

通知 实验安排

数字逻辑电路课程有16个学时的实验，初步安排如下：

计算机1班 第8,10周 周三上午 1-4 (8:00-11:40)
计算机2班 第9,11周 周三上午 1-4 (8:00-11:40)
计算机3班 第9-12周 周二的下午5-6节
计算机4班 第10-13周 周一晚上 9-10 (19:00-20:50)
计算机5班 第9-12周 周三晚上9-10 (19:00-20:50)
物联网1班 第8,10周 周三下午5-8 (14:00-17:40)
物联网2班 第9,11周 周三下午5-8 (14:00-17:40)

计算机1班 第13, 14周 周二上午4节 (8:00-11:40)
计算机2班 第13, 14周 周二下午4节 (14:00-17:40)
计算机3班 第13, 14周 周二晚上4节 (18:30-22:10)
计算机4班 第13, 14周 周四上午4节 (8:00-11:40)
计算机5班 第13, 14周 周四下午4节 (14:00-17:40)
物联网1班 第13, 14周 周三下午4节 (14:00-17:40)
物联网2班 第13, 14周 周三晚上4节(18:30-22:10)
实验设备台套数有限，所以一次只能安排一个班实验。

实验地点：综合实验楼305房间

第四章 作业布置

本次（3月28号）作业要求：

每个同学自己从第四章的课后习题中选3题做到作业本中，至于做哪3题，每个同学自己选择。不作硬性规定。我在检查作业时，只看是否做了4题。（从你购买的课本上选题做就可以了。）

数 字 逻 辑

丁 贤 庆

ahhfdxq@163.com

4、减法运算

若 n 位二进制的原码为 $N_{\text{原}}$ ，则与它相对应的2的补码为

$$N_{\text{补}} = 2^N - N_{\text{原}}$$

补码与反码的关系式

$$N_{\text{补}} = N_{\text{反}} + 1$$

设两个数 A 、 B 相减，利用以上两式
可得

$$A - B = A + B_{\text{补}} - 2^n = A + B_{\text{反}} + 1 - 2^n$$

在实际应用中，通常是将减法运算变为加法运算来处理，即采用加补码的方法完成减法运算。

1) $A-B \geq 0$ 的情况。

2) $A-B < 0$ 的情况。

$A=0101$, $B=0010$

$$\begin{array}{rcccc} & \boxed{0} & 1 & 0 & 1 & A \\ & \boxed{1} & 1 & 0 & 1 & B_{\text{反}} \\ + & & & & 1 & \\ \hline 1 & \boxed{0} & 0 & 1 & 1 & \end{array}$$

舍弃

当 $A-B \geq 0$ 时，舍弃的进位为1，所得结果就是差的原码，不需再求反补。

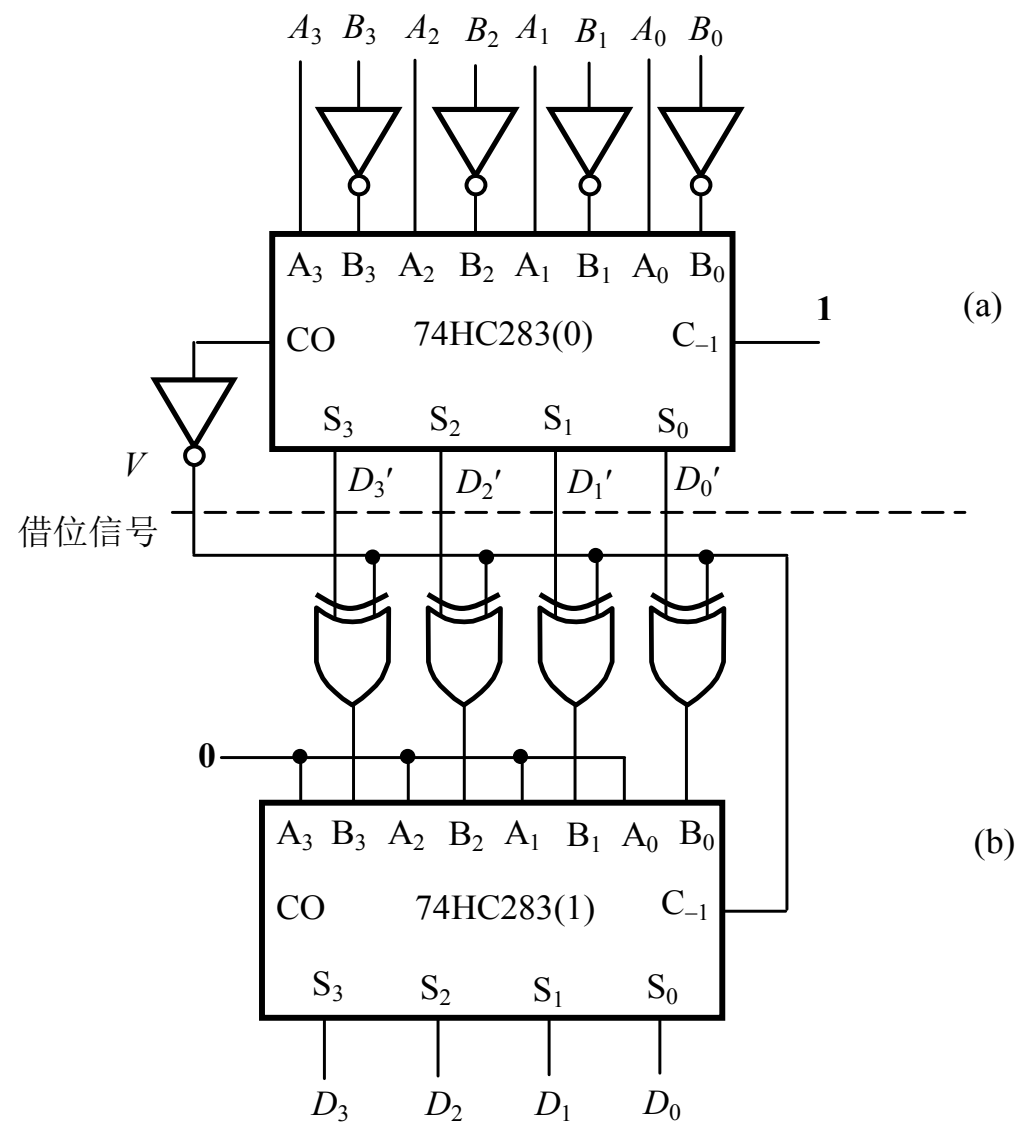
$A=0001$, $B=0101$

$$\begin{array}{rcccc} & \boxed{0} & 0 & 1 & 0 & A \\ & \boxed{1} & 0 & 1 & 0 & B_{\text{反}} \\ + & & & & 1 & \\ \hline 0 & \boxed{1} & 1 & 0 & 1 & \end{array}$$

舍弃

当 $A-B < 0$ 时，舍弃的进位为0，所得结果是补码，要得到原码需再求补。

输出为原码的4位减法运算逻辑图



4.5 组合可编程逻辑器件

4.5.1 PLD的结构、表示方法及分类

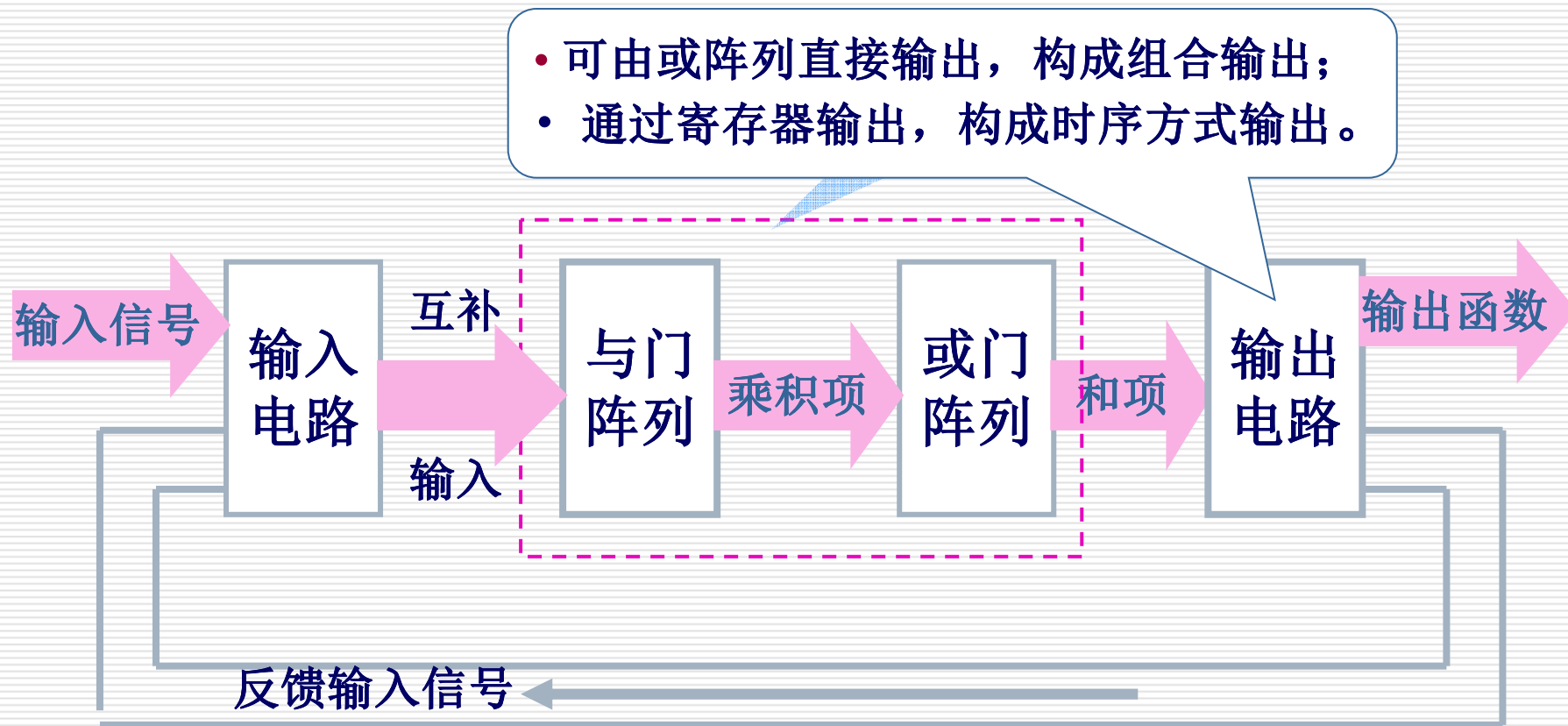
4.5.2 组合逻辑电路的PLD实现

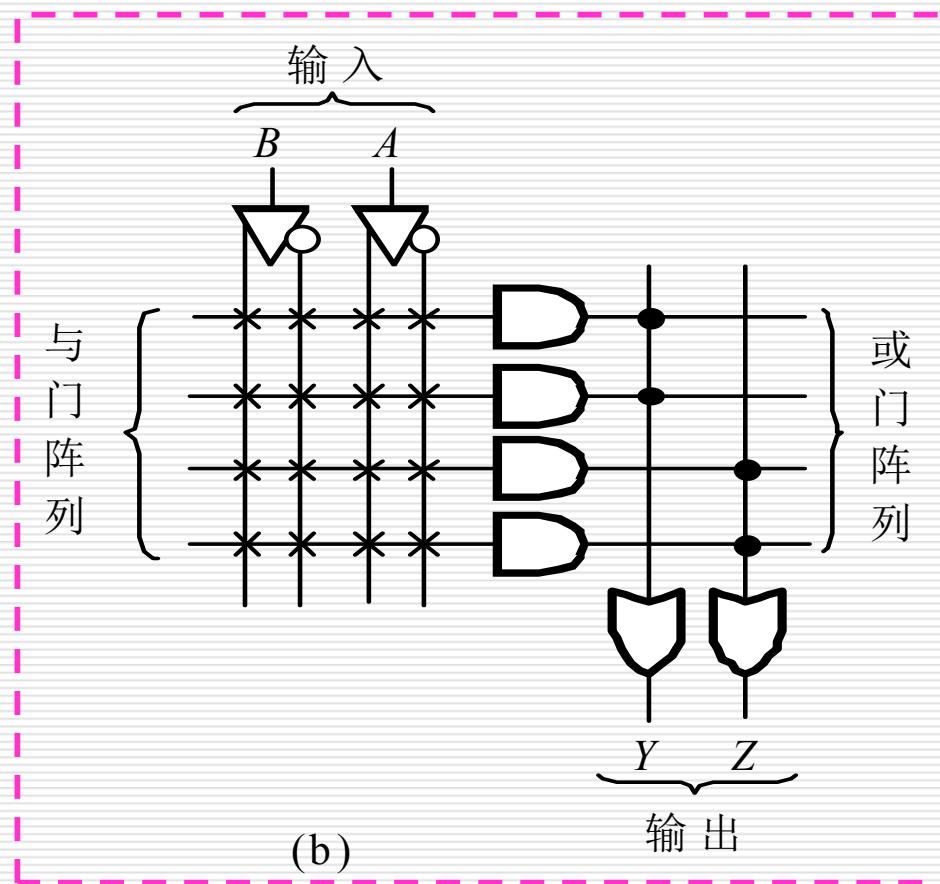
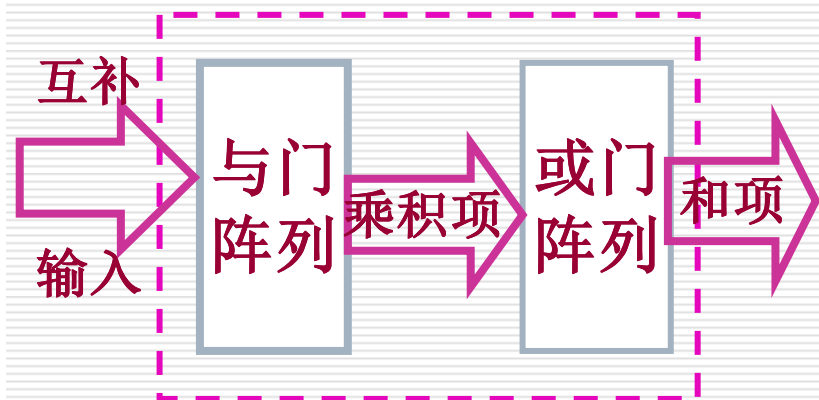
4.5 组合可编程逻辑器件

