扩展

并行进位：低位片的进位作为高位片的使能（同步）

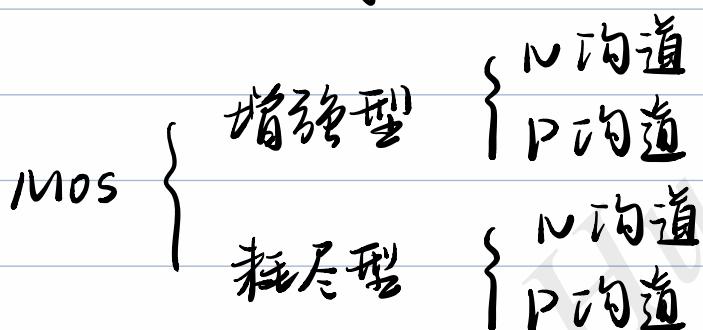
串行进位：低位片的进位作为高位片的时钟（异步）

8.6 逻辑门电路

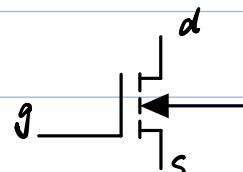
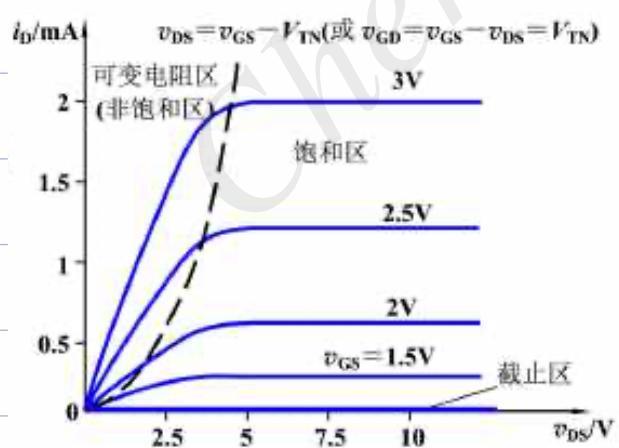
1 逻辑门电路简介



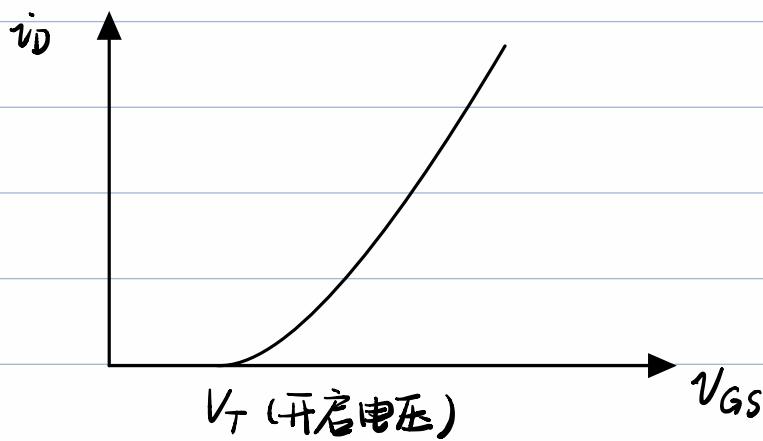
2 NMOS 管及其开关特性



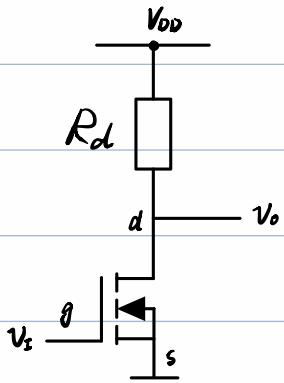
e.g. N沟道增强型 MOS管



详见模电笔记



MOS管开关电路



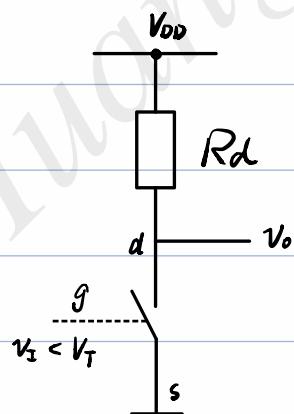
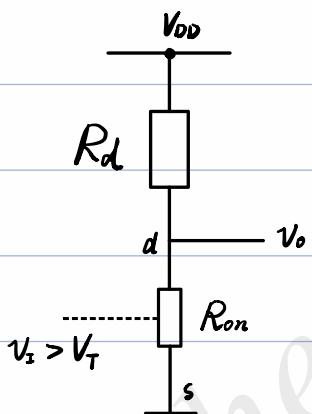
R_d 限流，功耗大
可用一个PMOS代替 R_d
 \Rightarrow 构成CMOS反相器

① $v_i < v_t \rightarrow$ 截止 \rightarrow 断开 \rightarrow 输出 1

输出电压 $v_o = V_{DD}$

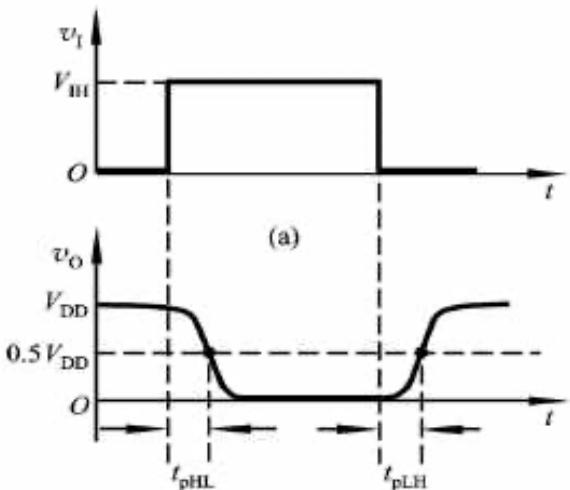
② $v_i > v_t \rightarrow$ 可变电阻区 \rightarrow 闭合 \rightarrow 输出 0

v_{GS} 足够大时， R_{on} 很小， $R_d \gg R_{on}$



MOS管相当于一个由 v_{GS} 控制的无触点开关
输入低电平，输出高电平
输入高电平，输出低电平

动态特性



存在极间电容

t_{pHl} t_{plh}
传输延迟时间

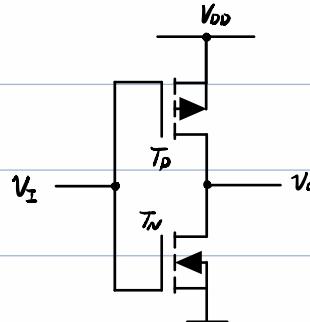
3 基本CMOS逻辑门电路

(1) CMOS 反相器

$$L = \overline{A}$$

用PMOS代替电阻

即互补NMOS和PMOS构成CMOS 反相器



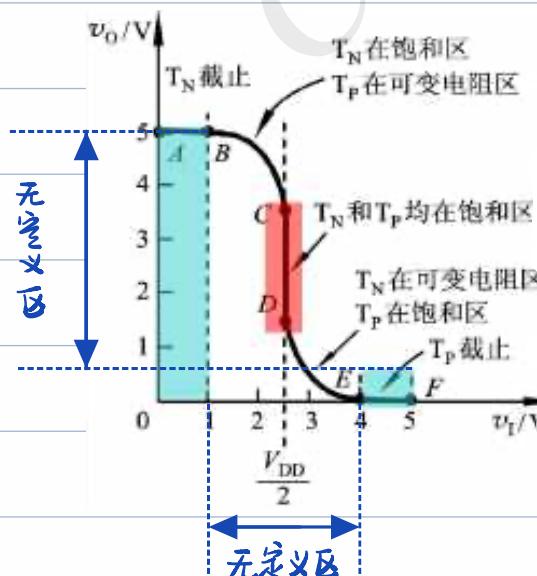
输入高电平 $v_I \approx V_{DD}$ 输出低电平 $v_O \approx 0$
输入低电平 $v_I = 0$ 输出高电平 $v_O \approx V_{DD}$

特点

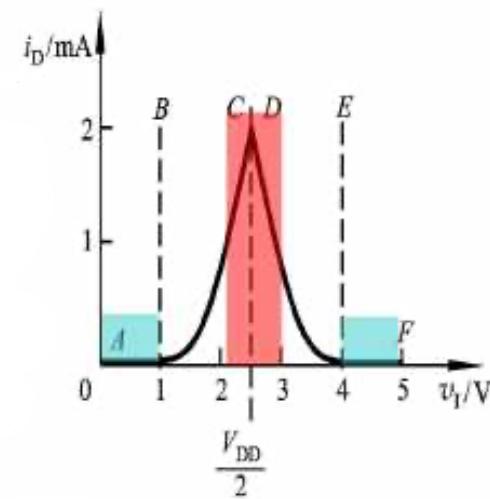
- ①无论 v_I 高/低电平, T_N 和 T_P 总是一个导通而另一个截止, CMOS 的静态功耗几乎为零.
- ②MOS 管导通电阻低, 截止电阻高, 充放电时间常数小, 具有更强的带负载能力.
- ③MOS 管 $I_G \approx 0$, 输入电阻高.