可编程逻辑器件是一种可以由用户定义和设置逻辑功能的器件。该类器件具有逻辑功能实现灵活、集成度高、处理速度快和可靠性高等特点。

4.5.1 PLD的结构、表示方法及分类

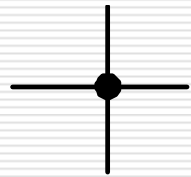
1、PLD的基本结构



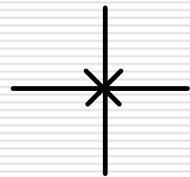


2. PLD的逻辑符号表示方法

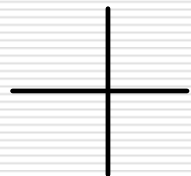
(1) 连接的方式



硬线连接单元



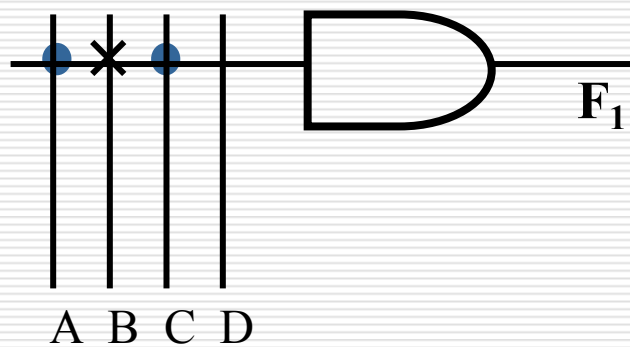
被编程接通单



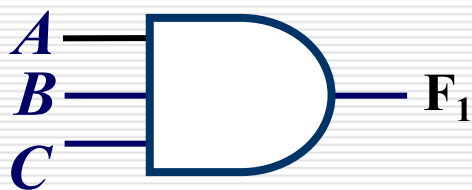
被编程擦除单元

(2) 基本门电路的表示方式

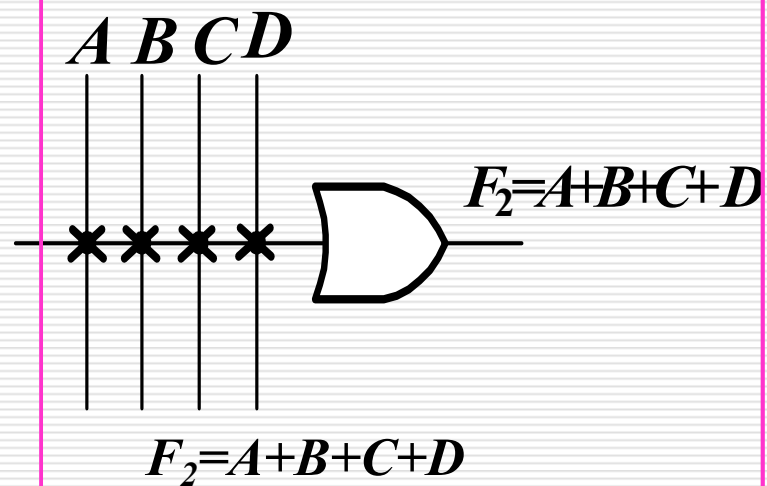
与门



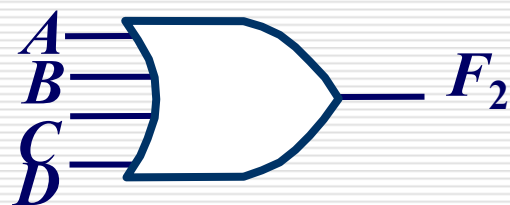
$$F_1 = A \cdot B \cdot C$$

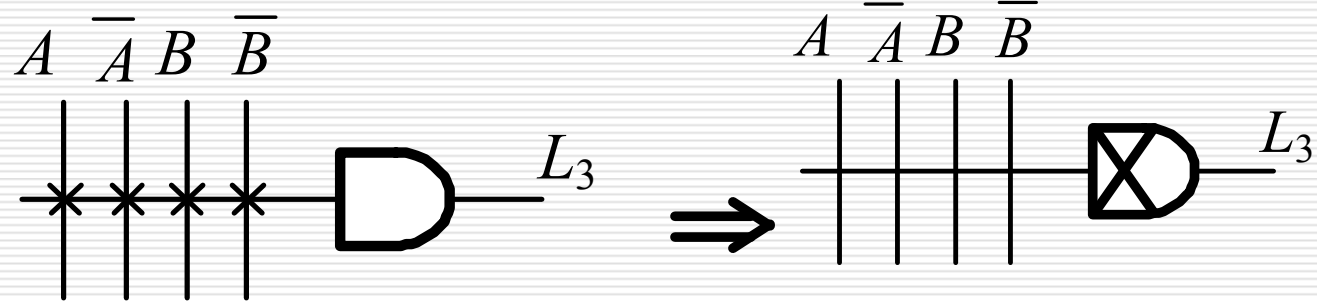


或门

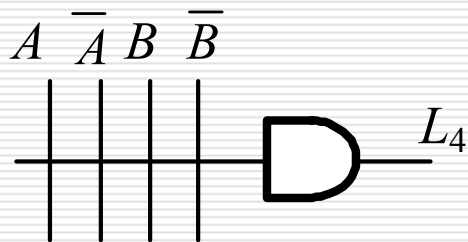


$$F_2 = A + B + C + D$$

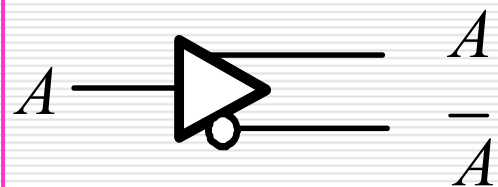




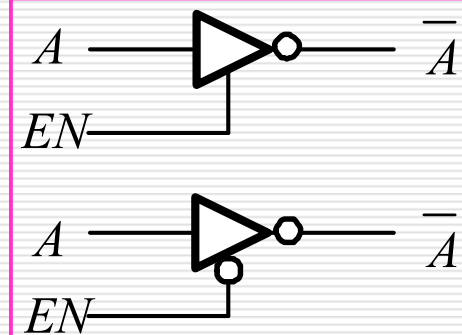
输出恒等于0的与门



输出为1的与门



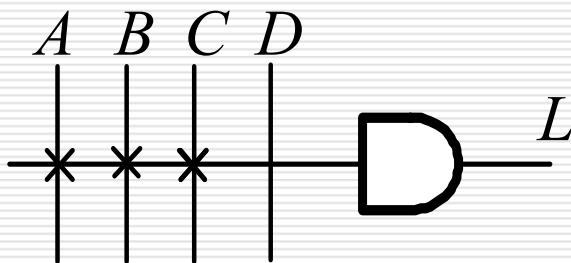
输入缓冲器



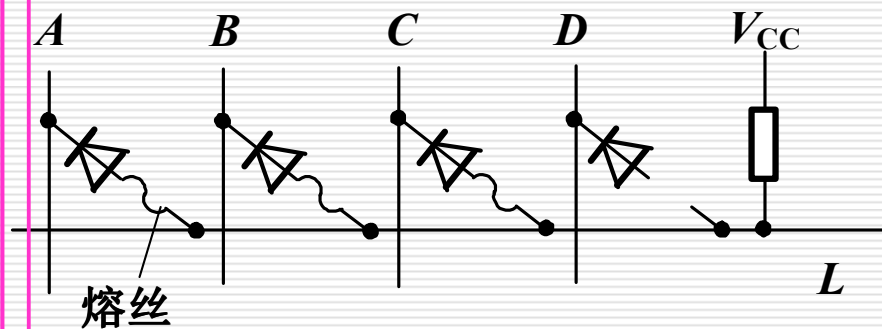
三态输出缓冲器

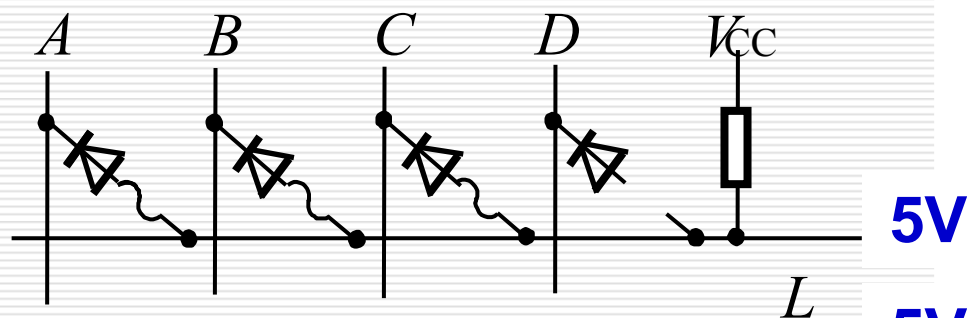
(3) 编程连接技术

PLD表示的与门



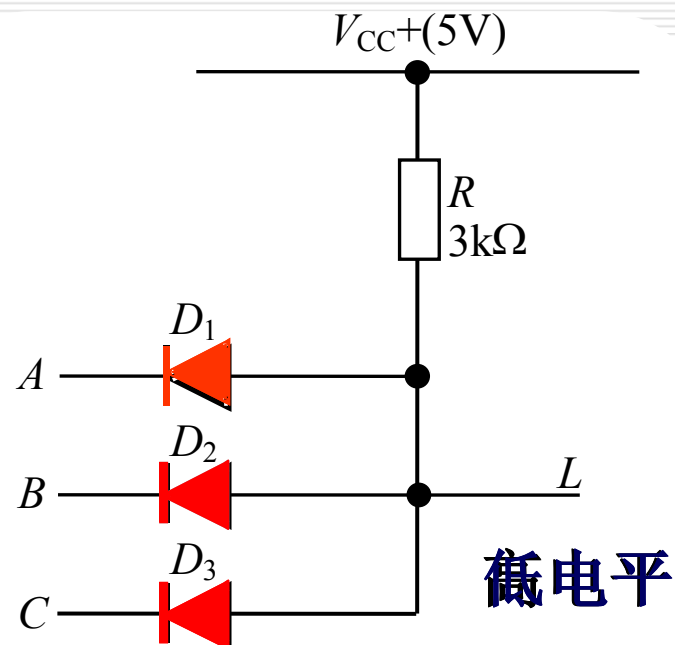
熔丝工艺的与门原理图



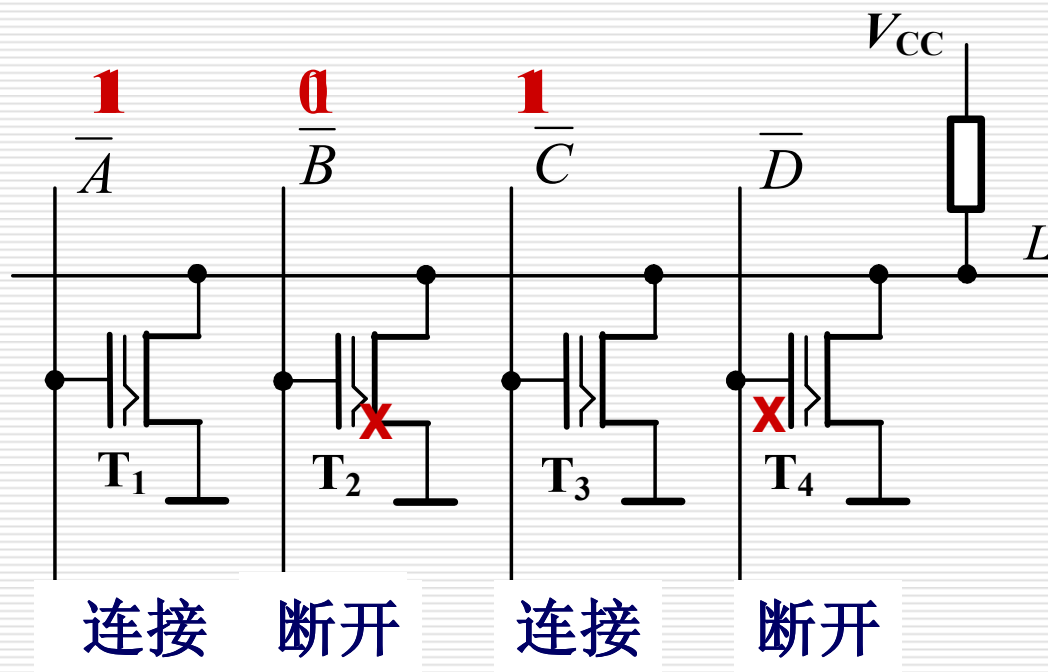


A 、 B 、 C 有一个输入低电平0V

A 、 B 、 C 三个都输入高电平+5V



$$L=A \cdot B \cdot C$$



A 、 B 、 C 中有一个为0
输出为0;

A 、 B 、 C 都为1
输出为1。

$$L=ABC$$

$$L=AC$$

器件的开关状态不同, 电路实现逻辑函数也就不同

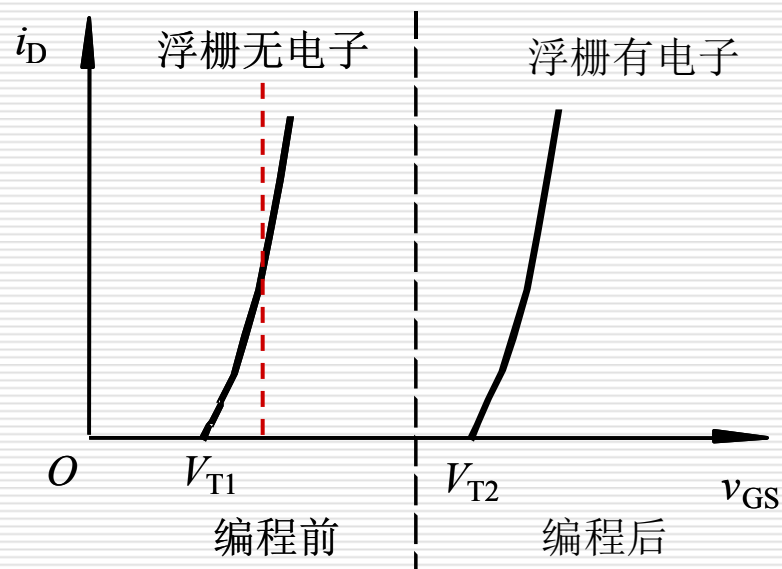
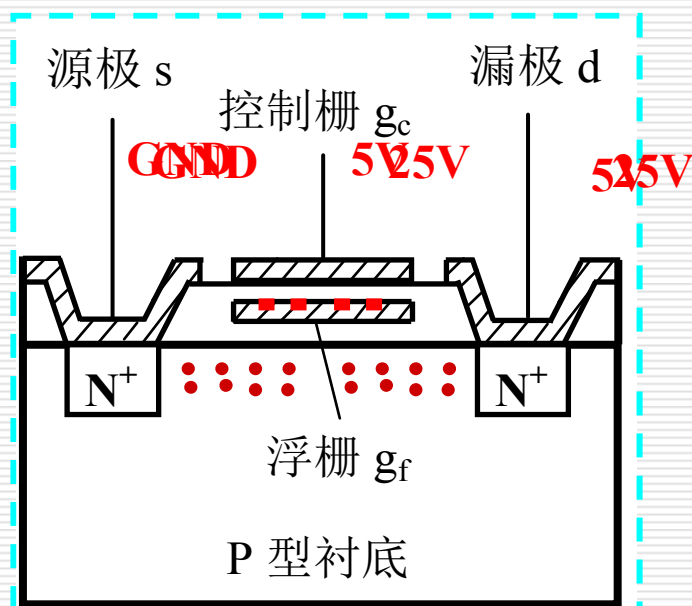
(4) 浮栅MOS管开关



用不同的浮栅MOS管连接的PLD，编程信息的擦除方法也不同。SIMOS管连接的PLD，采用紫外光照射擦除；Flotox MOS管和快闪叠栅MOS管，采用电擦除方法。

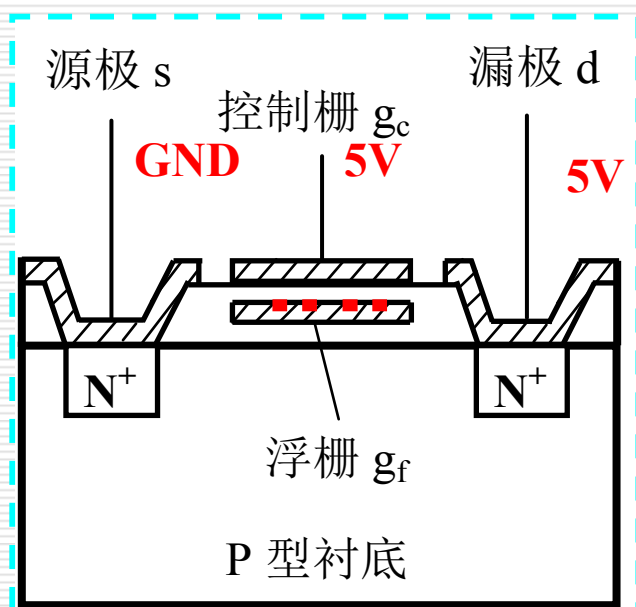
a. 叠栅注入MOS(SIMOS)管

当浮栅上没有电荷时，给控制栅加上大于 V_{T1} 的控制电压，MOS管导通。

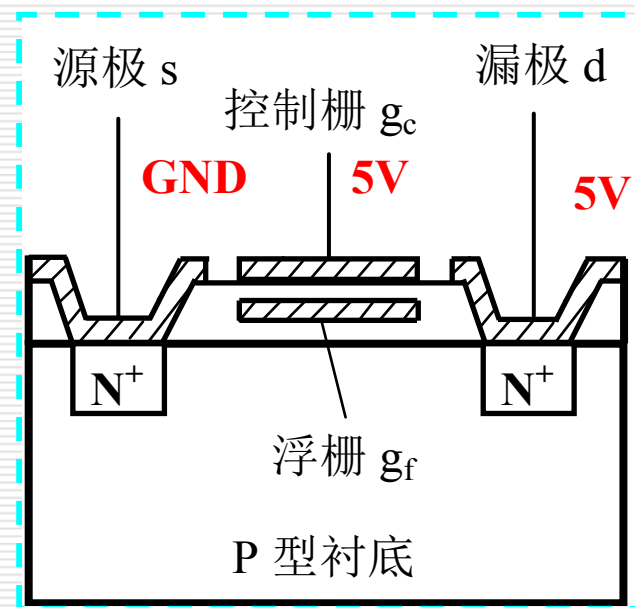


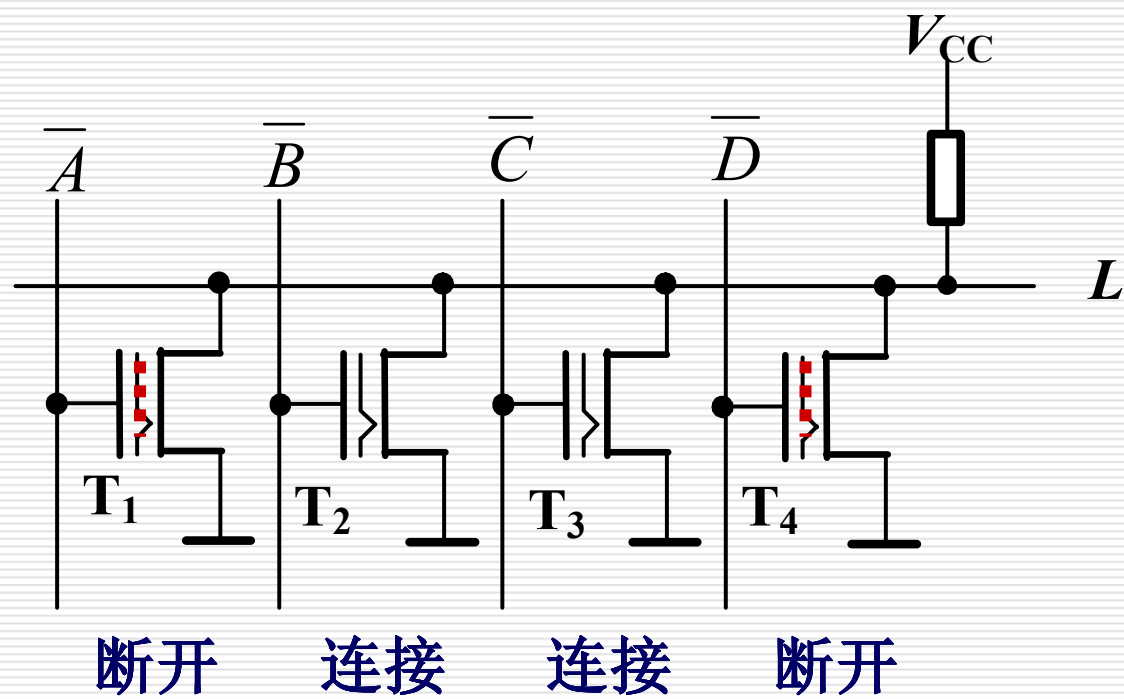
若要擦除，可用紫外线或X射线，距管子2厘米处照射15-20分钟。

截止



导通





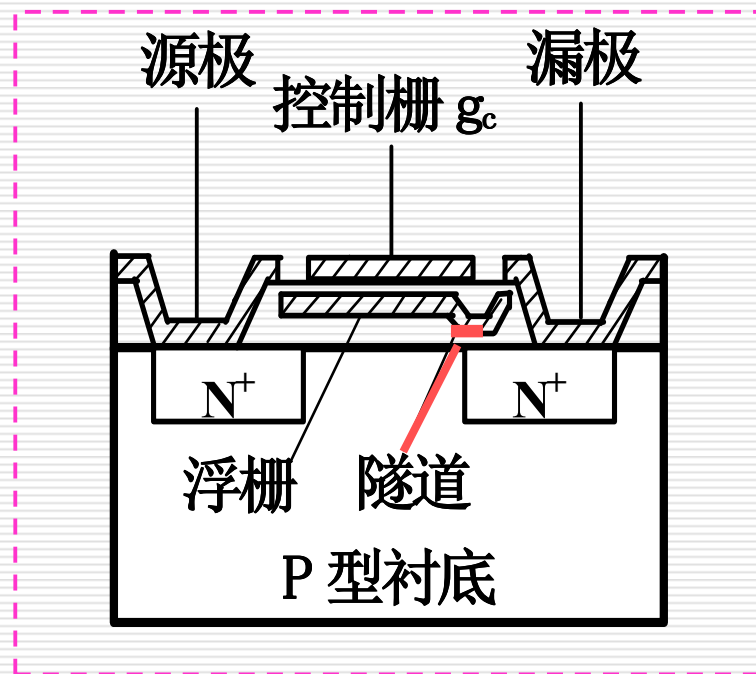
$$L=B \cdot C$$

b.浮栅隧道氧化层MOS(Flotox MOS)管

浮栅延长区与漏区 N^+ 之间的交叠处有一个厚度约为 80\AA (埃)的薄绝缘层——隧道区。

当隧道区的电场强度大到一定程度，使漏区与浮栅间出现导电隧道，形成电流将浮栅电荷泄放掉。

隧道MOS管是用电擦除的，擦除速度快。

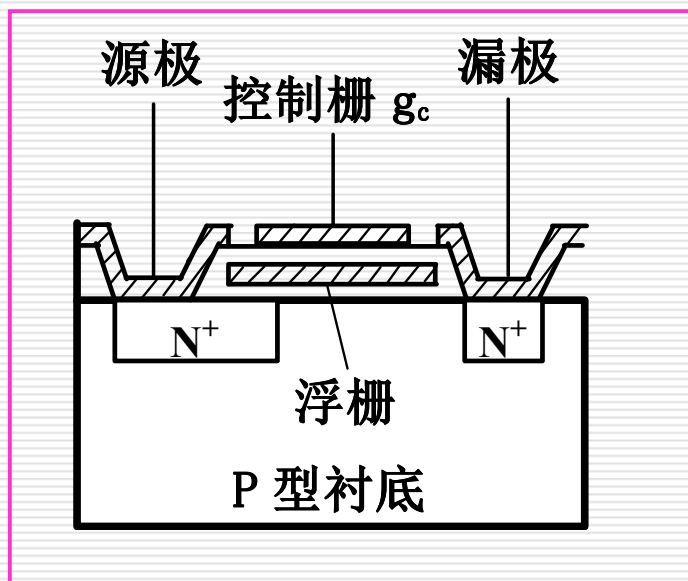


c.快闪叠栅MOS管开关 (Flash Memory) (自学)

结构特点:

1. 闪速存储器存储单元MOS管的源极N+区大于漏极N+区，而SIMOS管的源极N+区和漏极N+区是对称的；

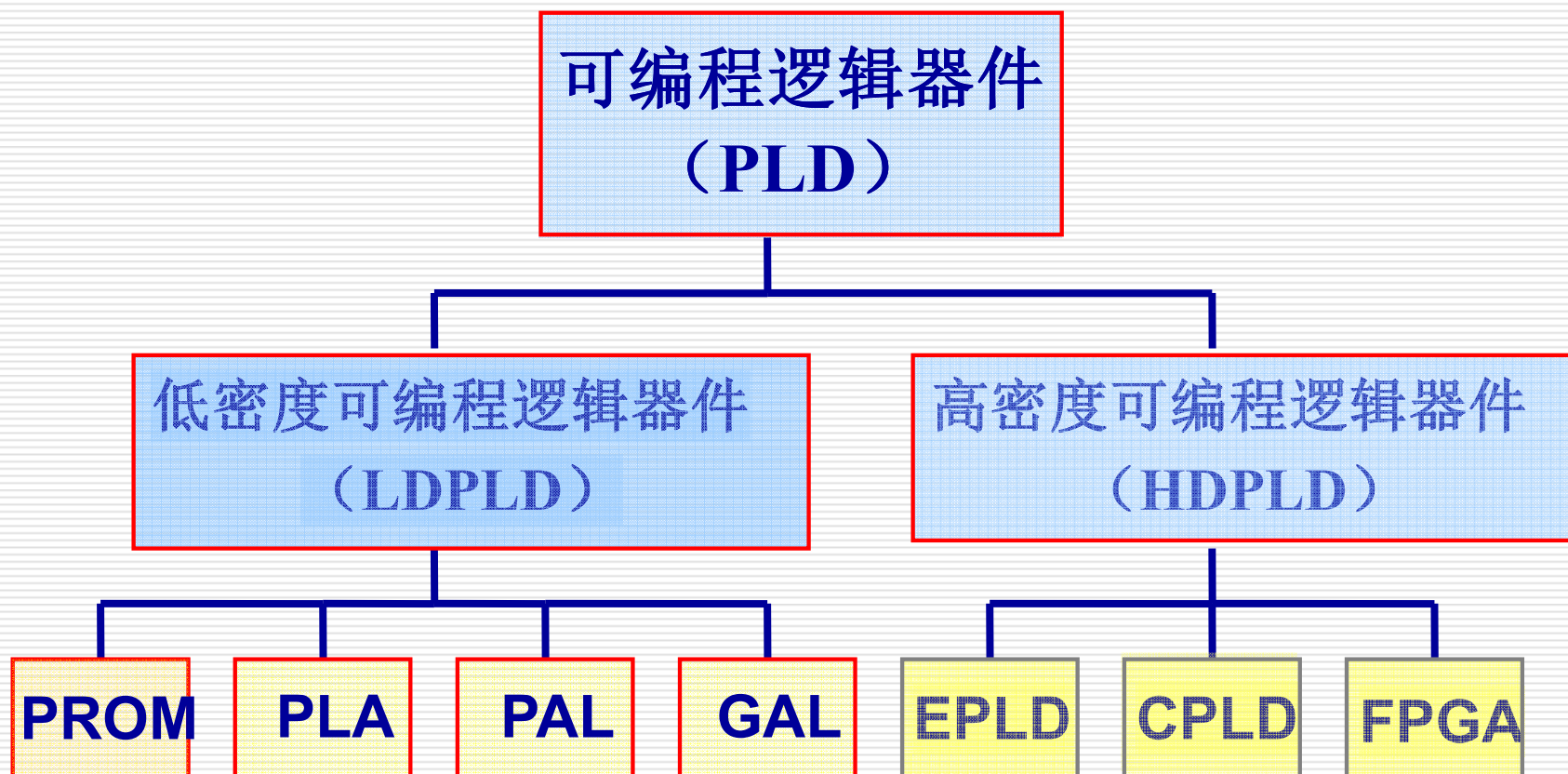
2. 浮栅到P型衬底间的氧化绝缘层比SIMOS管的更薄。



特点：结构简单、集成度高、
编程可靠、擦除快捷。

3.PLD的分类

(1) 按集成密度划分为



(2) 按结构特点划分

- 简单PLD (PAL, GAL)

- 复杂的可编程器件(CPLD) :