电压和电流传输特性

$$v_O = f(v_I)$$



$$i_D = f(v_I)$$



输入 / 输出逻辑电平
允许波动范围

(2) 其他基本CMOS逻辑门电路

CMOS与非门

2个NMOS串联 & 2个PMOS并联

$$L = \overline{AB}$$

几个输入端的与非门必须有n个NMOS管串联和n个PMOS管并联

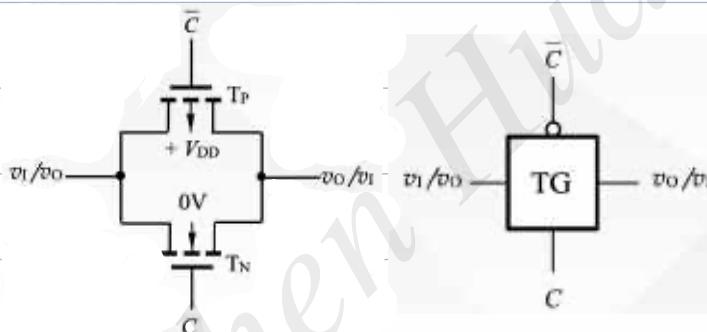
CMOS或非门

2个NMOS并联 & 2个PMOS串联

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

CMOS传输门

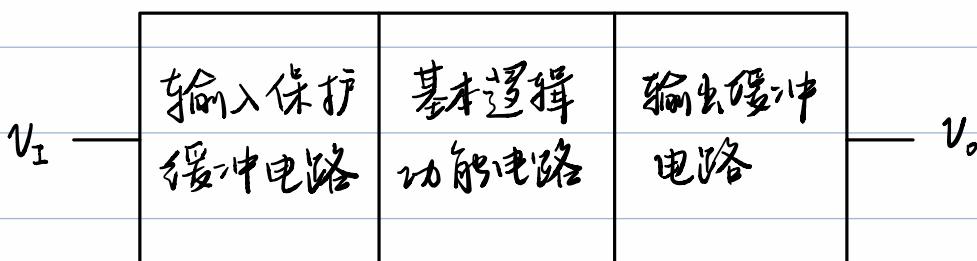
NMOS和PMOS三极管源极和漏极分别连在一起



C	\bar{C}
1	0
其余	传输
	断开

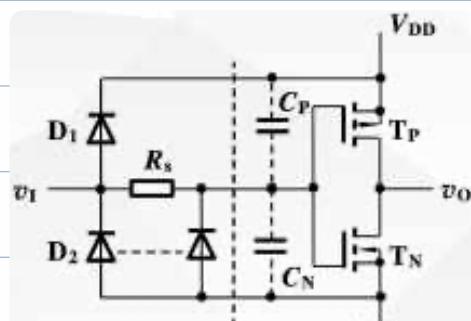
4 CMOS逻辑门的不同输出结构

(1) 保护和缓冲电路



输入端保护电路

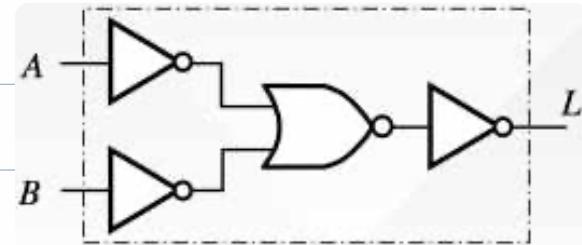
当输入电压不在正常范围时，二极管导通，限制电压的增加。



CMOS逻辑门的缓冲电路

输入、输出端加了反相器
作为缓冲电路

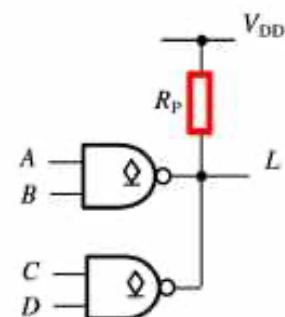
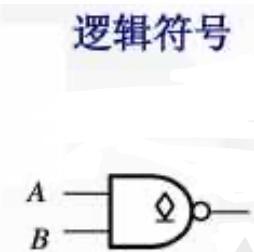
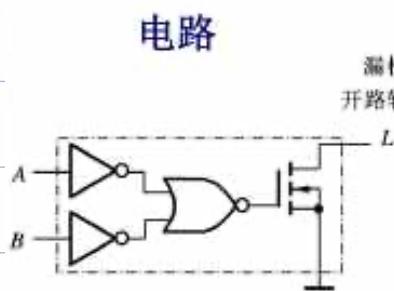
$$L = \overline{\overline{A} + \overline{B}} = \overline{AB}$$



(2) CMOS逻辑门的不同输出结构

CMOS漏极开路(OD)门与非门

① 结构及工作原理



$$L = \overline{AB} \cdot \overline{CD}$$

② 上拉电阻对OD门动态性能的影响

- 当其他门电路作为OD门的负载时，OD门称为驱动门，后接的称为负载门。
- 上拉电阻 R_P 影响OD门的开关速度。
 R_P 越小，负载电容的充电时间常数越小，开关越快。
- 避免用OD门驱动大的电容性负载。

③ 上拉电阻的计算

$$R_{P(\min)} = \frac{V_{DD} - V_{OL(\max)}}{I_{OL(\max)} - I_{IL(\text{total})}}$$

$$R_{P(\max)} = \frac{V_{DD} - V_{OL(\max)}}{I_{OL(\max)} - I_{IL(\text{total})}}$$

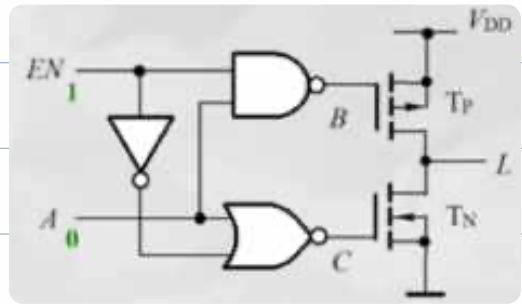
④ 应用

实现多个逻辑门输出端的线与
直接驱动大电流负载

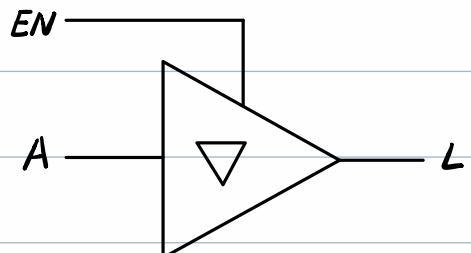
实现逻辑电平转换

三态(TSL)输出门电路

使能 EN	输入 A	输出 L
1	0	0
1	1	1
0	X	高阻



逻辑功能：高电平有效的同相逻辑门



5 CMOS逻辑门电路的重要参数

输入和输出的高、低电平 (高: 1 低 0)

$$V_{IL(\max)}, V_{IH(\min)}, V_{OL(\max)}, V_{OH(\min)}$$

噪声容限：保证输出电平不变下，输入电平允许波动的范围

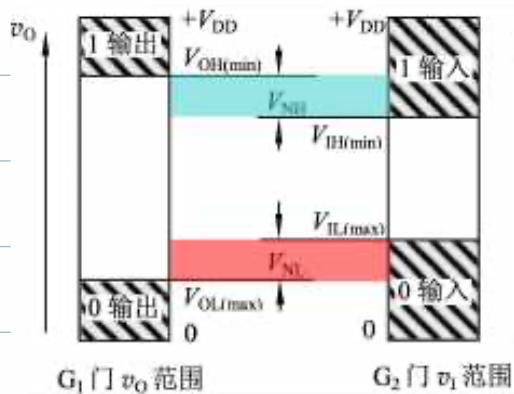
表示门电路的抗干扰能力

负载输入高电平时的噪声容限

$$V_{NH} = V_{OH(\min)} - V_{IH(\min)}$$

负载输入低电平时的噪声容限

$$V_{NL} = V_{IL(\max)} - V_{OL(\max)}$$



传输延迟时间

$$t_{PHL} = t_{PLH}$$

功耗

静态功耗

门电路空载时电源总电流 I_{DD} 与电源电压 V_{DD} 的乘积

- CMOS 静态功耗很低，有动态功耗

- TTL 静态功耗是主要的

动态功耗 电路在输出状态转换时的功耗

$$P_L = (C_{PD} + C_L) V_{DD}^2 f$$

延时—功耗积 $D \cdot P = t_{pd} P_D$

扇入/扇出数

扇入数 取决于逻辑门的输入端个数

扇出数 正常工作情况下，能带同类门的最大数目

$$\begin{cases} \text{高电平扇出数} & N_{OH} = \frac{I_{OH}(\text{驱动门})}{I_{OL}(\text{负载门})} \\ \text{低电平扇出数} & N_{OL} = \frac{I_{OL}(\text{驱动门})}{I_{OL}(\text{负载门})} \end{cases}$$

若 $N_{OH} \neq N_{OL}$ ，二者中数值小的为扇出数

6 逻辑描述中的几个问题

(1) 正负逻辑问题

正负逻辑规定

正逻辑 $\begin{cases} \text{高电平} \rightarrow 1 \\ \text{低电平} \rightarrow 0 \end{cases}$

负逻辑 $\begin{cases} \text{高电平} \rightarrow 0 \\ \text{低电平} \rightarrow 1 \end{cases}$

正负逻辑的等效变换

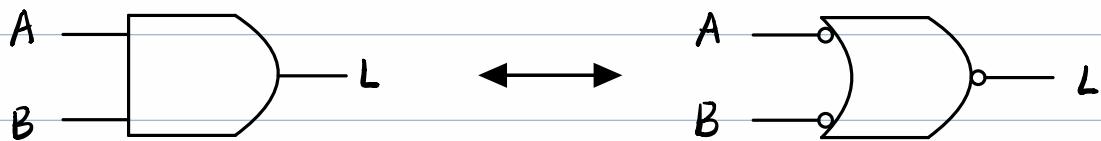
与非 \Leftrightarrow 或非

与 \Leftrightarrow 或

非 \Leftrightarrow 非

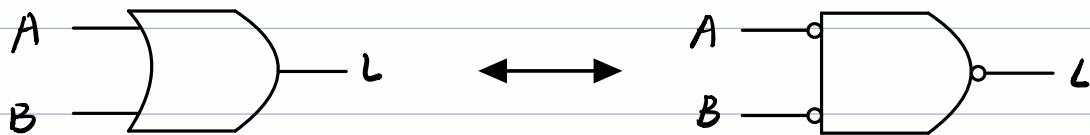
(2) 基本逻辑门的等效符号及其应用

$\begin{cases} \text{高电平有效} \\ \text{低电平有效} \end{cases}$



$$L = AB$$

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



$$L = A + B$$

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

7 逻辑门电路使用中的几个实际问题

(1) 接口问题 (电压兼容、电流匹配)