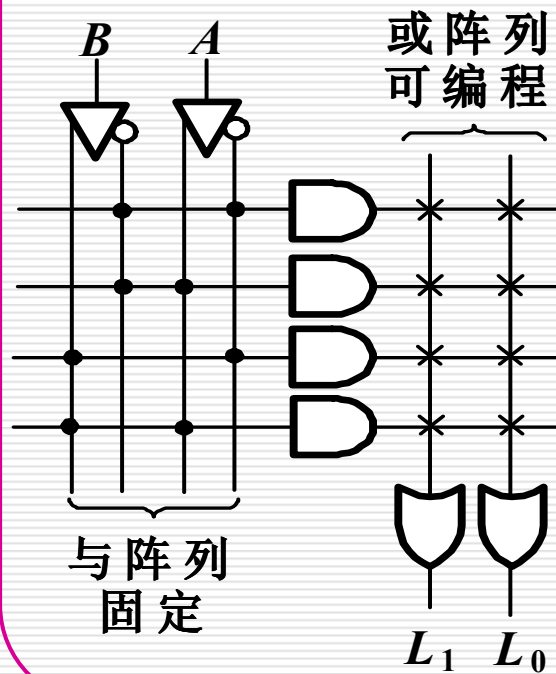
CPLD的代表芯片如: Altera的MAX系列

- 现场可编程门阵列(FPGA)

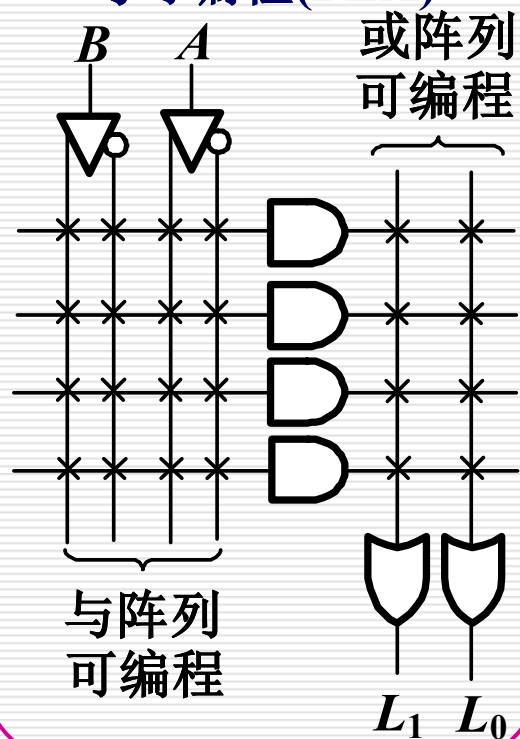
(3) 按PLD中的与、或阵列是否编程分

PLD中的三种与、或阵列

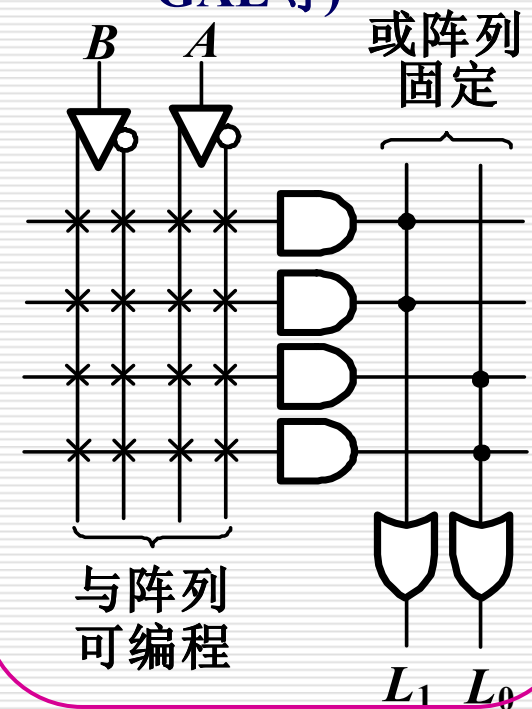
与阵列固定，或阵列可编程(PROM)



与阵列、或阵列均可编程(PLA)



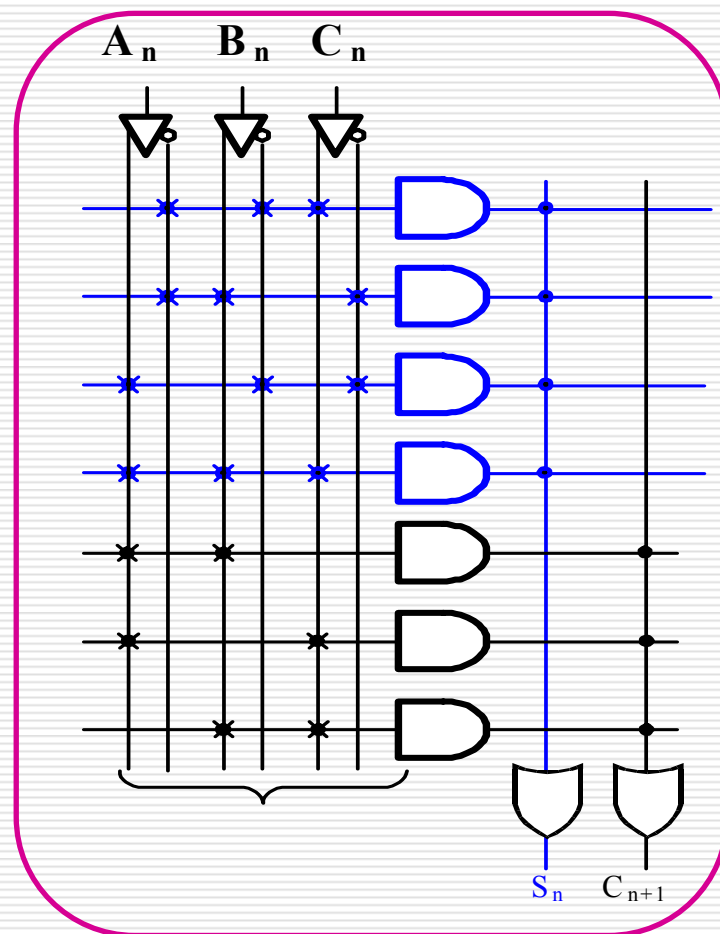
与阵列可编程，或阵列固定(PAL和GAL等)



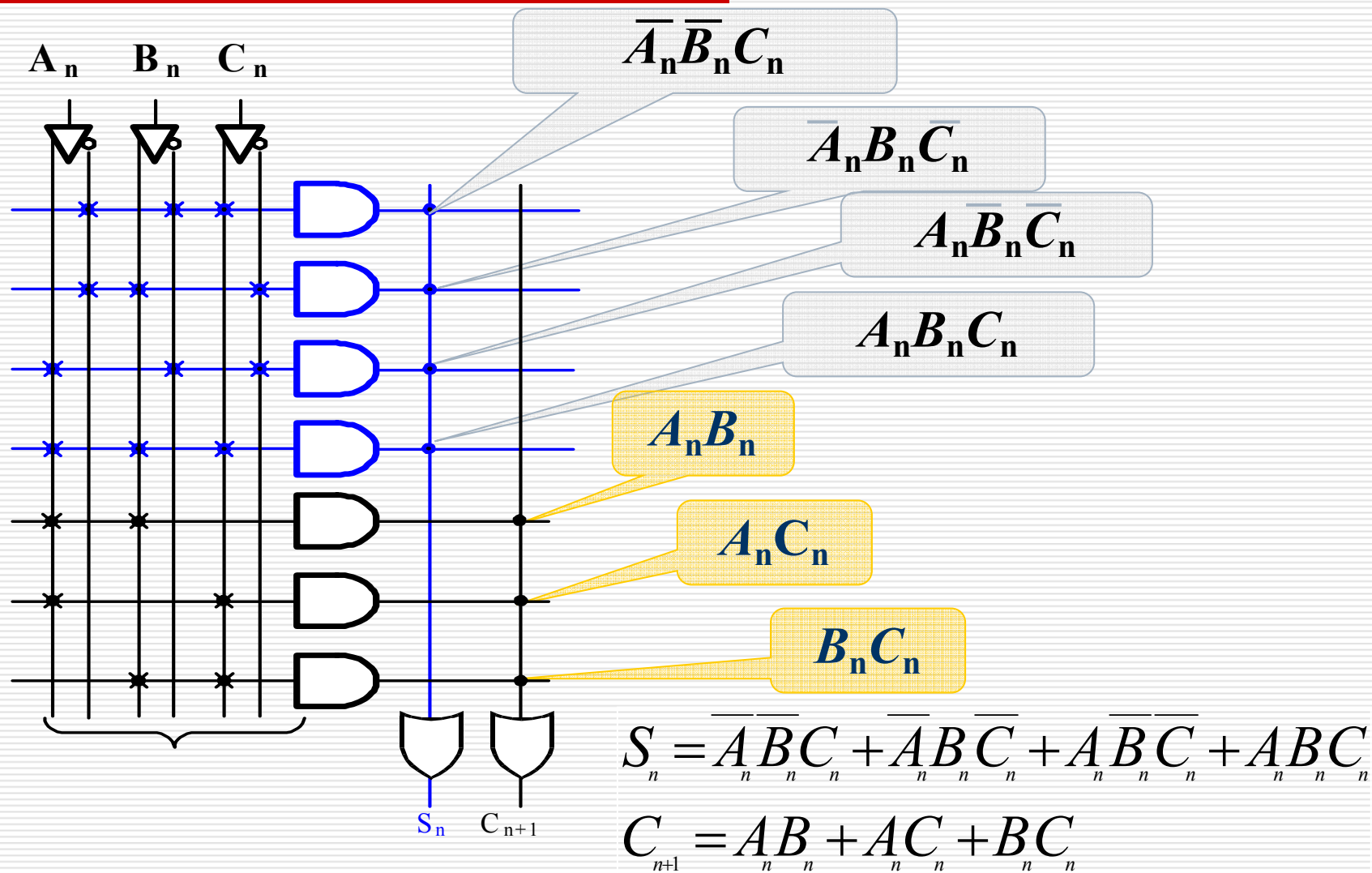
4.5.2 组合逻辑电路的 PLD 实现

例1 由PLA构成的逻辑电路如图所示，试写出该电路的逻辑表达式，并确定其逻辑功能。

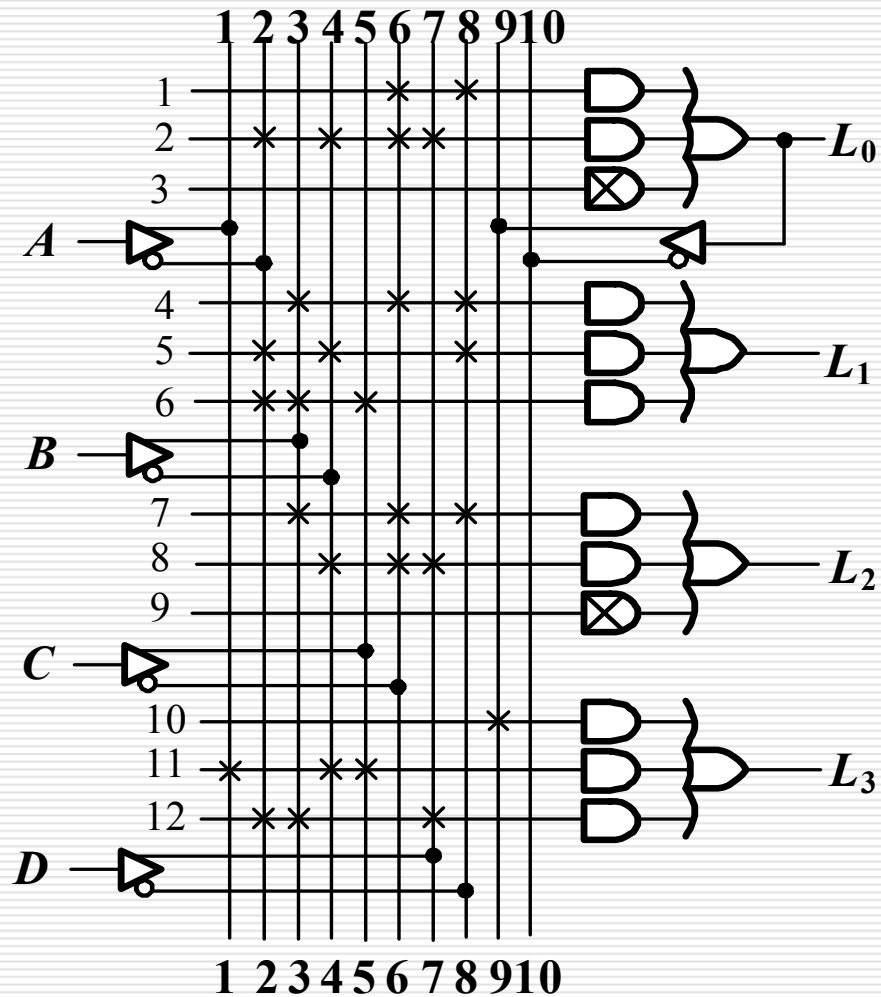
写出该电路的逻辑表达式：



全加器



试写出该电路的逻辑表达式。



$$L_0 = \overline{\overline{C}}\overline{\overline{D}} + \overline{\overline{A}}\overline{\overline{B}}\overline{\overline{C}}\overline{\overline{D}}$$

$$L_1 = \overline{\overline{B}}\overline{\overline{C}}\overline{\overline{D}} + \overline{\overline{A}}\overline{\overline{B}}\overline{\overline{D}} + \overline{\overline{A}}\overline{\overline{B}}\overline{\overline{C}}$$

$$L_2 = \overline{\overline{B}}\overline{\overline{C}}\overline{\overline{D}} + \overline{\overline{B}}\overline{\overline{C}}\overline{\overline{D}}$$

$$L_3 = L_0 + \overline{\overline{A}}\overline{\overline{B}}\overline{\overline{C}} + \overline{\overline{A}}\overline{\overline{B}}\overline{\overline{D}}$$

4.6 用VerilogHDL描述组合逻辑电路

4.6.1 组合逻辑电路的行为级建模

4.6.2 分模块、分层次的电路设计

4.6.1 组合逻辑电路的行为级建模

组合逻辑电路的行为级描述一般使用**assign**结构和过程赋值语句、条件语句（**if-else**）、多路分支语句（**case-endcase**）和**for**循环语句等。

1、条件语句（if语句）

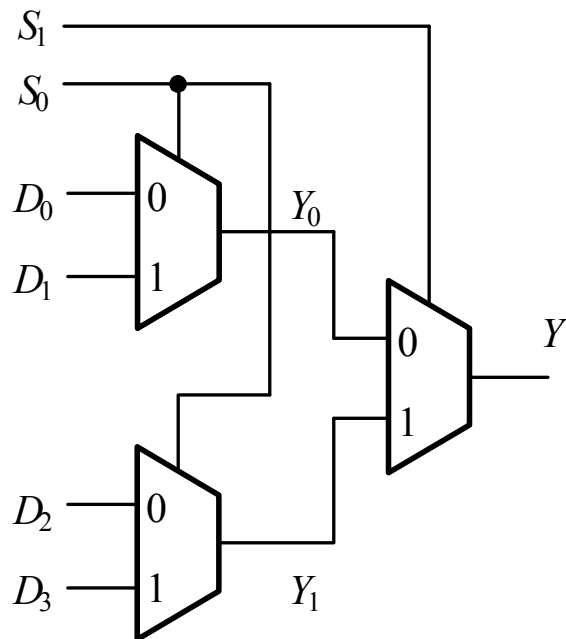
条件语句就是根据判断条件是否成立，确定下一步的运算。

Verilog语言中有3种形式的if语句：

- (1) `if (condition_expr) true_statement;`
- (2) `if (condition_expr) true_statement;`
`else false_statement;`
- (3) `if (condition_expr1) true_statement1;`
`else if (condition_expr2) true_statement2;`
`else if (condition_expr3) true_statement3;`
`.....`
`else default_statement;`

if后面的条件表达式一般为逻辑表达式或关系表达式。执行if语句时，首先计算表达式的值，若结果为0、x或z，按“假”处理；若结果为1，按“真”处理，并执行相应的语句。

例：使用if-else语句对4选1数据选择器的行为进行描述



```
module mux4to1_bh(D, S, Y);  
    input [3:0] D; //输入端口  
    input [1:0] S; //输入端口  
    output reg Y; //输出端口及变量数据类型  
    always @(D, S) //电路功能描述  
        if (S == 2'b00)    Y = D[0];  
        else if (S == 2'b01) Y = D[1];  
        else if (S == 2'b10) Y = D[2];  
        else                Y = D[3];  
endmodule
```

注意，过程赋值语句只能给寄存器型变量赋值，因此，输出变量Y的数据类型定义为**reg**。

2、多路分支语句（**case语句**）

是一种多分支条件选择语句，一般形式如下

```
case (case_expr)
    item_expr1: statement1;
    item_expr2: statement2;
    .....
default: default_statement; //default语句可以省略
endcase
```

注意：当分支项中的语句是多条语句，必须在最前面写上关键词begin，在最后写上关键词end，成为顺序语句块。

另外，用关键词casex和casez表示含有无关项x和高阻z的情况。

例：对具有使能端En 的4选1数据选择器的行为进行Verilog描述。
当En=0时，数据选择器工作，En=1时，禁止工作，输出为0。

```
module mux4to1_bh (D, S, Y);  
    input [3:0] D, [1:0] S;  
    output reg Y;  
    always @(D, S, En) //2001, 2005 syntax  
        begin  
            if (En==1) Y = 0; //En=1时，输出为0  
            else //En=0时，选择器工作  
                case (S)  
                    2'd0: Y = D[0];  
                    2'd1: Y = D[1];  
                    2'd2: Y = D[2];  
                    2'd3: Y = D[3];  
                endcase  
            end  
        end  
endmodule
```

例：对基本的4线-2线优先编码器的行为进行Verilog描述。

```
module priority(W, Y)
    input [3:0] W;
    output reg [1:0] Y;
    always @(W)
        casex (W)
            4'b1xxx: Y = 3;
            4'b01xx: Y = 2;
            4'b001x: Y = 1;
            4'b0001: Y = 0;
            default: begin z = 0; Y = 2'bx; end //W无效时,z=0,Y为高阻
        endcase
    endendmodule
```

3、for循环语句

一般形式如下

for (initial_assignment; condition; step_assignment) statement;

initial_assignment为循环变量的**初始值**。

Condition为**循环的条件**，若为真，执行过程赋值语句**statement**，若不成立，循环结束，执行**for**后面的语句。

step_assignment为**循环变量的步长**，每次迭代后，循环变量将增加或减少一个步长。

试用Verilog语言描述具有高电平使能的3线-8线译码器.

```
module ecoder3to8_bh(A,En,Y);
    input [2:0] A, En;
    output reg [7:0]Y;
    integer k;    //声明一个整型变量k
    always @(A, En) //
        begin
            Y = 8'b1111_1111; //设译码器输出的默认值
            for(k = 0; k <= 7; k = k+1) //下面的if-else语句循环8次
                if ((En==1) && (A== k) )
                    Y[k] = 0; //当En=1时，根据A进行译码
                else
                    Y[k] = 1; //处理使能无效或输入无效的情况
        end
    endmodule
```

循环8次 {

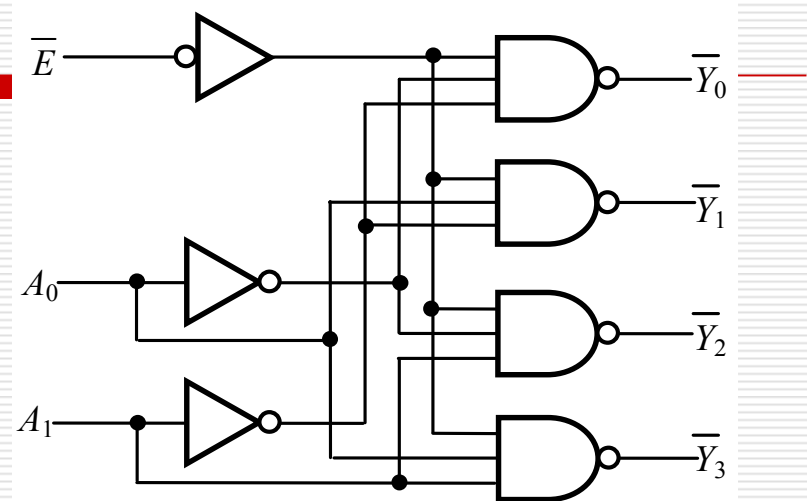
例：用**条件运算符**描述了一个2选1的数据选择器。

```
module mux2x1_df (A,B,SEL,L);  
    input A,B,SEL;  
    output L;  
    assign L = SEL ? A : B;  
    endmodule
```

在连续赋值语句中，如果SEL=1，则输出L=A；否则L=B。

```
module mux2x1_df (A,B,SEL,L);  
    input A,B,SEL;  
    output reg Y;  
    always @( D1,D0,S ) //用always语句和条件运算符建模  
        L = S ? D1 : D0;  
    endmodule
```

例：用数据流建模方法对2线-4线译码器的行为进行描述。

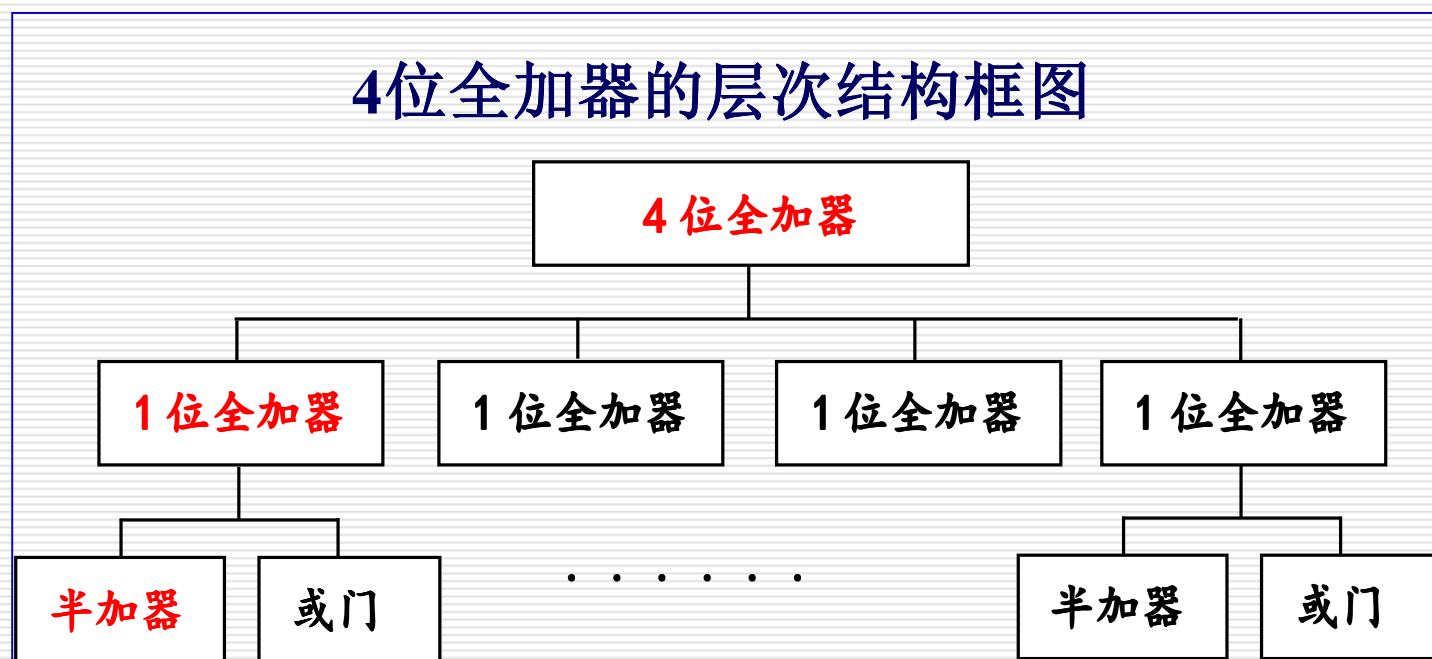


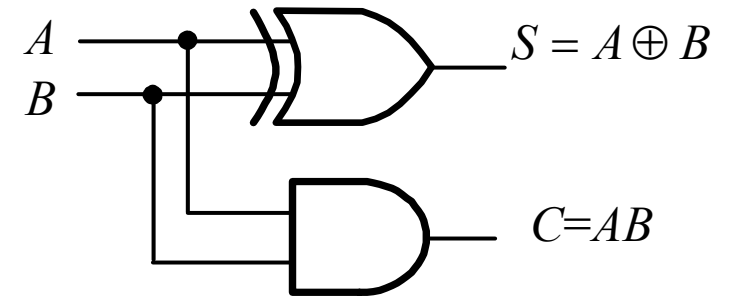
```
module decoder_df (A1,A0,E,Y);
    input A1,A0,E;
    output [3:0] Y;
    assign Y[0] = ~(~A1 & ~A0 & ~E);
    assign Y[1] = ~(~A1 & A0 & ~E);
    assign Y[2] = ~(A1 & ~A0 & ~E);
    assign Y[3] = ~(A1 & A0 & ~E);
endmodule
```

4.6.2 分模块、分层次的电路设计

分层次的电路设计:在电路设计中,将两个或多个模块组合起来描述电路逻辑功能的设计方法。

设计方法: 自顶向下和自底向上两种常用的设计方法





```
module halfadder (S,C,A,B);  
    input A,B;  
    output S,C;  
    xor (S,A,B);  
    and (C,A,B);  
endmodule
```

半加器的门级描述

```
module fulladder (S,CO,A,B,CI);
```

```
    input A,B,CI;
```

```
    output S,CO;
```

```
    wire S1,D1,D2; //内部节点信号
```

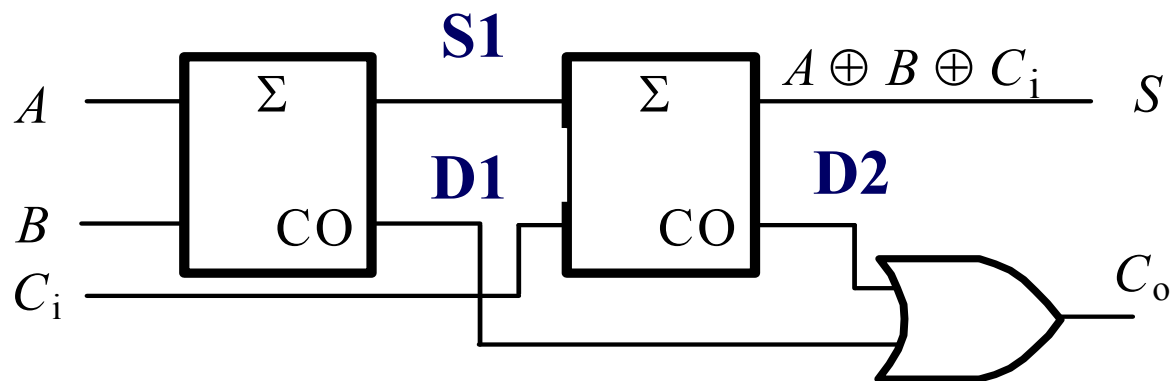
```
    halfadder HA1 (S1,D1,A,B);
```

```
    halfadder HA2 (S,D2,S1,CI);
```

```
    or g1(CO,D2,D1);
```

```
endmodule
```

全加器的描述-调用半加器



```
module _4bit_adder (S,C3,A,B,C_1);
```

```
    input [3:0] A,B;
```

```
    input C_1;
```

```
    output [3:0] S;
```

```
    output C3;
```

```
    wire C0,C1,C2; //内部进位信号
```

```
    fulladder FA0 (S[0],C0,A[0],B[0],C_1),
```

```
    FA1 (S[1],C1,A[1],B[1],C0),
```

```
    FA2 (S[2],C2,A[2],B[2],C1),
```

```
    FA3 (S[3],C3,A[3],B[3],C2);
```

```
    endmodule
```

4位全加器的描述
--调用1位全加器

第五章

锁存器和触发器

时序逻辑电路概述

1、时序逻辑电路与锁存器、触发器：

时序逻辑电路：

工作特征：时序逻辑电路的工作特点是任意时刻的输出状态不仅与该当前的输入信号有关，而且与此前电路的状态有关。

结构特征：由组合逻辑电路和存储电路组成, 电路中存在反馈。

锁存器和触发器是构成时序逻辑电路的基本逻辑单元 。

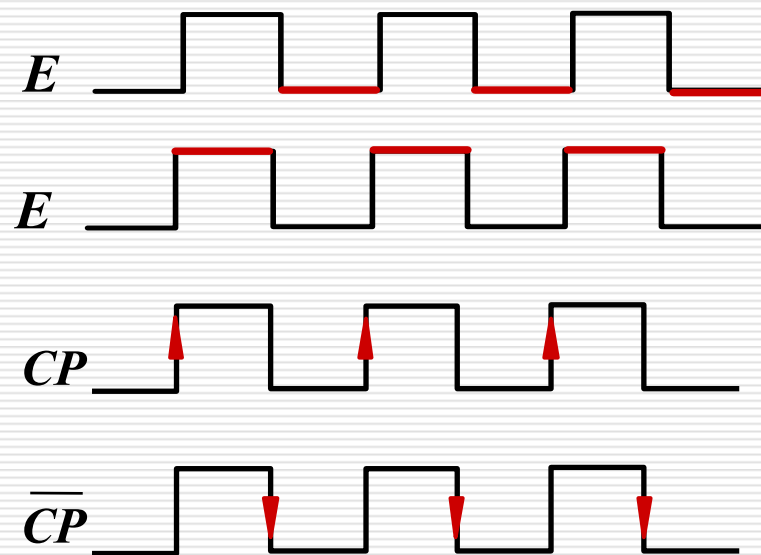
2、锁存器与触发器

共同点：具有0和1两个稳定状态，一旦状态被确定，就能自行保持。一个锁存器或触发器能存储一位二进制码。

不同点：

锁存器---对脉冲电平敏感的存储电路，在特定输入脉冲电平作用下改变状态。

触发器---对脉冲边沿敏感的存储电路，在时钟脉冲的上升沿或下降沿的变化瞬间改变状态。



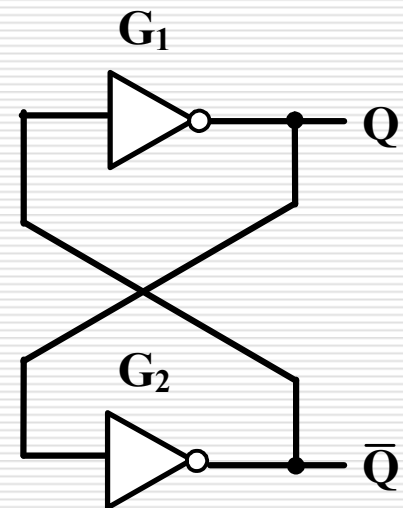
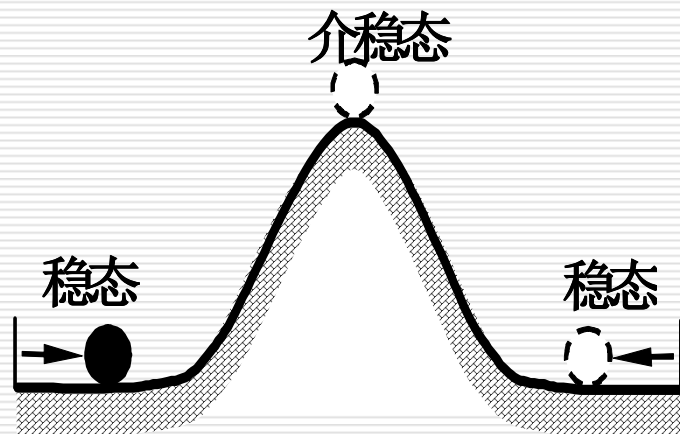
5.1 双稳态电路

5.1.1 双稳态的概念

5.1.2 最基本的双稳态电路

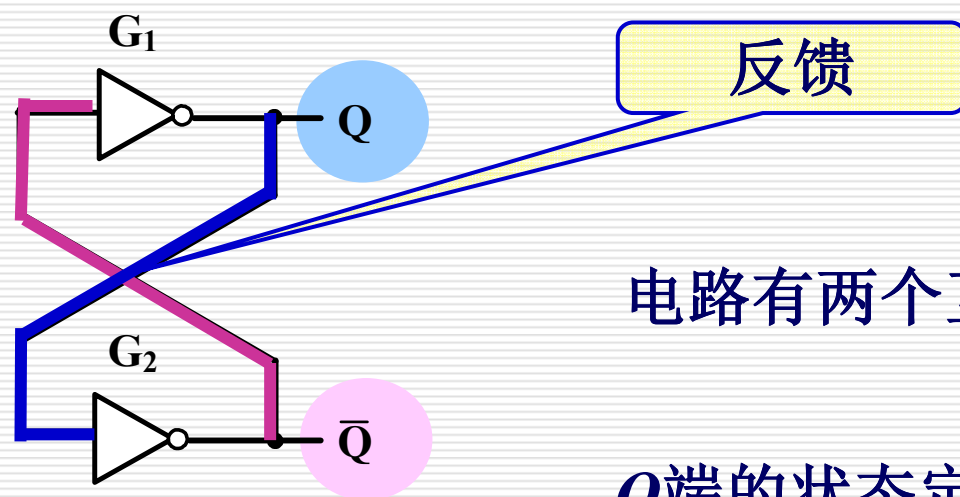
5.1 双稳态电路

5.1.1 双稳态的概念



5.1.2 最基本的双稳态电路

1. 电路结构



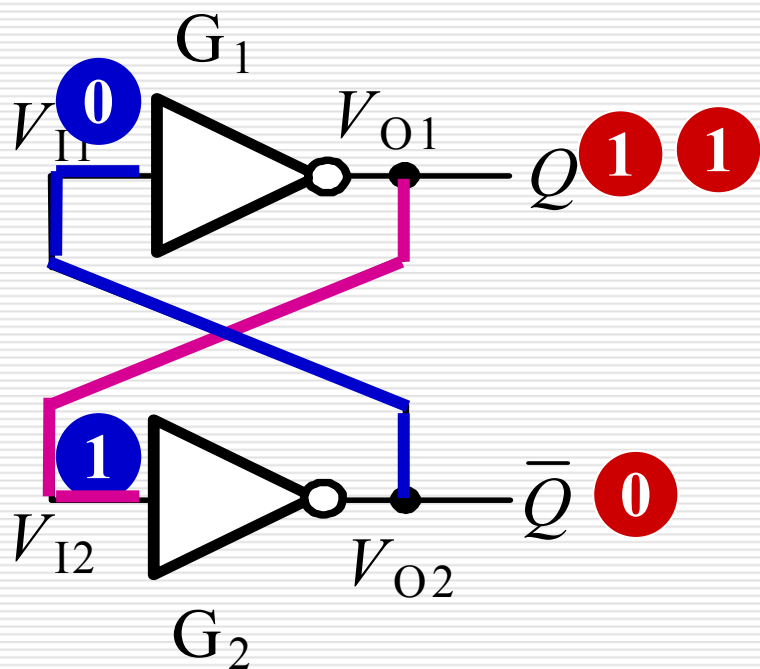
电路有两个互补的输出端

Q 端的状态定义为电路输出状态。

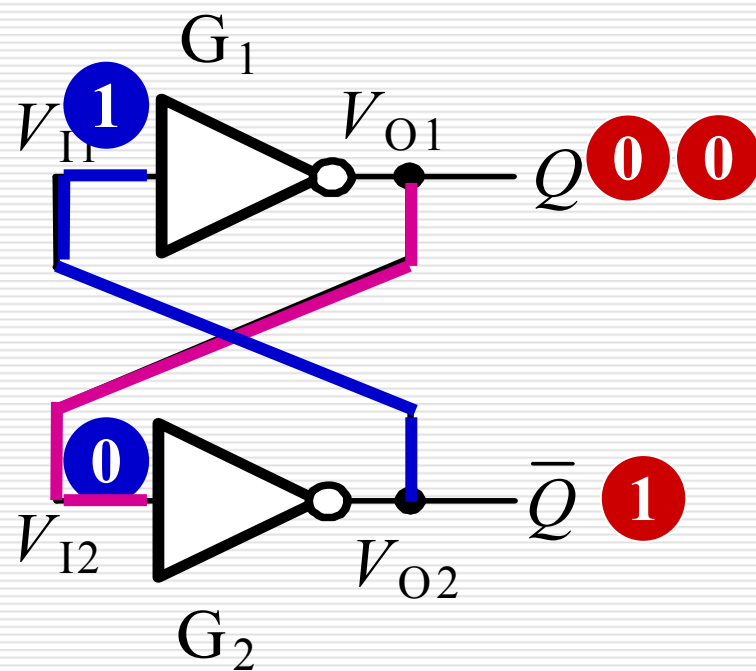
2、数字逻辑分析

——电路具有记忆1位二进制数据的功能。

如 $Q = 1$

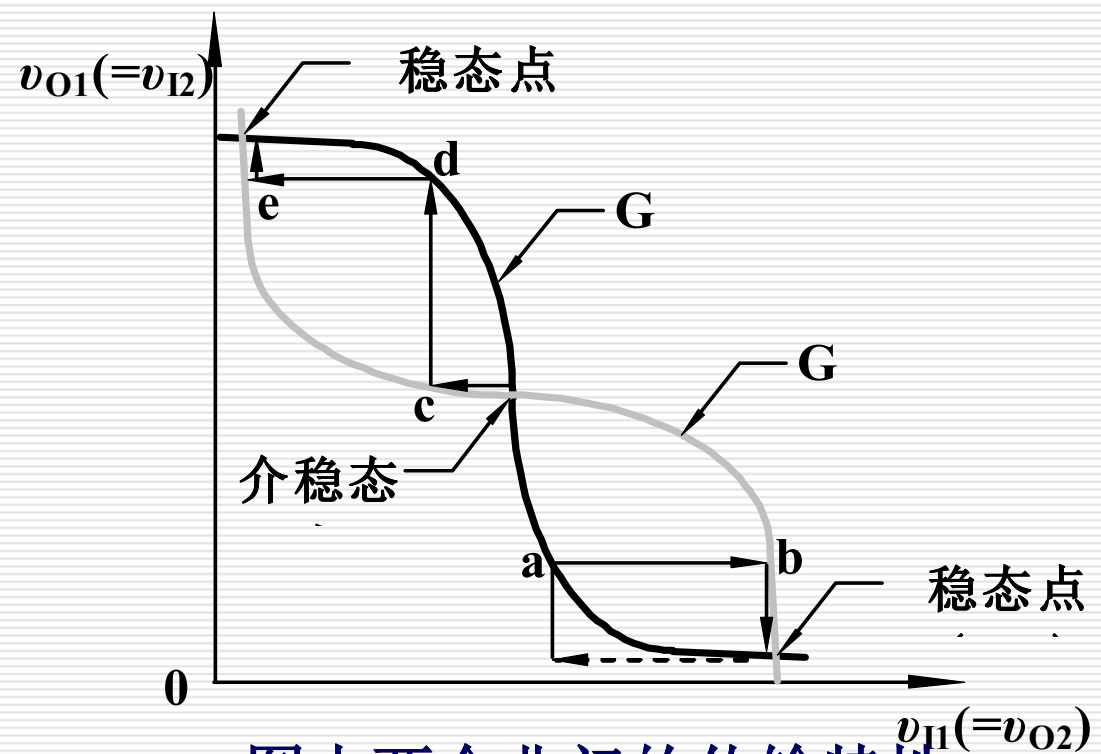
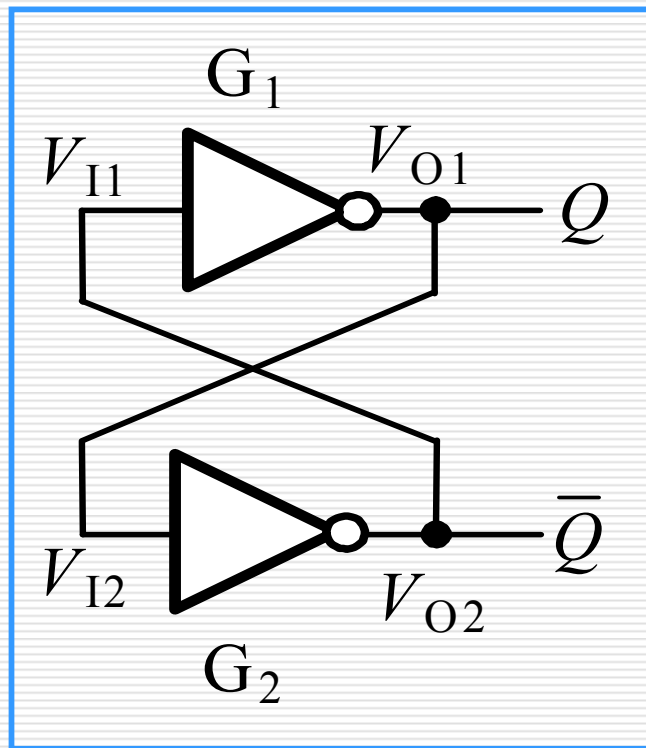