门电路输入、输出电压范围

电压兼容: 驱动的输出必须要在负载的输入范围内

输出问题: 驱动器件必须为负载提供足够的拉电流或灌电流。

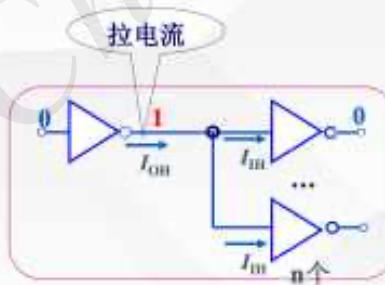


图4 负载器件的电流要求-拉电流

$$|I_{OH(max)}| \geq |I_{OH(total)}|$$

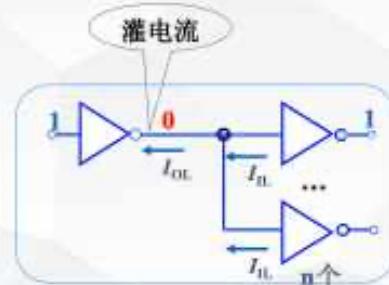


图5 负载器件的电流要求-灌电流

$$|I_{OL(max)}| \geq |I_{OL(total)}|$$

解决方案：电平转换器、外接上拉电阻……

(2) 驱动其它负载

显示性负载

机电性负载

(3) 抗干扰措施

多余输入端的处理措施

△ 原则：不改变电路工作状态及可靠性

① 与其他输入端并联

② 与门、与非门输入端接正电源

或门、或非门输入端接地

去耦合滤波电容

在直流电源和地之间接去耦合滤波电容，消除干扰信号

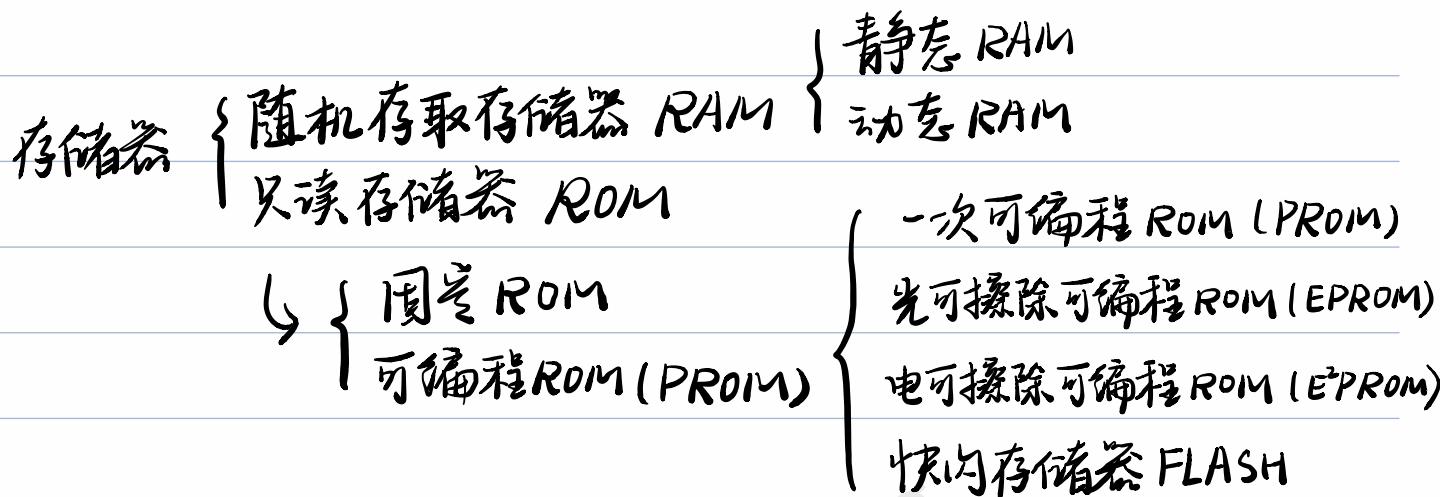
接地和安装工艺

电源地、信号地分开；模拟地、数字地分开

8.7 半导体存储器

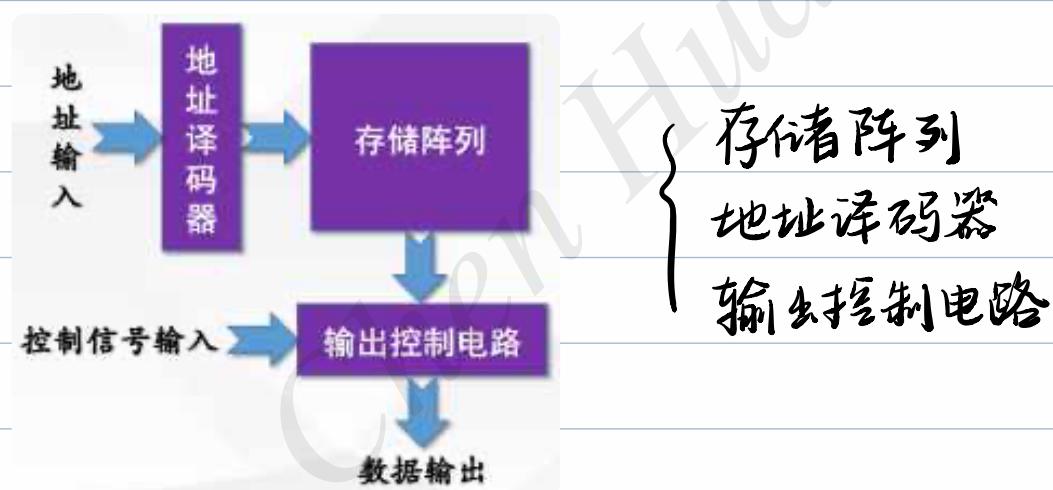
1 半导体存储器概述和分类

半导体存储器用来存储大量的二值数据



2 ROM的结构和工作原理

(1) 基本结构



字：计算机中作为一个整体被存取传递的一组数据

字数：字的总量 = 2^n ← 地址线总数量 / 地址码位数

字长：字中所含位数

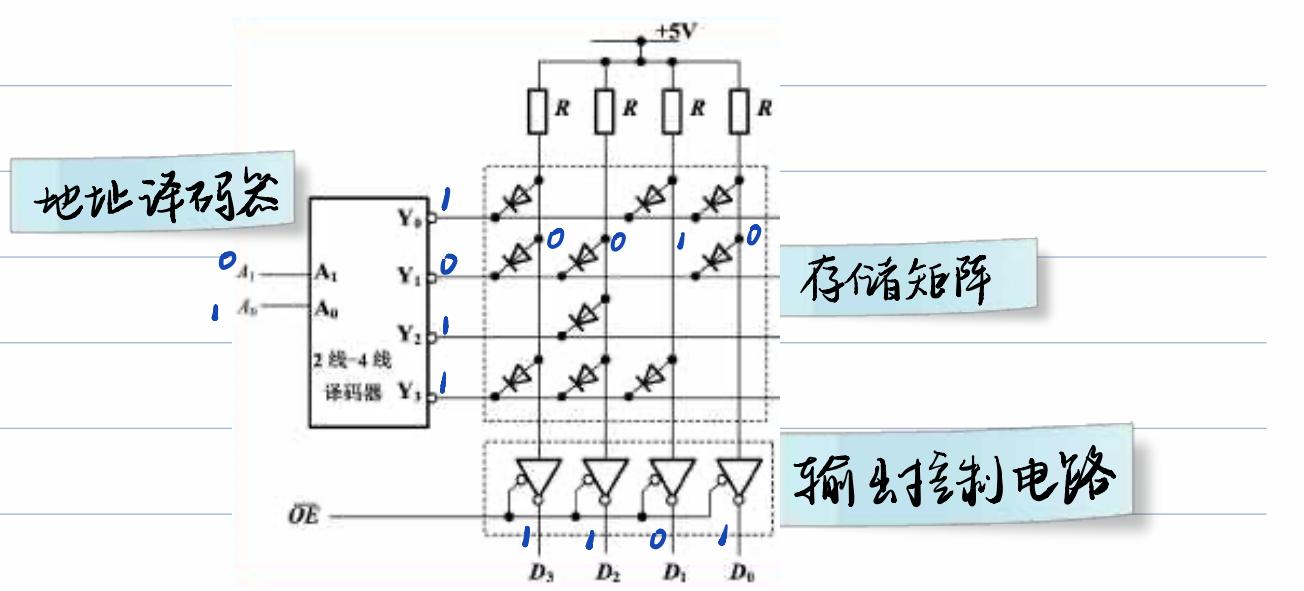
地址单元：构成同一个字的存储单元

地址：每个字的编号

存储容量：字数 × 位数 ($2^n \times m$)