如 $Q = 0$



3. 模拟特性分析

$$v_{O1} = v_{I2} \quad v_{I1} = v_{O2}$$



图中两个非门的传输特性

5.2 SR锁存器

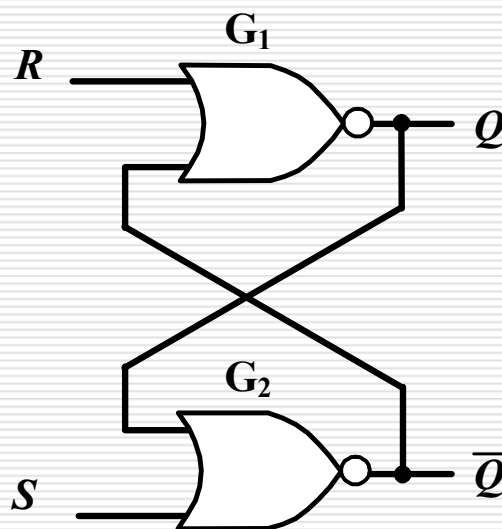
5.2.1 基本SR 锁存器

5.2.2 门控SR锁存器

5.2 SR锁存器

5.2.1 基本SR 锁存器

1. 工作原理

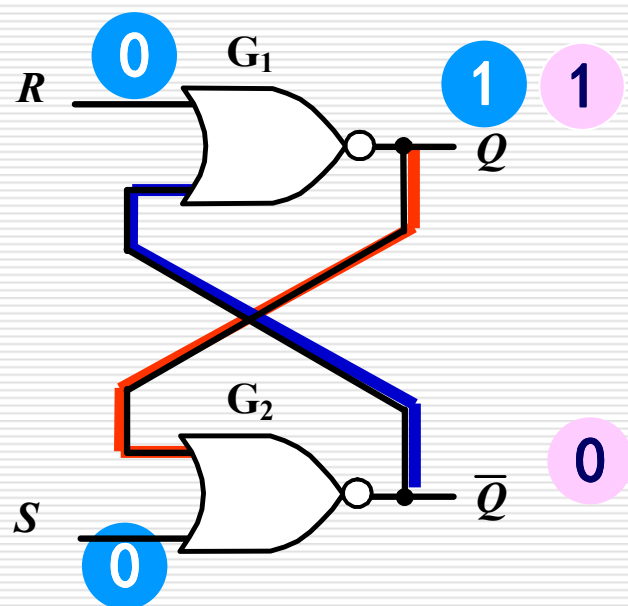


现态： R 、 S 信号作用前 Q 端的状态，现态用 Q^n 表示。 次态： R 、 S 信号作用后 Q 端的状态，次态用 Q^{n+1} 表示。

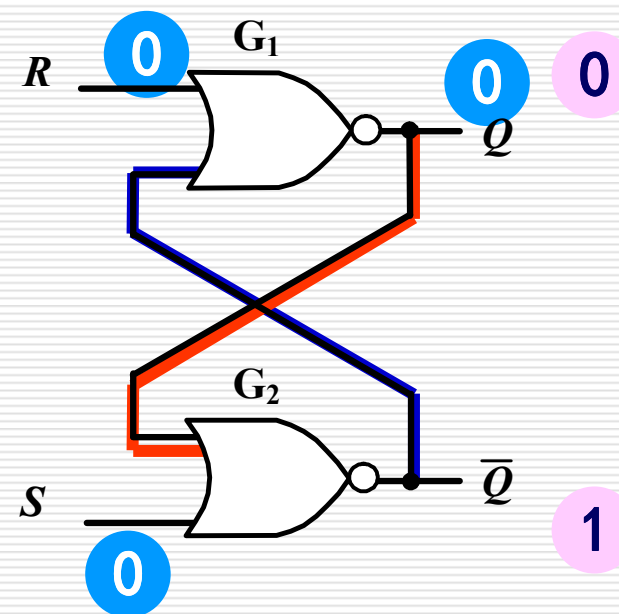
1. 工作原理

$R=0$ 、 $S=0$

状态不变



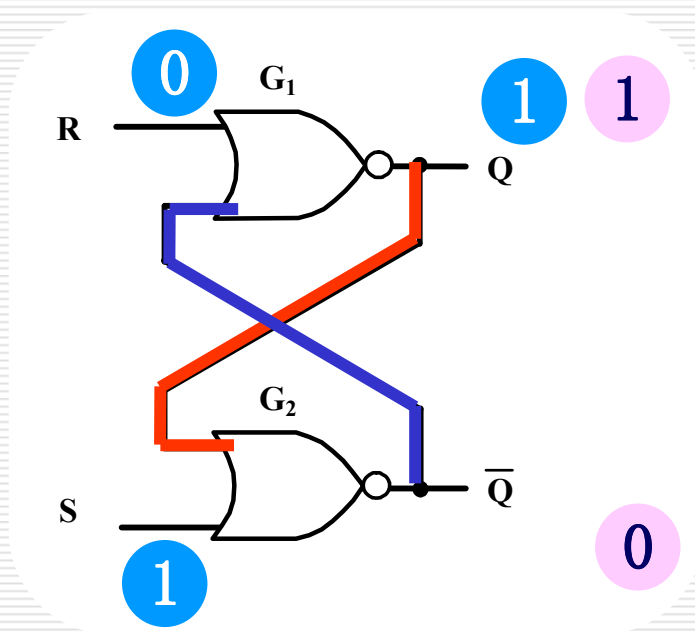
若现态 $Q^n = 1$



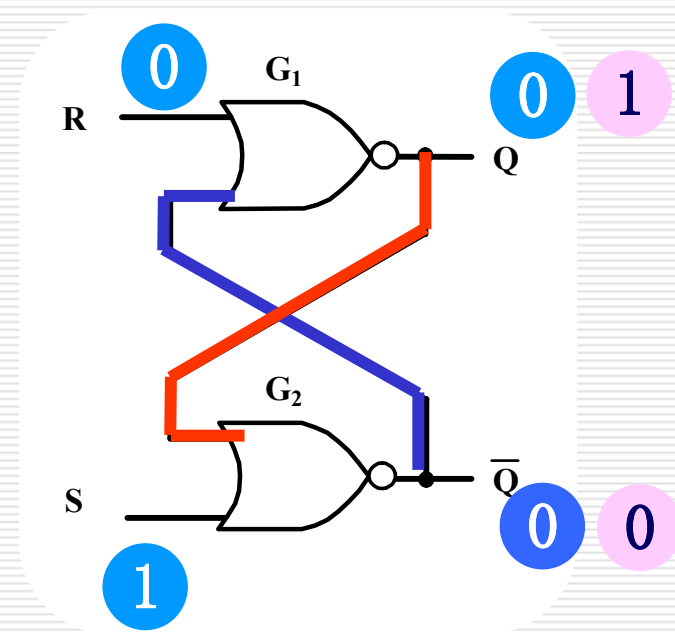
若现态 $Q^n = 0$

$R=0$ 、 $S=1$ 置1

无论现态 Q^n 为0或1，锁存器的次态为1态。信号消失后新的状态将被记忆下来。



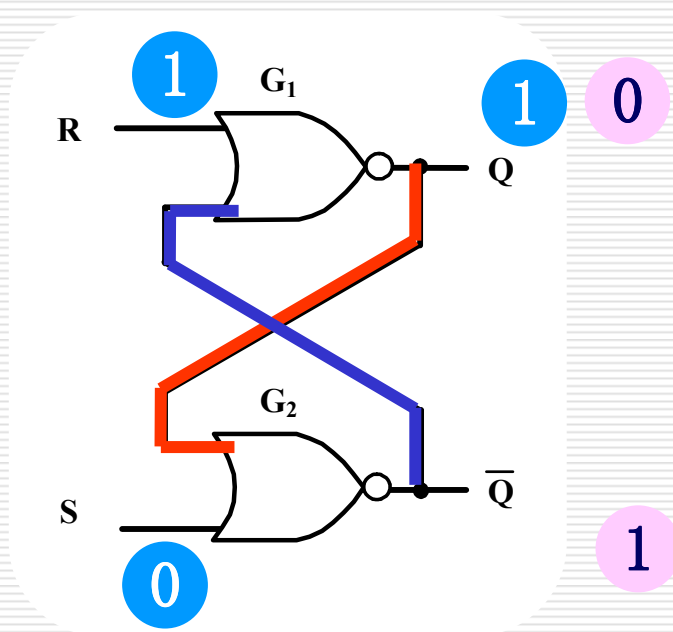
若现态 $Q^n = 1$



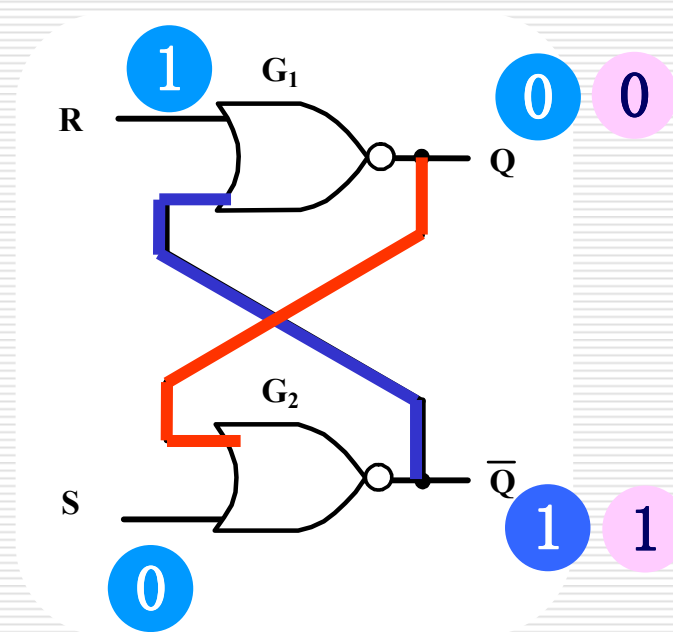
若现态 $Q^n = 0$

$R=1$ 、 $S=0$ 置0

无论现态 Q^n 为0或1，锁存器的次态为0态。信号消失后新的状态将被记忆下来。



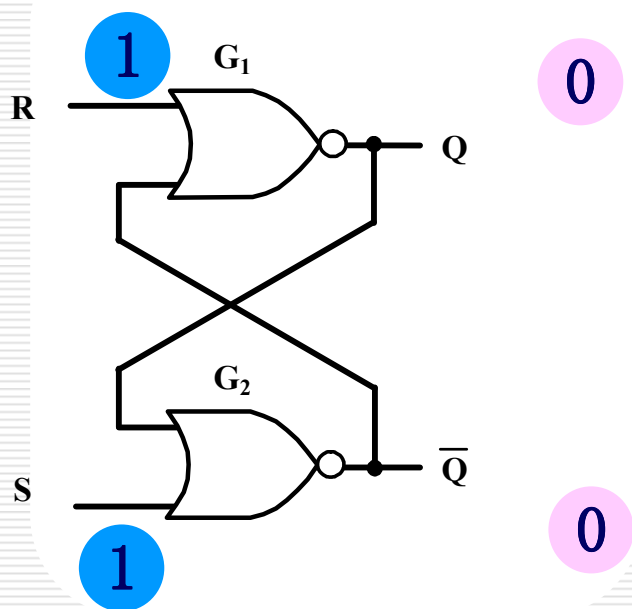
若现态 $Q^n = 1$



若现态 $Q^n = 0$

$S=1$ 、 $R=1$ 状态不确定

无论现态 Q^n 为0或1，触发器的次态 Q^{n+1} 、 \overline{Q}^{n+1} 都为0。

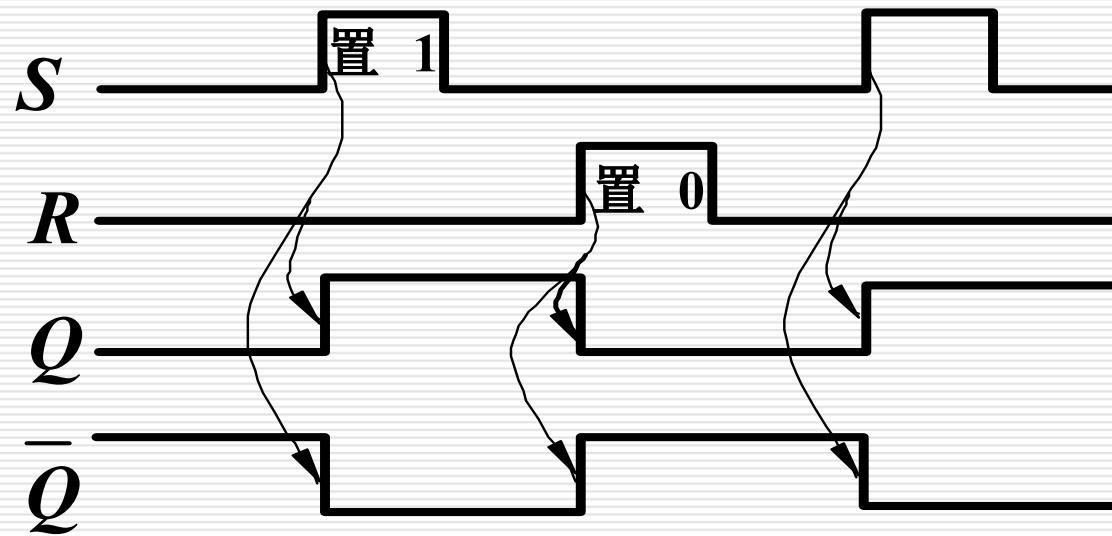


触发器的输出既不是0态，也不是1态

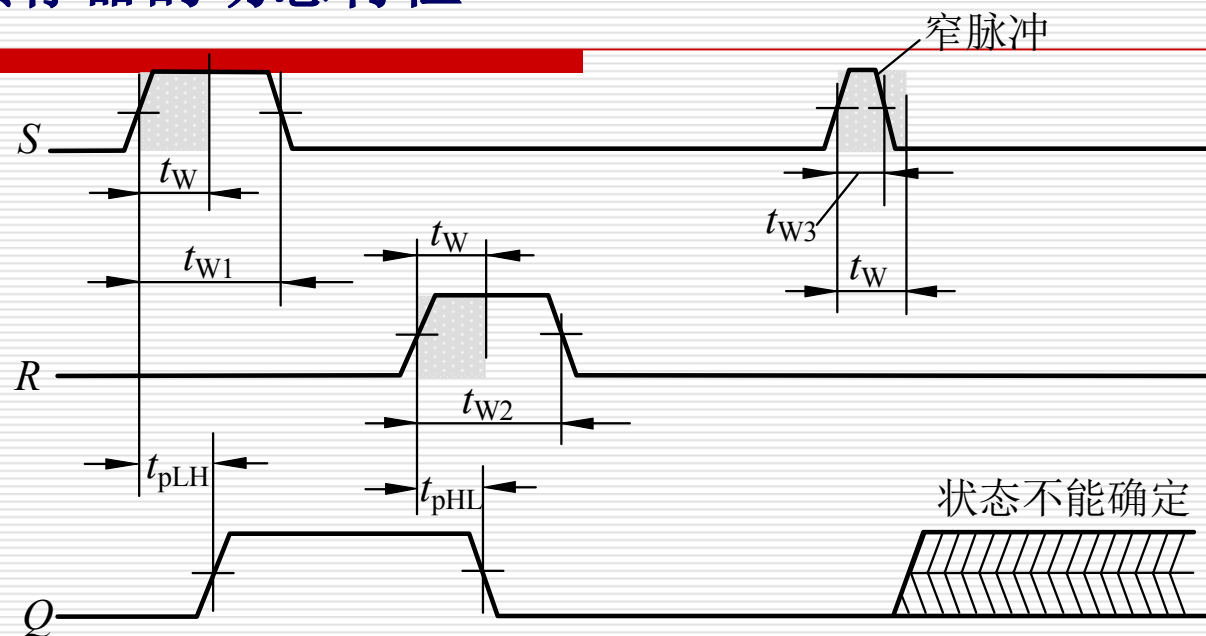
当 S 、 R 同时回到0时，由于两个与非门的延迟时间无法确定，使得触发器最终稳定状态也不能确定。

约束条件: $SR = 0$

工作波形



2. 基本SR锁存器的动态特性



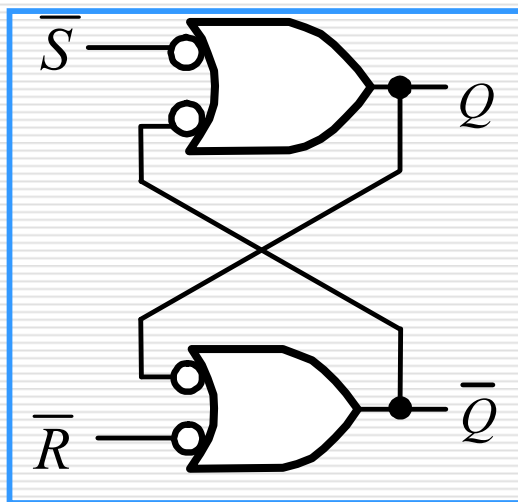
定时图

t_{pLH} 和 t_{pHL} 分别为输出由高到低和由低到高时，相对于输入的延迟时间。

脉冲宽度 t_W ：如果输入脉冲宽度 $< t_W$ ，Q未越过介稳态点，S端信号撤出，会使输出状态不稳定。图中 t_{W1} 和 t_{W2} 均 $> t_W$ 。

3. 用与非门构成的基本SR锁存器

a. 电路图

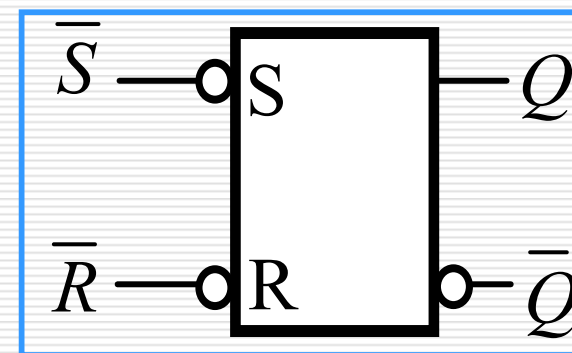


b. 功能表

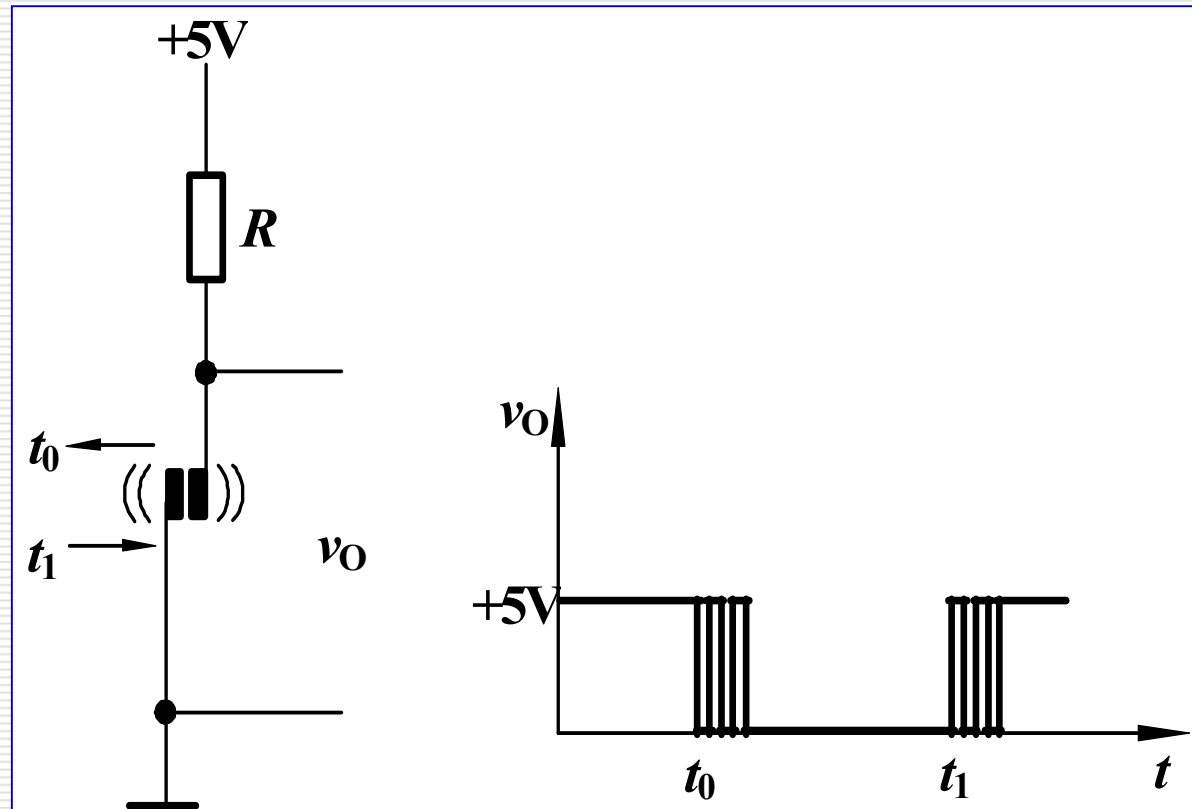
\overline{R}	\overline{S}	Q	\overline{Q}
1	1	不变	不变
1	0	1	0
0	1	0	1
0	0	1	1

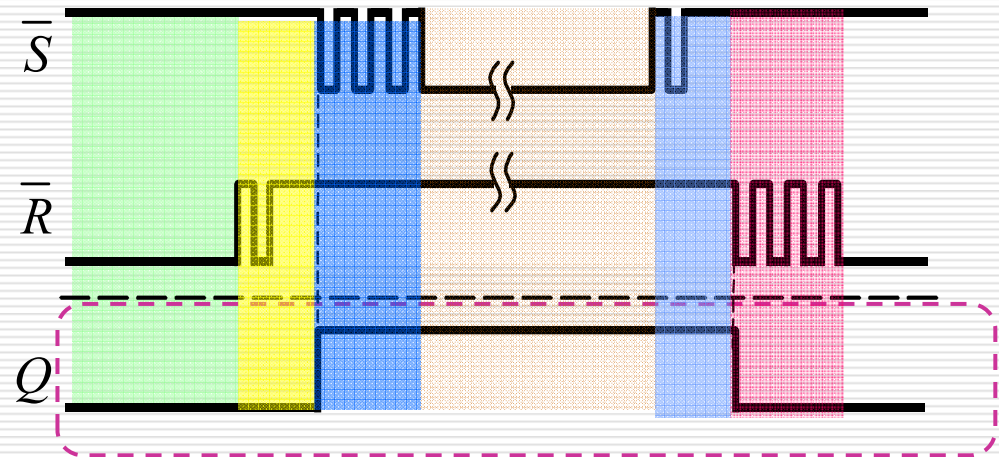
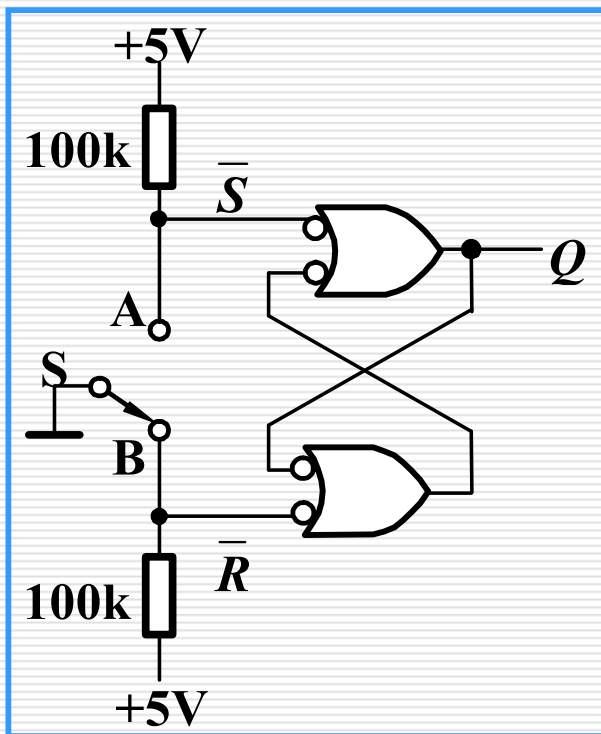
约束条件: $\overline{S} + \overline{R} = 1$

c. 国标逻辑符号



例 运用基本SR锁存器消除机械开关触点抖动引起的脉冲输出。



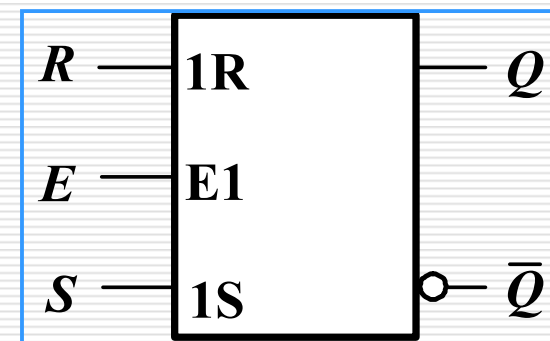
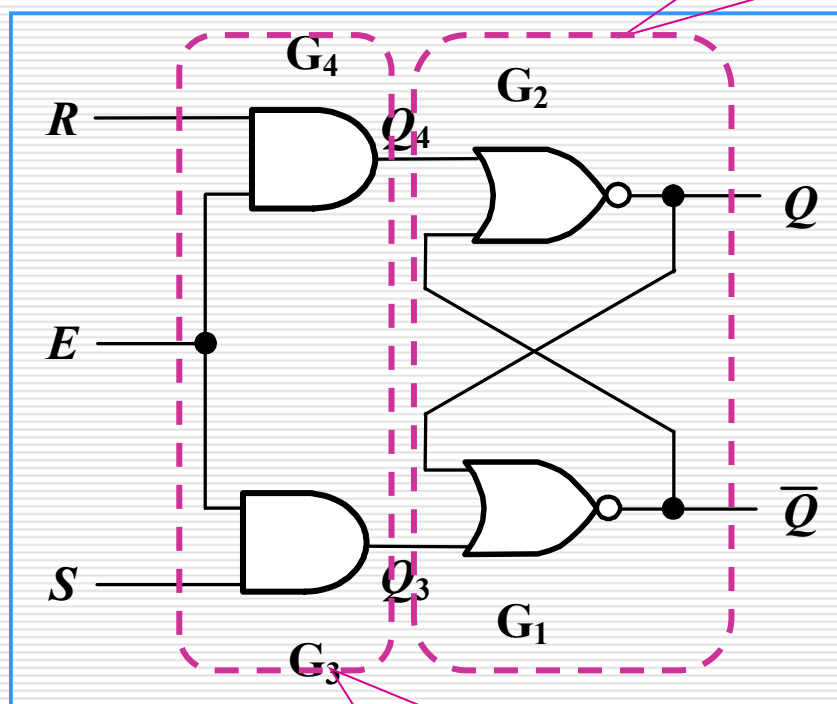


5.2.2 门控SR 锁存器

1. 电路结构

简单SR锁存器

国标逻辑符号



使能信号控制门电路

2、工作原理

$E=0$: 状态不变

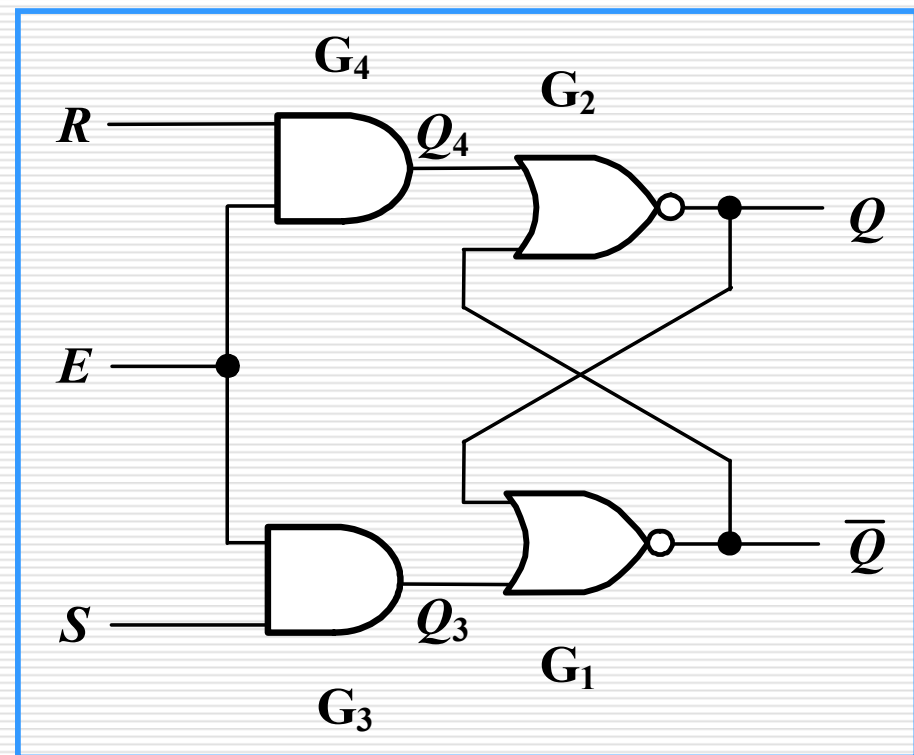
$E=1$: $Q_3 = S$ $Q_4 = R$
状态发生变化。

$S=0, R=0$: $Q^{n+1} = Q^n$

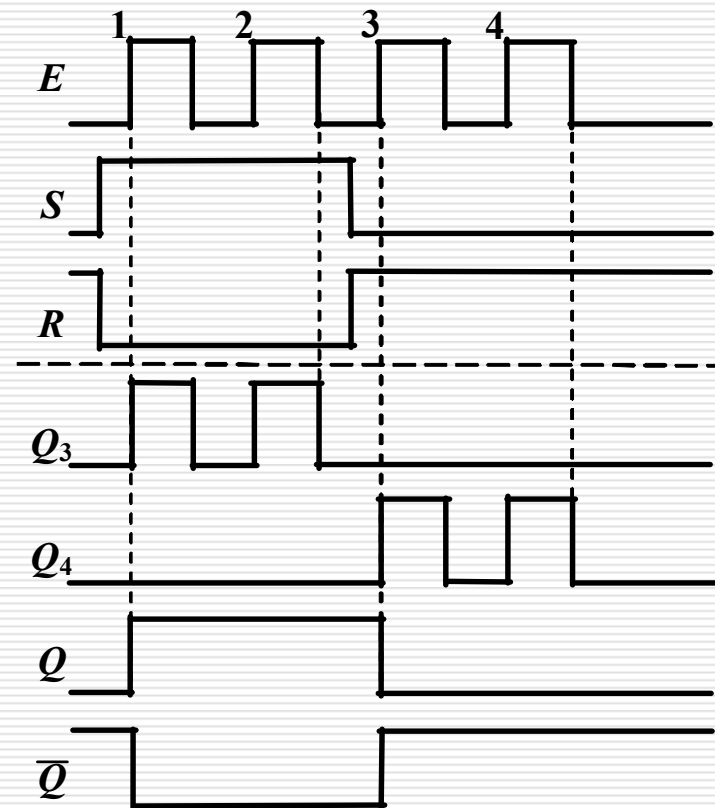
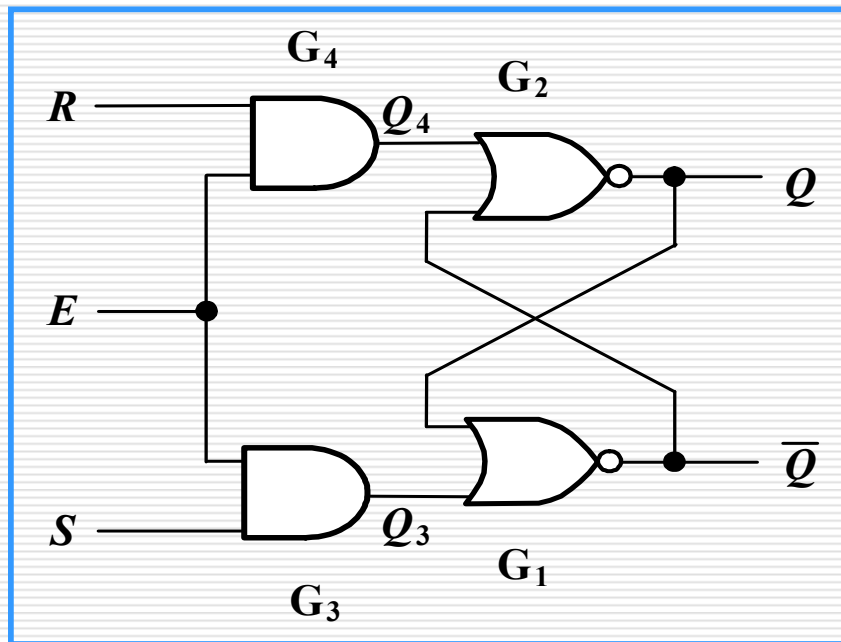
$S=1, R=0$: $Q^{n+1} = 1$

$S=0, R=1$: $Q^{n+1} = 0$

$S=1, R=1$: $Q^{n+1} = \Phi$



例：逻辑门控SR锁存器的 E 、 S 、 R 的波形如下图虚线上边所示，锁存器的原始状态为 $Q = 0$ ，试画出 Q_3 、 Q_4 、 Q 和 \bar{Q} 的波形。



5.3 D锁存器

5.3.1 D 锁存器的电路结构

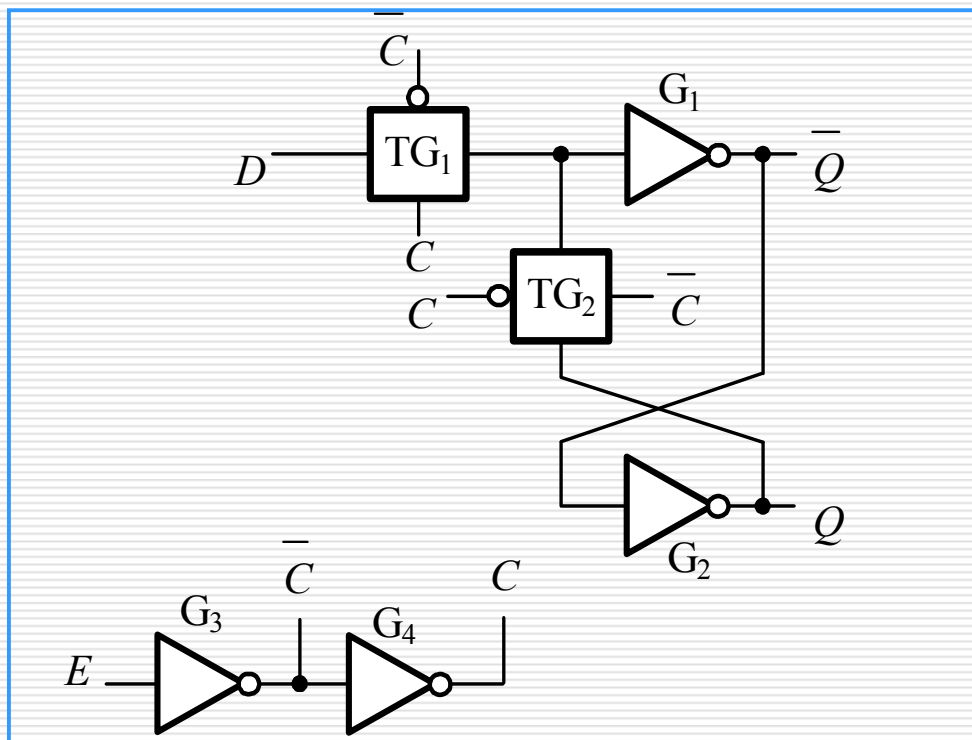
5.3.2 典型的 D 锁存器集成电路

5.3.3 D 锁存器的动态特性

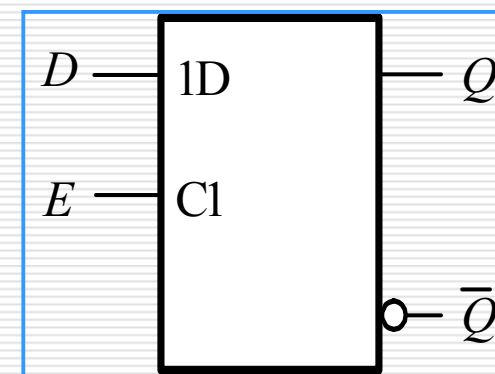
5.3.1 D锁存器的电路结构

1. 传输门控D锁存器

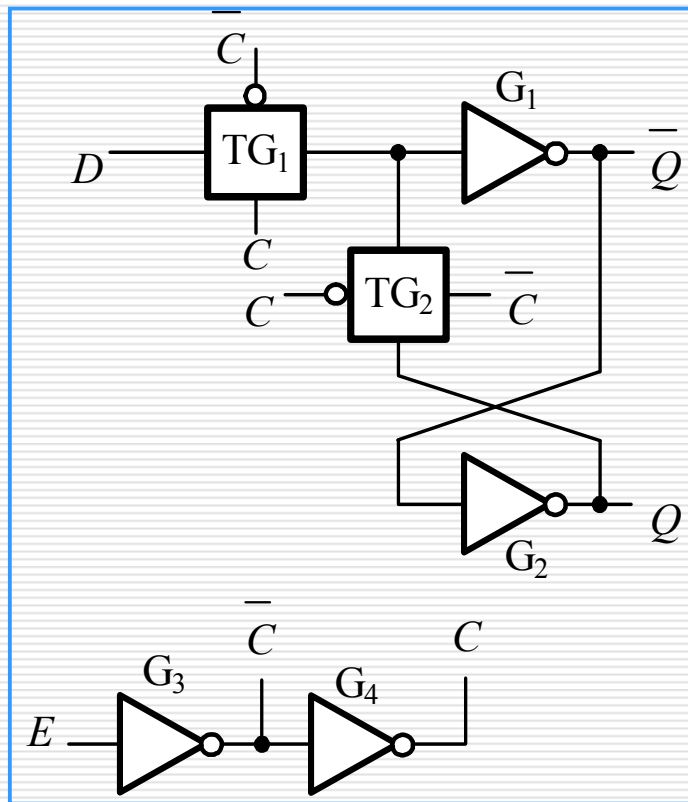
(1) 逻辑电路图



逻辑符号

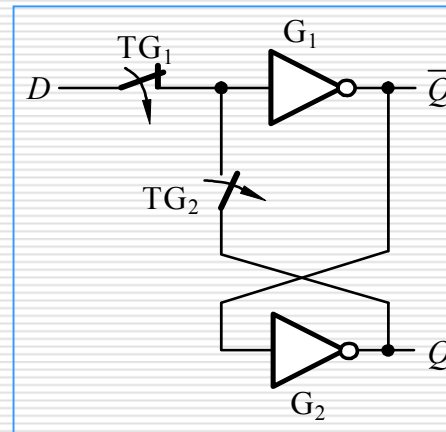


(2)工作原理



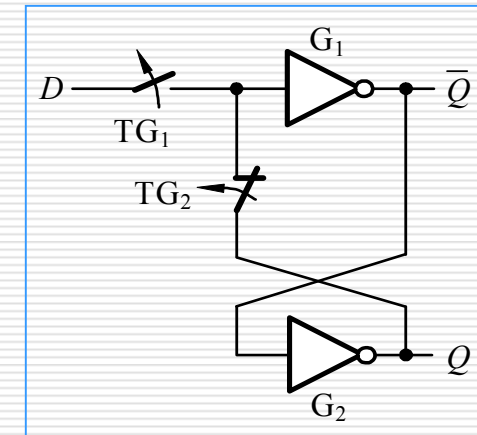
(a) $E=1$ 时

TG_1 导通,
 TG_2 断开
 $Q = D$



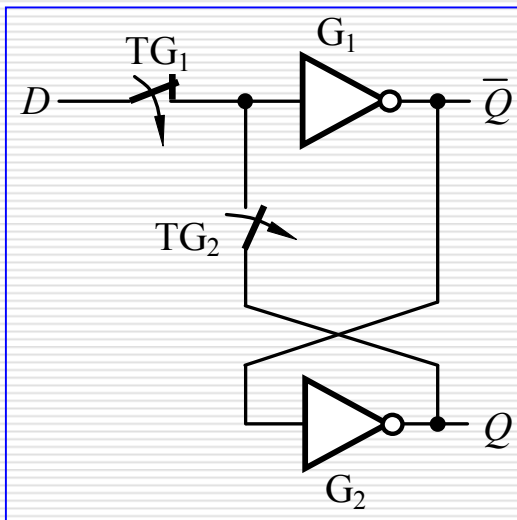
(b) $E=0$ 时

TG_2 导通,
 TG_1 断开
 Q 不变



(3) 逻辑功能

D锁存器的功能表



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

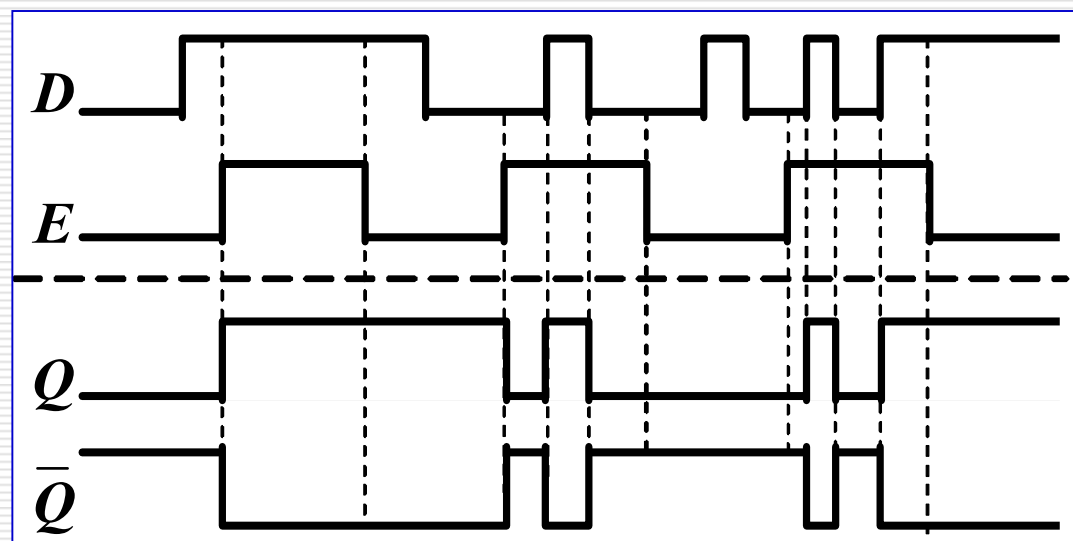
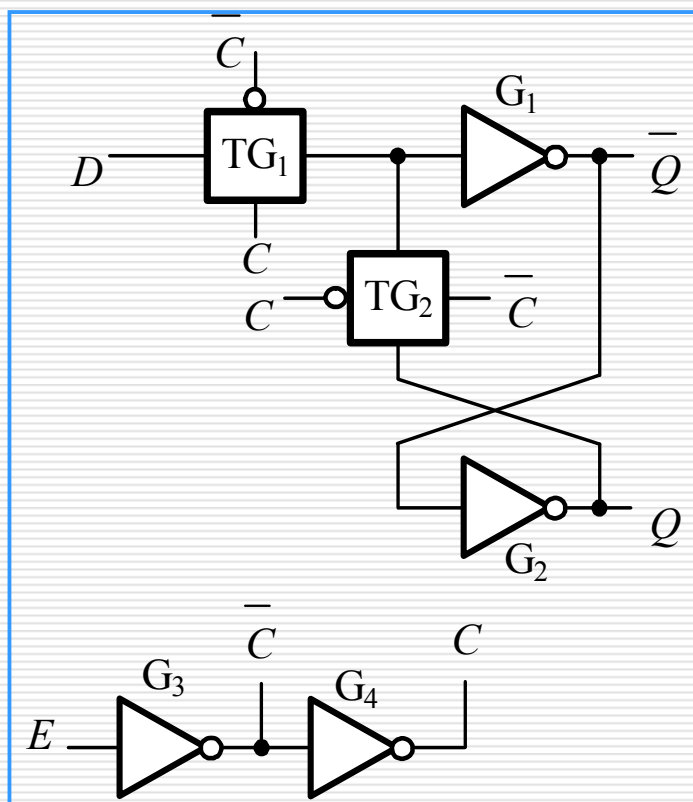
$E=0$,