(2) 控制原理

通过字线与位线交叉处是否连接有二极管来存储二值信息。



| 交点处有二极管 → 1
 | 无 0
 | { $\overline{OE} = 0$ 高阻
 | $\overline{OE} = 1$ 数据输出

使能	地址	$\overline{Y_0}$	$\overline{Y_1}$	$\overline{Y_2}$	$\overline{Y_3}$	D ₃	D ₂	D ₁	D ₀	内容
\overline{OE}	A ₁ A ₀	0	1	1	1	1	0	1	1	1011
0	0 0	0	1	1	1	1	1	0	1	1011
0	0 1	1	0	1	1	1	1	0	1	1011
0	1 0	1	1	0	1	0	1	0	0	0100
0	1 1	1	1	1	0	1	1	1	0	1110
1	x x									高阻

(3) 二维译码与存储阵列

行列交叉处连接 MOS 管

4 RAM 的结构和工作原理

{ 与 ROM 的最大区别：易失性

优点：可以快速随机存储数据

RAM { 静态 RAM (SRAM) 存储单元类似锁存器
动态 RAM (DRAM) 用电容器存储电荷保存 0/1

(1) SRAM

SRAM 的基本结构及输入输出

地址线 n $A_0 \sim A_{n-1}$

双向数据线 m $I/O_0 \sim I/O_{m-1}$

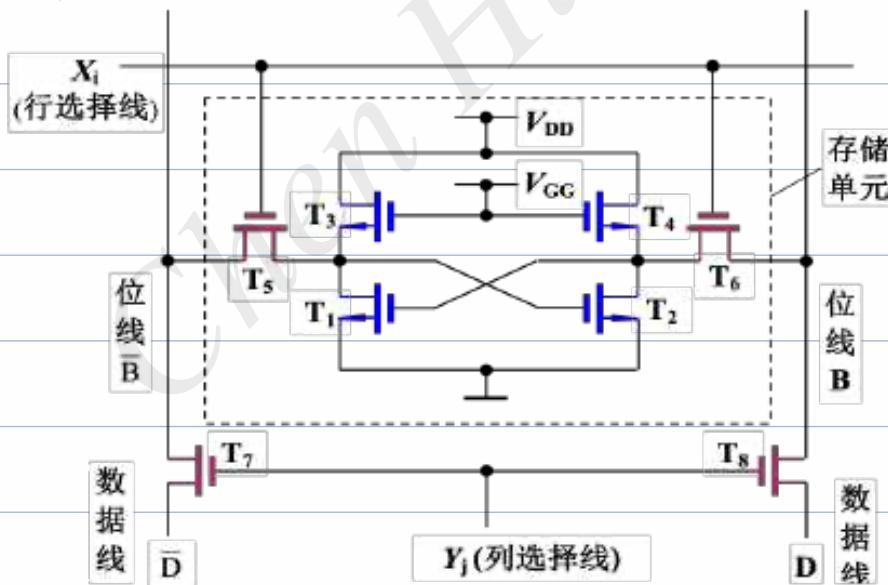
容量 $2^n \times m$

工作模式	\overline{CE}	\overline{WE}	\overline{OE}	$I/O_0 \sim I/O_{m-1}$
保持	1	X	X	高阻
读	0	1	0	数据输出
写	0	0	X	数据输入
输出无效	0	1	1	高阻

SRAM 的存储单元

只有在行、列选择线均为高电平时，才能访问存储单元

SR 锁存器



(2) DRAM

1 → 电容充电

0 → 电容放电

存储容量的扩展

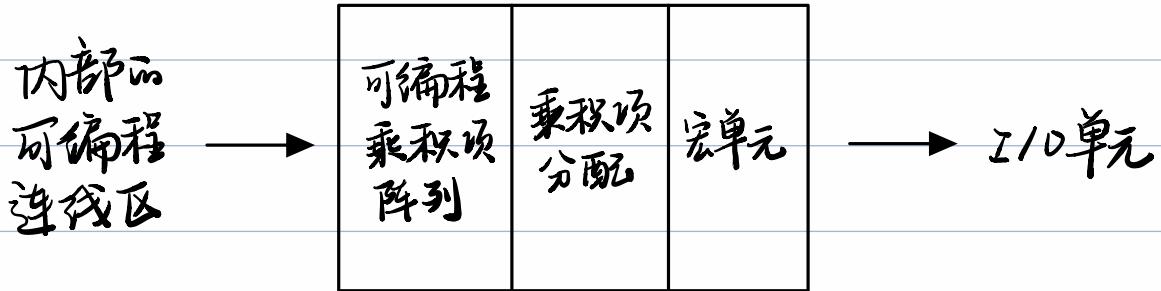
{ 位扩展：芯片并联

字数扩展：外加译码器控制存储器芯片的片选使能

输入端来实现

§8 可编程逻辑器件

1 复杂可编程逻辑器件 (CPLD)



(1) 逻辑块

基本原理：① 用“与一或”阵列和触发器可实现任何组合/时序逻辑函数。

② 编程基于 EEPROM 或 FLASH

(2) 可编程内部连线

(3) I/O 单元

2 现场可编程门阵列 (FPGA)