Q 不变

$E=1$

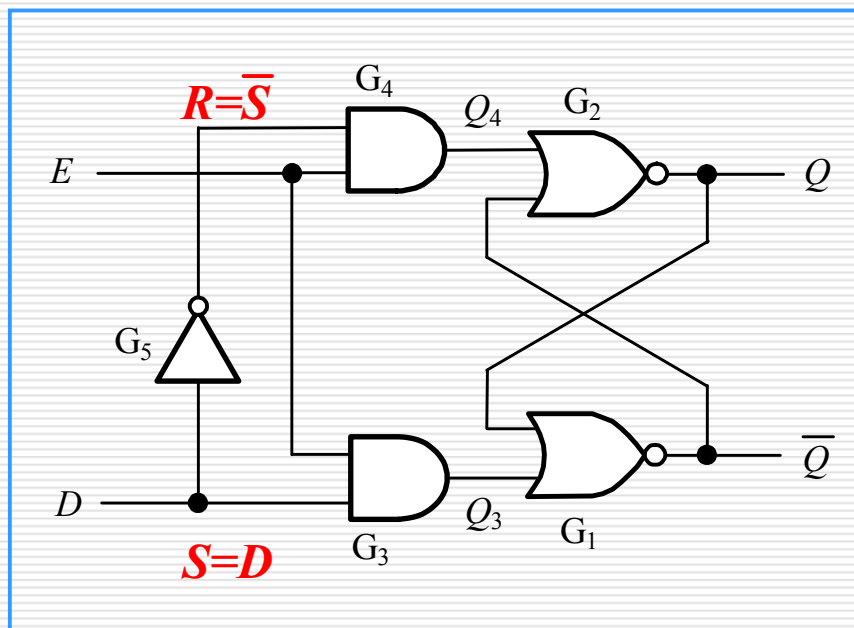
$Q = D$

(4) 工作波形



2. 逻辑门控D锁存器

逻辑电路图



D锁存器的功能表

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

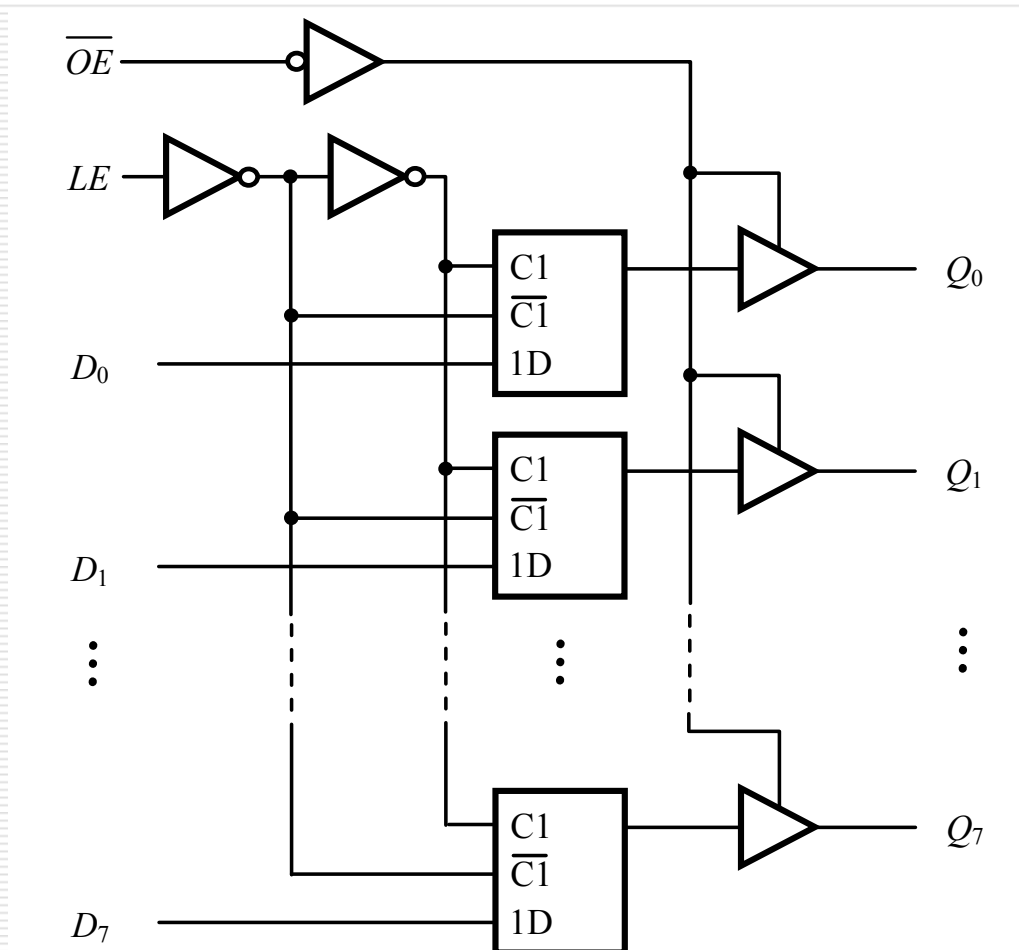
$E=0$ Q 不变

$E=1$ $D=0$ $S=0$ $R=1$ $Q=0$

$D=1$ $S=1$ $R=0$ $Q=1$

5.3.2 典型的D锁存器集成电路

74HC/HCT373 八D锁存器



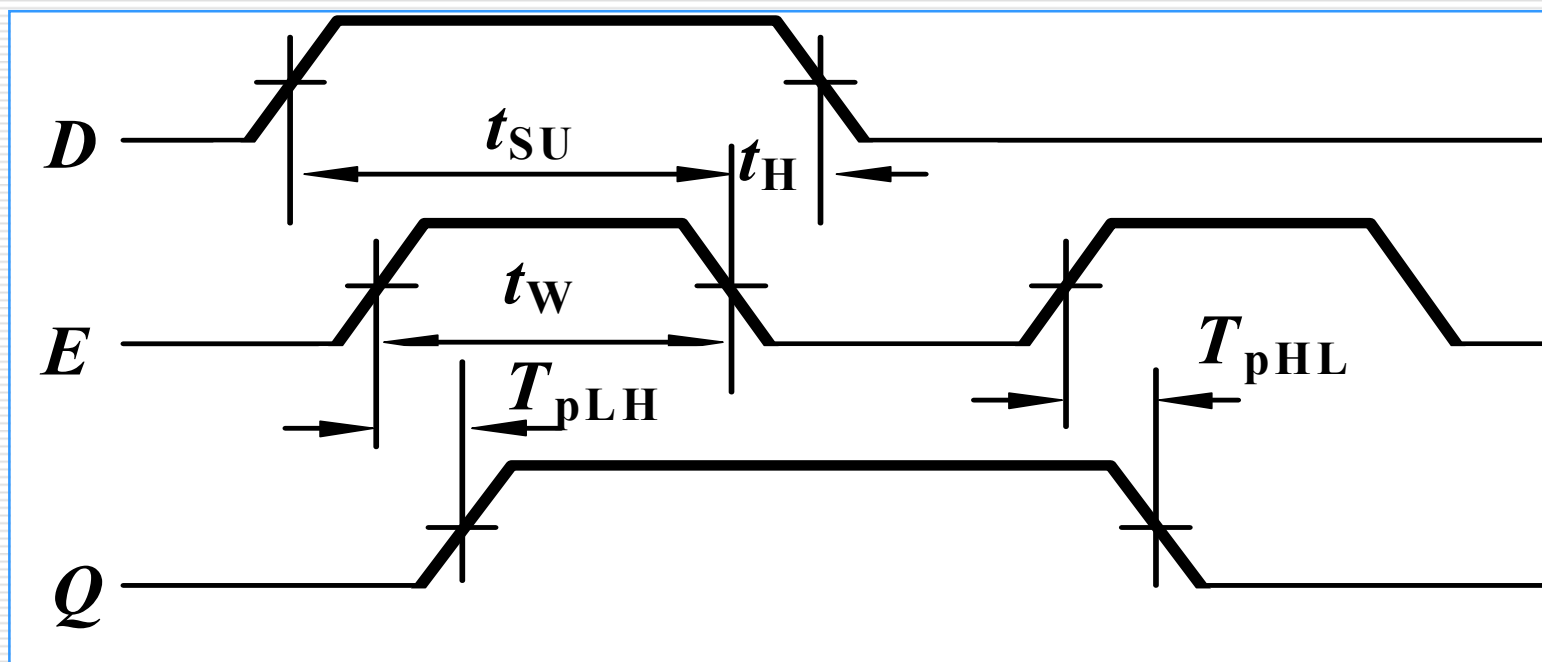
74HC/HCT373的功能表

工作模式	输 入			内部锁存器 状 态	输 出
	OE	LE	D_n		Q_n
使能和读锁存器 (传送模式)	L	H	L	L	L
	L	H	H	H	H
锁存和读锁存器	L	L	L*	L	L
	L	L	H*	H	H
锁存和禁止输出	H	×	×	×	高阻
	H	×	×	×	高阻

L*和H*表示门控电平 LE 由高变低之前瞬间 D_n 的逻辑电平。

5.3.3 D锁存器的动态特性

定时图:表示电路动作过程中, 对各输入信号的时间要求以及输出对输入信号的响应时间。



有建立时间 t_{SU} 、保持时间 t_H 、脉冲宽度 t_W 等。

5.4 触发器的电路结构和工作原理

5.4.1 主从D触发器的电路结构和工作原理

5.4.2 典型主从D触发器集成电路

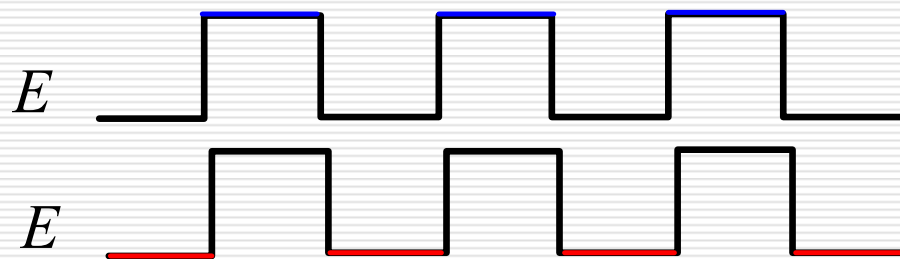
5.4.3 主从D触发器的动态特性

5.4.4 其他电路结构的触发器

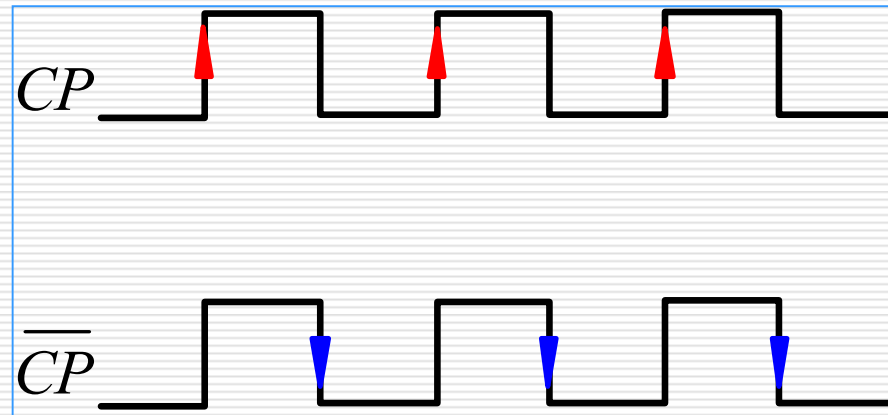
5.4 触发器的电路结构和工作原理

1. 锁存器与触发器

锁存器在**E**的高(低)电平期间对信号敏感



触发器在**CP**的上升沿(下降沿)对信号敏感



在VerilogHDL中对锁存器与触发器的描述语句是不同的

5.4 触发器的电路结构和工作原理

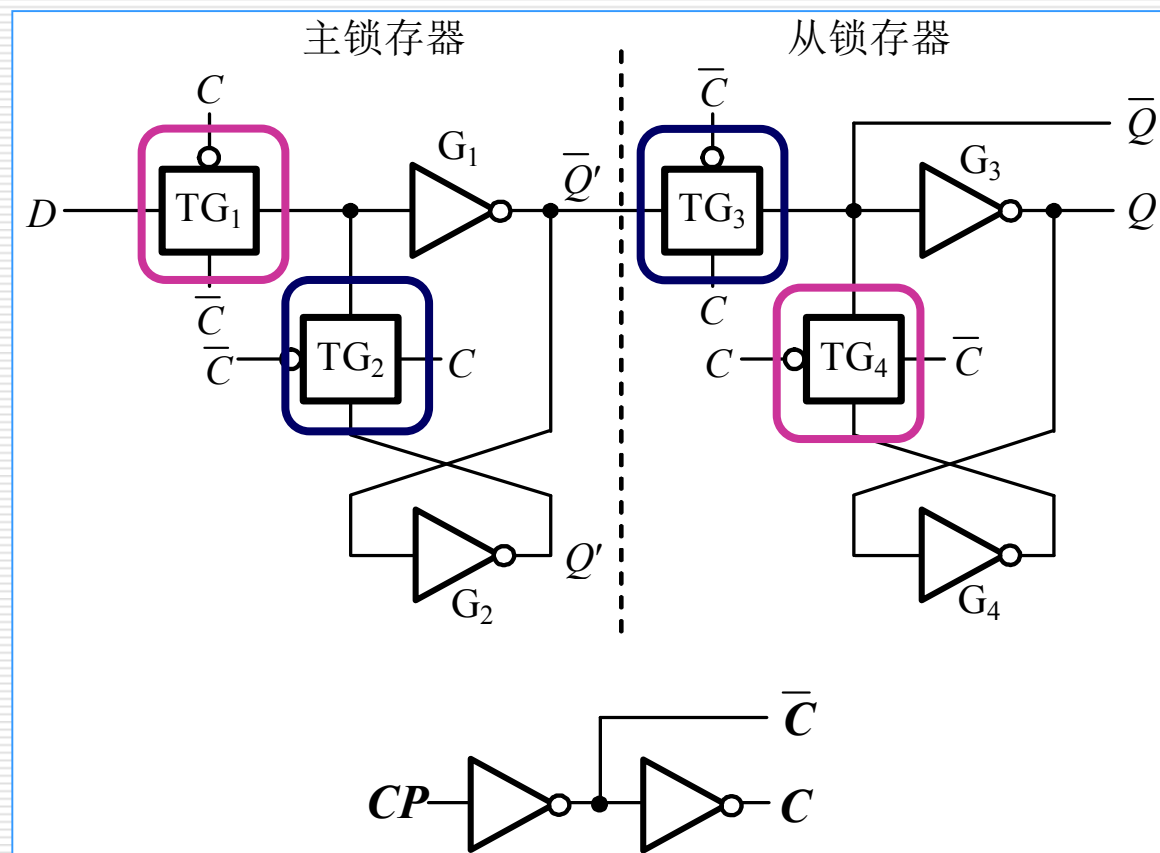
5.4.1 主从D触发器的电路结构和工作原理

1. 电路结构

主锁存器与从锁存器结构相同

TG₁和**TG₄**的工作状态相同

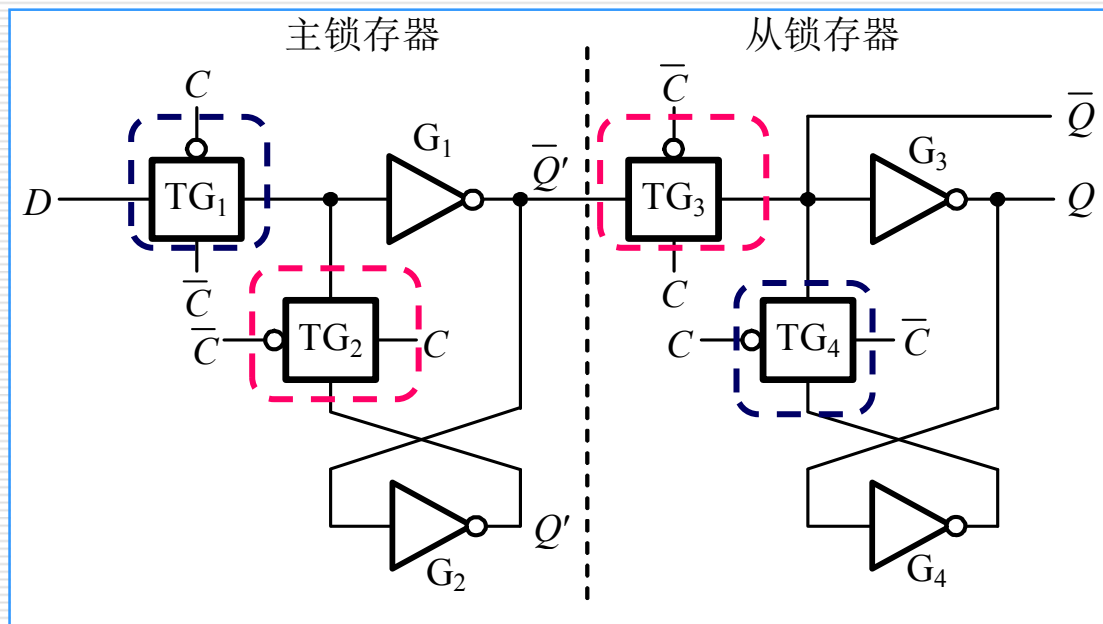
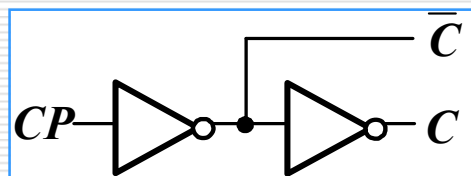
TG₂和**TG₃**的工作状态相同



2. 工作原理

(1) $CP=0$ 时:

$\bar{C}=1, C=0,$



TG_1 导通, TG_2 断开——输入信号 D 送入主锁存器。

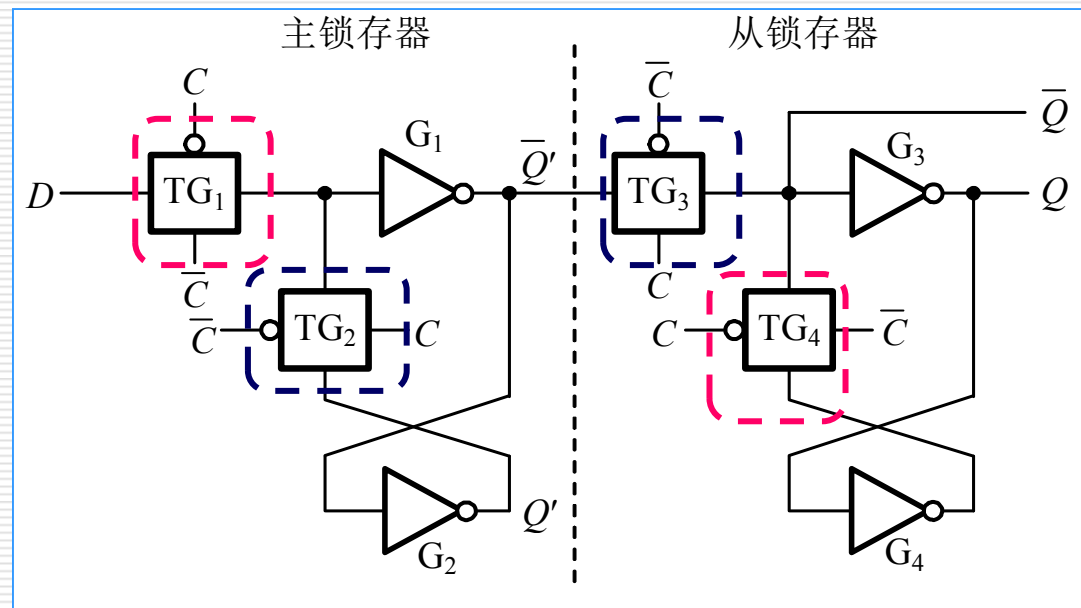
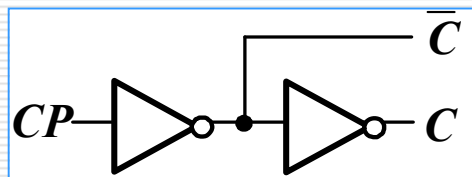
Q' 跟随 D 端的状态变化, 使 $Q'=D$ 。

TG_3 断开, TG_4 导通——从锁存器维持在原来的状态不变。

2. 工作原理

(2) CP由0跳变到1：

$$\bar{C} = 0, C = 1,$$



TG_1 断开, TG_2 导通——输入信号 D 不能送入主锁存器。

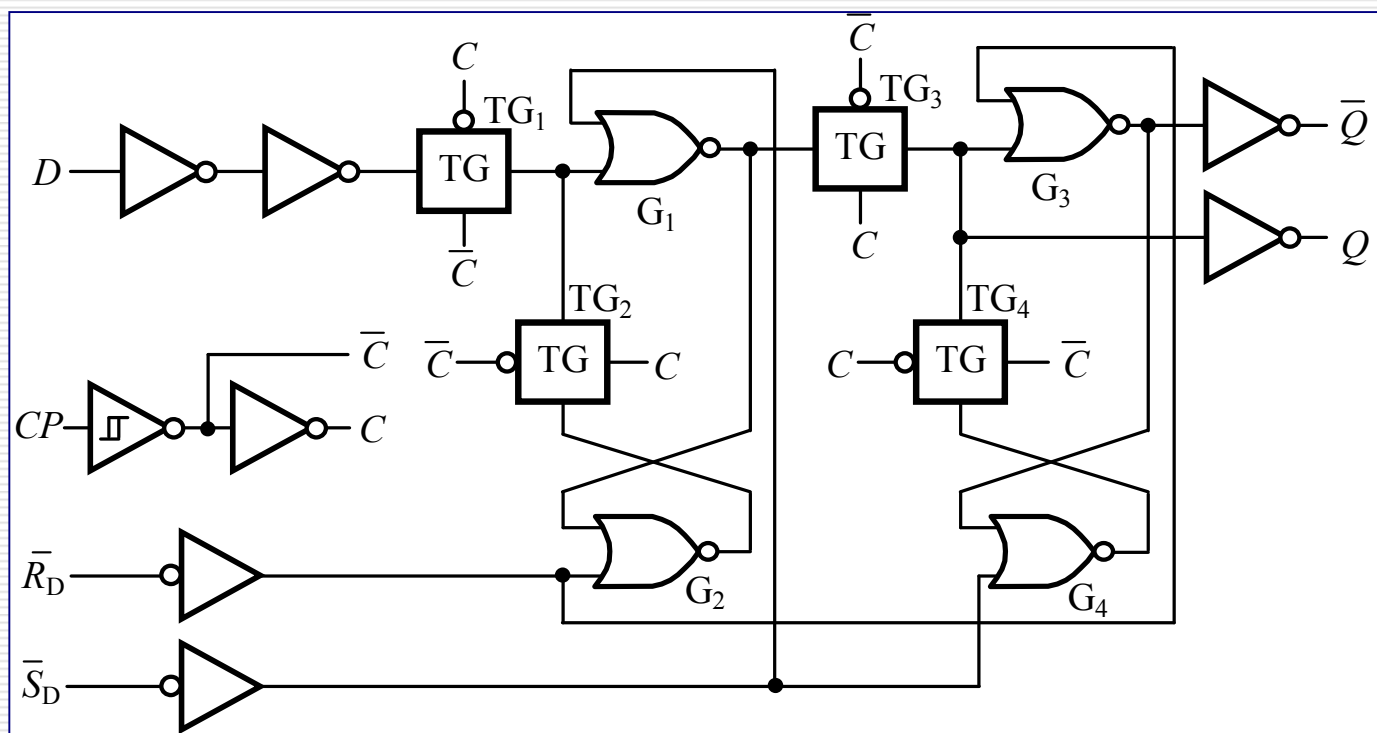
主锁存器维持原态不变。

TG_3 导通, TG_4 断开——从锁存器 Q' 的信号送 Q 端。

触发器的状态仅仅取决于 CP 信号上升沿到达前瞬间的 D 信号

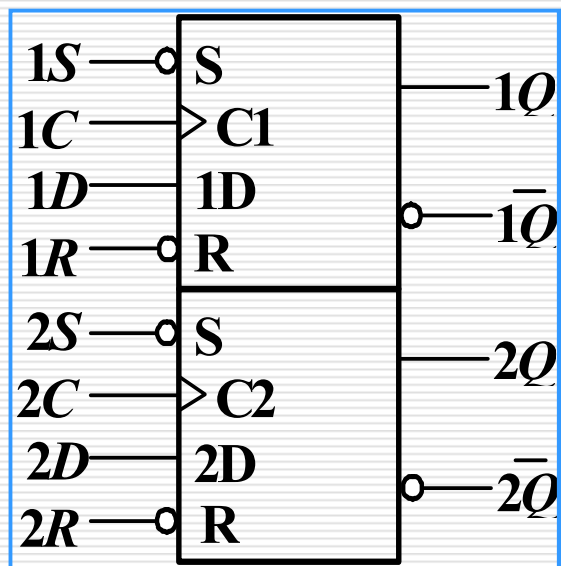
5.4.2 典型主从D触发器集成电路

74HC/HCT74 中D触发器的逻辑图



74HC/HCT74的逻辑符号和功能表

74HC/HCT74的功能表



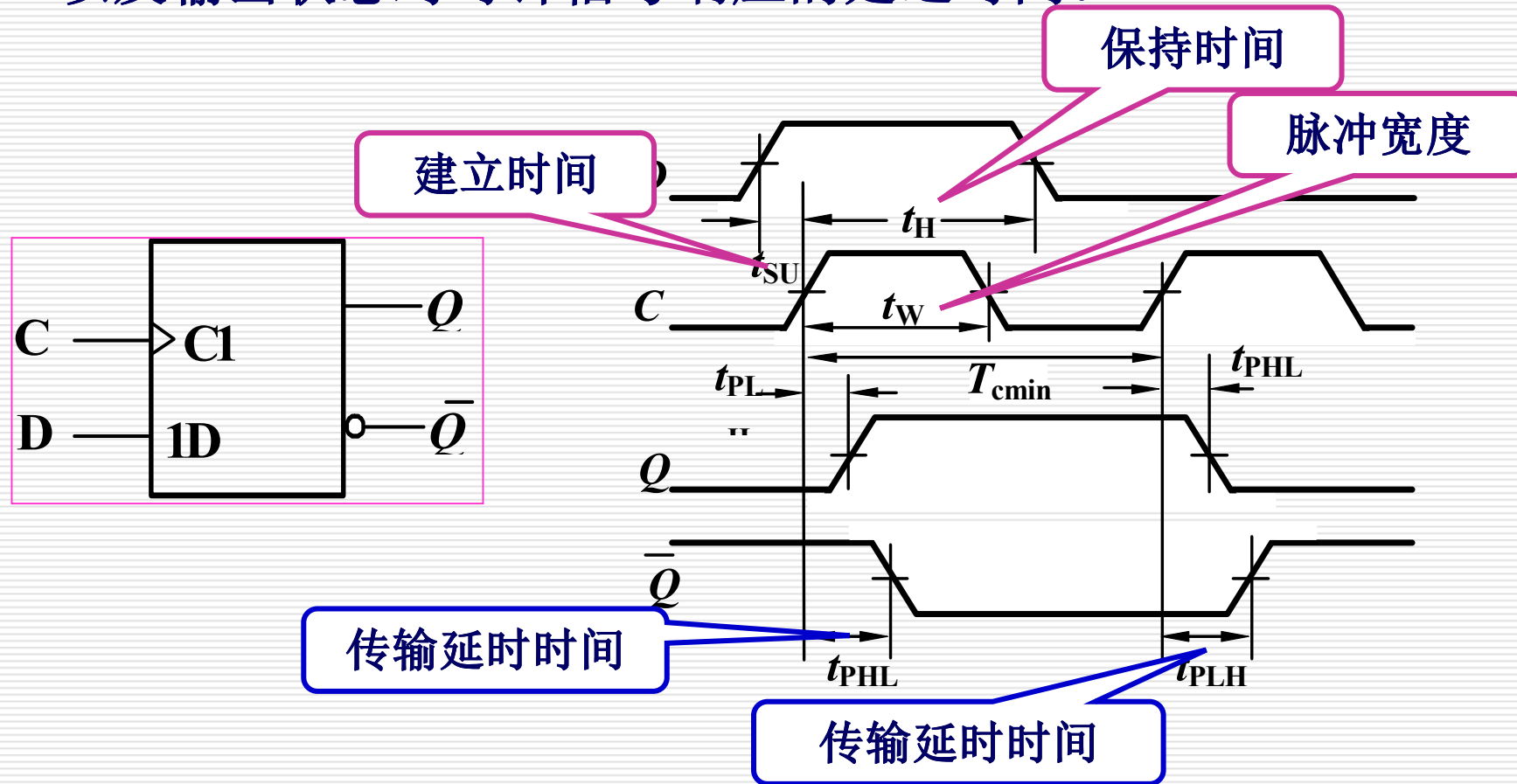
国标逻辑符号

输 入				输 出	
\overline{S}_D	\overline{R}_D	CP	D	Q	\overline{Q}
L	H	×	×	H	L
H	L	×	×	L	H
L	L	×	×	H	H
\overline{S}_D	\overline{R}_D	CP	D	Q^{n+1}	\overline{Q}^{n+1}
H	H	↑	L	L	H
H	H	↑	H	H	L

具有直接置1、直接置0，正边沿触发的D功能触发器

5.4.3 主从D触发器的动态特性

动态特性反映其触发器对输入信号和时钟信号间的时间要求，以及输出状态对时钟信号响应的延迟时间。



建立时间 t_{SU} ：保证与 D 相关的电路建立起稳定的状态，使触发器状态得到正确的转换。

保持时间 t_{H} ：保证 D 状态可靠地传送到 Q

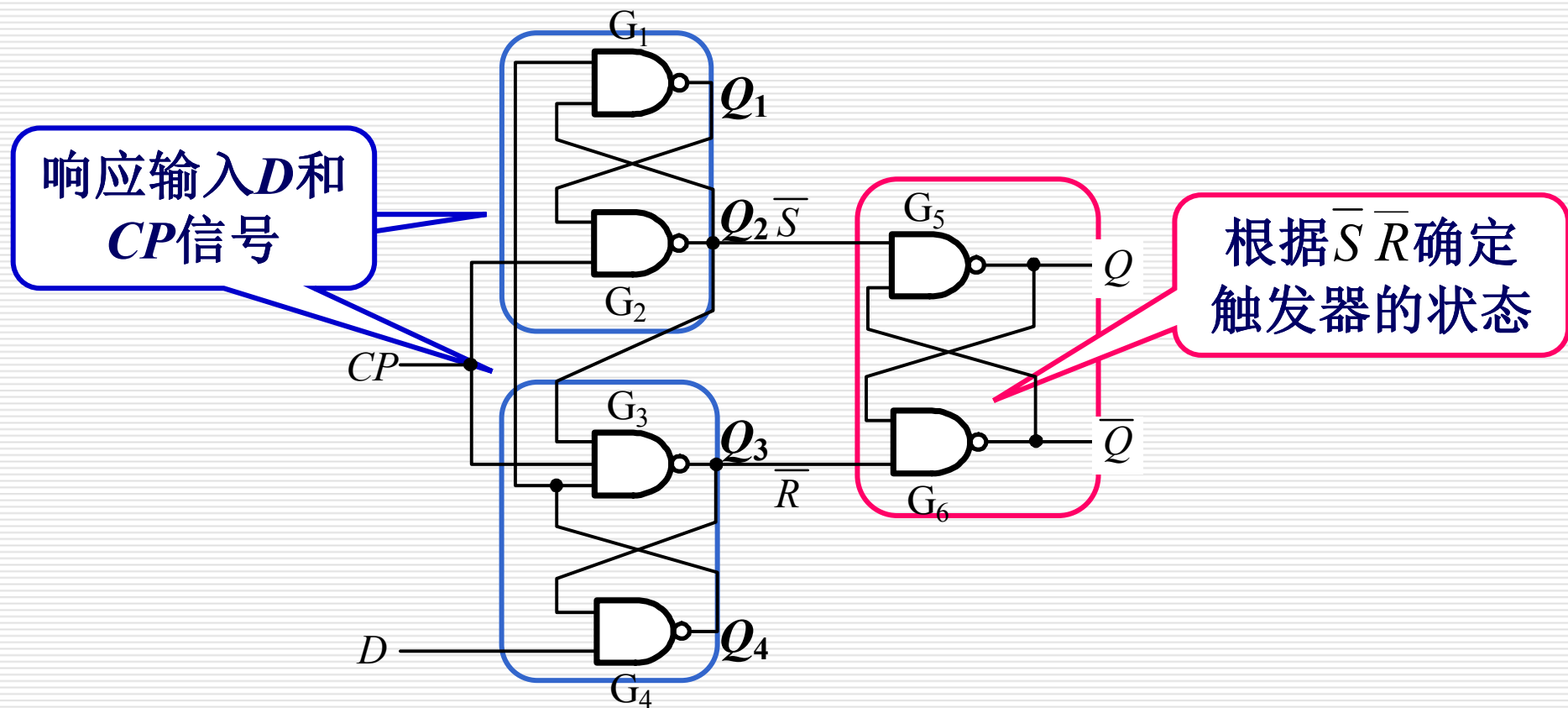
触发脉冲宽度 t_{W} ：保证内部各门正确翻转。

传输延迟时间 t_{PLH} 和 t_{PHL} ：时钟脉冲 CP 上升沿至输出端新状态稳定建立起来的时间

最高触发频率 f_{cmax} ：触发器内部都要完成一系列动作，需要一定的时间延迟，所以对于 CP 最高工作频率有一个限制。

5.4.4 其他电路结构的触发器

1. 维持阻塞触发器



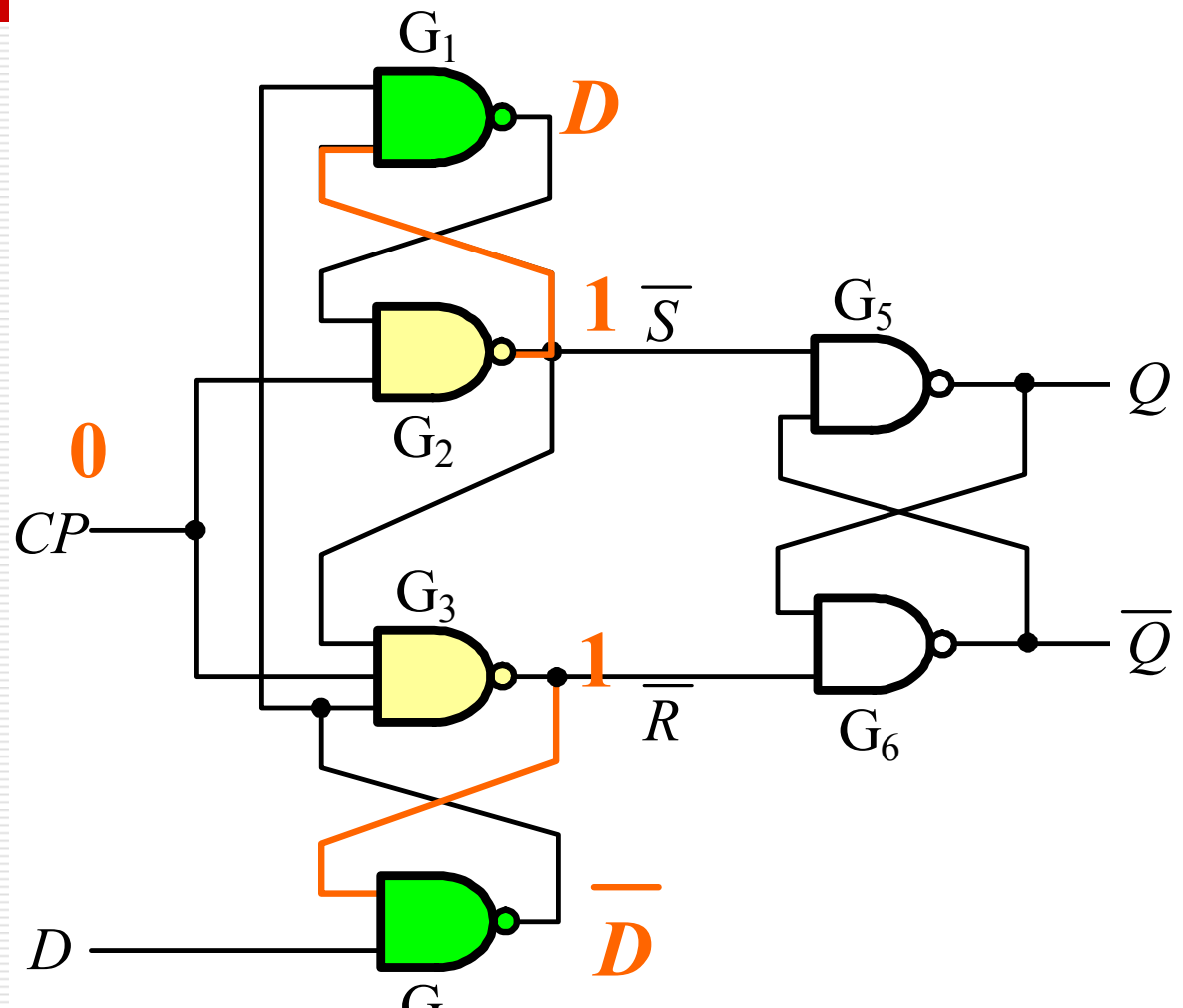
2、工作原理

$$CP = 0$$

$$Q_4 = \bar{D} \quad Q_1 = D$$

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

\bar{D} 信号存于 Q_4

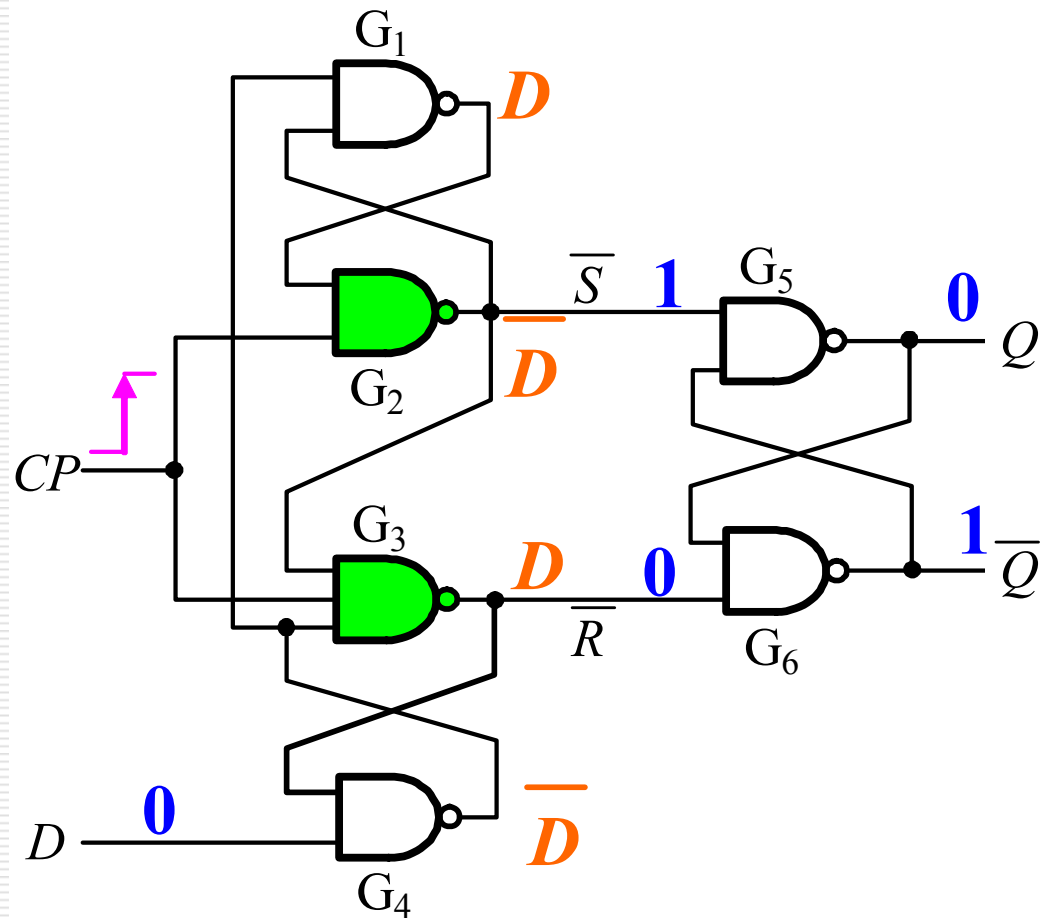


D 信号进入触发器, 为状态刷新作好准备

当 CP 由0跳变为1

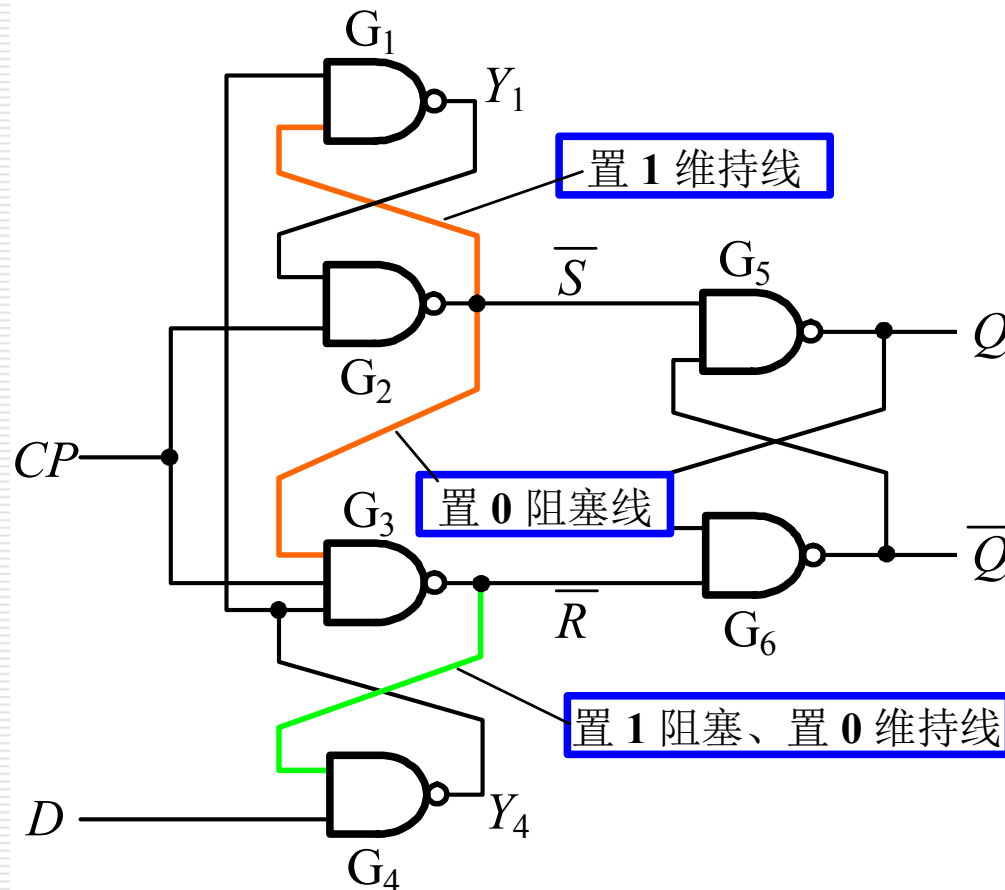
$$Q^{n+1} = D$$

在 CP 脉冲的上升沿，触发器按此前的 D 信号刷新



当 $CP = 1$

D 信号不影响 \overline{S} 、 \overline{R} 的状态， Q 的状态不变



在 CP 脉冲的上升沿到来瞬间使触发器的状态变化