(1) 基本原理

用查找表 (LUT) 和触发器实现逻辑函数

编程基于 SRAM

(2) 结构简介

逻辑块 CLB

互连开关 PIC

I/O 块

§9 模数与数模转换器

模数转换器：模拟信号 \rightarrow 数字信号

A/D Analog to Digital Converer

数模转换器：数字信号 \rightarrow 模拟信号

D/A Digital to Analog Converer

① D/A转换器

(1) 输入输出特性

$$A = KD$$

(2) 基本原理

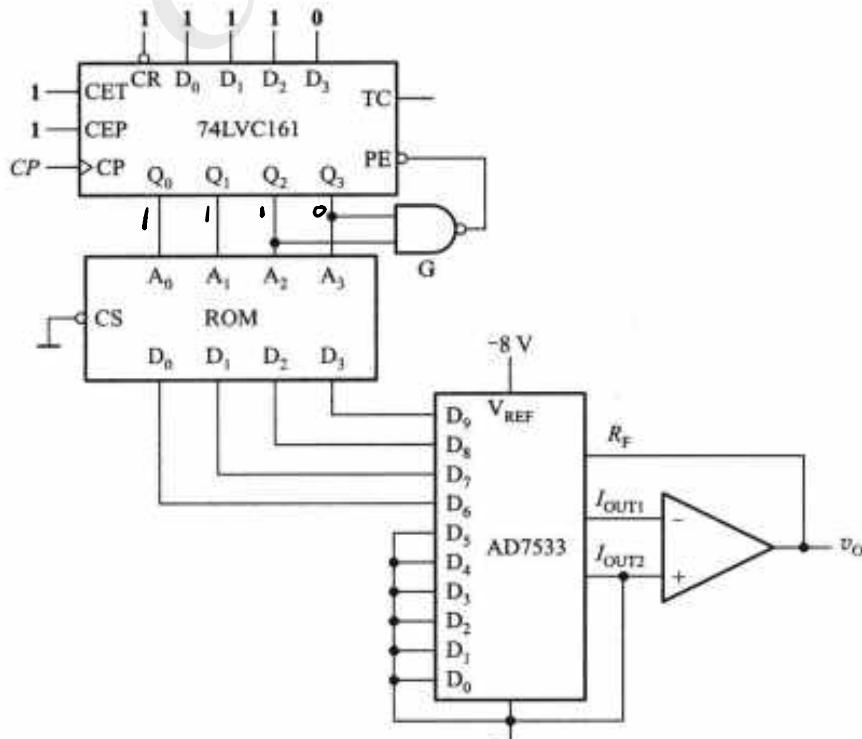
将二进制转换为十进制数

$$(N)_B = D_n \times 2^n + D_{n-1} \times 2^{n-1} + \dots + D_1 \times 2^1 + D_0 \times 2^0$$

将每一位代码按其权位大小转换成相应的模拟量，再将模拟量相加，即可得到与数字量成正比的模拟量，从而实现数字量 \rightarrow 模拟量的转换。

Example

由D/A转换器、计数器和ROM组成的波形产生电路



ROM in 数据表

A_3	A_2	A_1	A_0	D_0	D_1	D_2	D_3
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	0	1	1	1
0	1	1	0	1	0	0	0
0	1	1	1	0	1	1	1
1	0	0	0	0	0	0	1
1	0	0	1	0	0	1	0
1	0	1	0	0	0	1	1
1	0	1	1	0	1	0	0
1	1	0	0	0	0	1	1
1	1	0	1	0	1	1	0
1	1	1	0	0	1	1	1
1	1	1	1	1	0	0	0

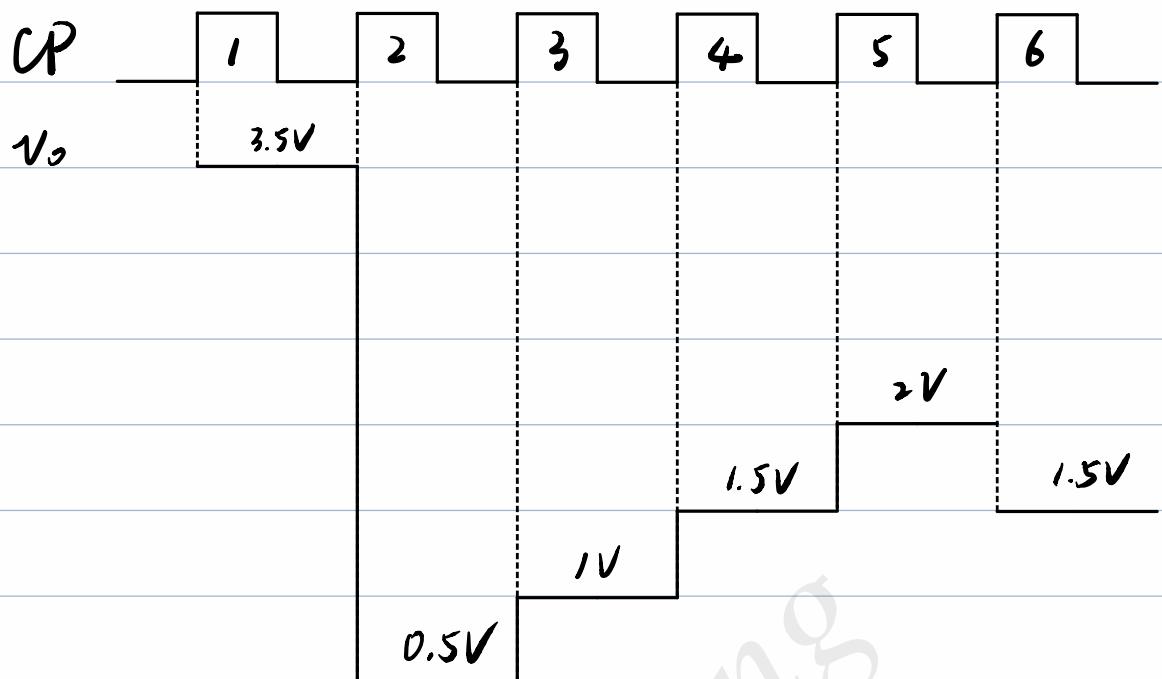
计数器状态表

$Q_3(A_3)$	$Q_2(A_2)$	$Q_1(A_1)$	$Q_0(A_0)$
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0

$$V_o = - \frac{V_{REF}}{2^{10}} \frac{R_f}{R} \sum_{i=0}^9 (D_i \cdot 2^i) \quad \text{令 } V_{REF} = -8V, R=R_f$$

D_9	D_8	D_7	D_6	V_o
0	1	1	1	3.5V
0	0	0	1	0.5V
0	0	1	0	1V
0	0	1	1	1.5V
0	1	0	0	2V
0	0	1	1	1.5V

对应 CP 波形画输出 V_o 的波形图



2 A/D 转换器

逐次比较型 A/D 转换器

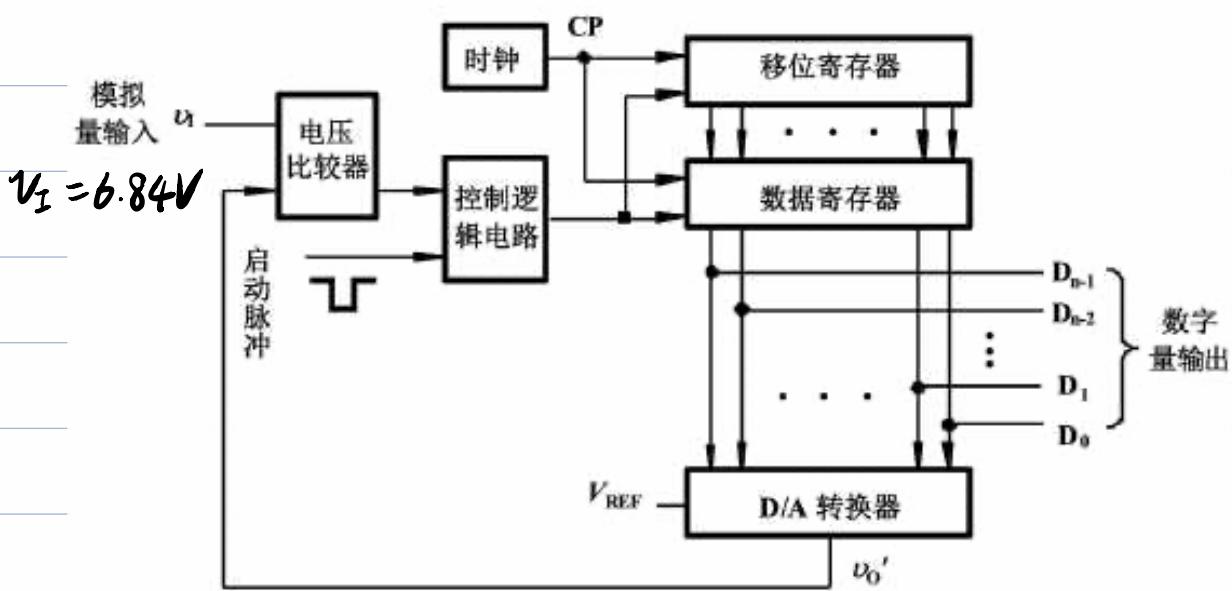
转换原理

① 取一个数字量加到 D/A 转换器上，输出一个对应的模拟电压 V_o ，与待转换的模拟量 V_I 进行比较。若 $V_o = V_I$ ，就得到了与模拟量 V_I 相对应的数字量。

② 通常从高位到低位逐位增加数字量，经过多次比较后，就能找到与模拟量 V_I 相对应的数字量。

Example

8 位逐次比较型 ADC



8位D/A转换器的输出模拟电压

$$V_O = - \frac{V_{REF}}{2^8} \frac{R_f}{R} \sum_{i=0}^7 (D_i \cdot 2^i)$$

当 $R_f = R$ 时, $V_{REF} = -10V$

① 第1个CP

$$V_O = - \frac{V_{REF}}{2^8} \cdot 2^7 = 5V$$

$V_I \geq 5V$ 保留最高位1

10000000



② 第2个CP

$$V_O = - \frac{V_{REF}}{2^8} (2^7 + 2^6) = 7.5V$$

$V_I < 7.5V$ 去掉次高位1

11000000



③ 第3个CP

$$V_O = - \frac{V_{REF}}{2^8} (2^7 + 2^5) = 6.25V$$

$V_I > 6.25V$ 保留3rd

10100000

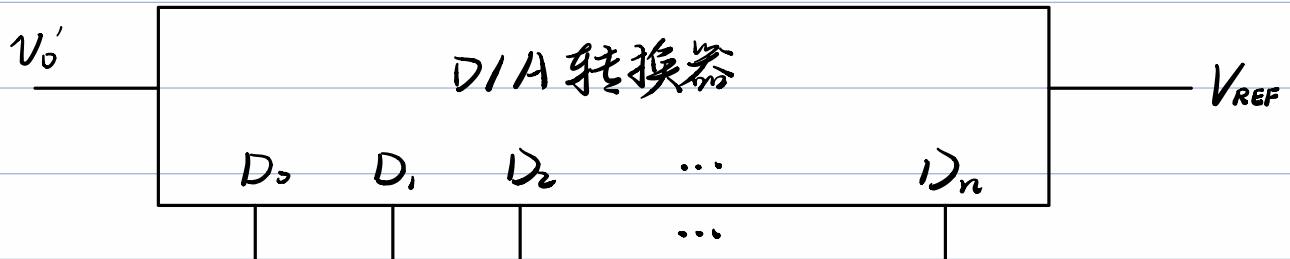


小结

① 逐次比较型A/D转换器完成一次转换所需时间与其位数n和时钟脉冲频率有关

② A/D转换器输出数字量的位数越多, 转换精度越高

逐次比较型A/D转换器中的D/A转换器



$$V_o = - \frac{V_{REF}}{2^n} \frac{R_f}{R} \sum_{i=0}^n (D_i \cdot 2^i)$$

当所有数字 $D_n \cdots D_2 D_1 D_0 = 1 \cdots 111$ 时, 得到
最大输出电压 V_{omax}