

2019-2020学年秋季学期

第五部分 寄存器

半导体存储电路

数字电路需要在电路中保存电路的状态、中间结果、配置信息甚至任何代码与数据。这就需要存储电路。

○ 半导体存储电路

○ 寄存器 (register)

- 锁存器 (latch) , 寄存器 (register)
- 高速电路, 无读写限制, 用于时序逻辑电路的设计
- 门电路的基本单元

○ 静态随机存储器 (SRAM: static random access memory)

- 缓存, 片上内存
- 高速电路, 面积小, 单一时刻只能读写一个数据, 用于在片上存储大量数据

○ 动态随机存储器 (DRAM: dynamic random access memory)

- 片内内存, 片外内存
- 速度较低, 面积很小, 读写带宽大, 地址受限

○ 非易失存储器 (non-volatile memory)

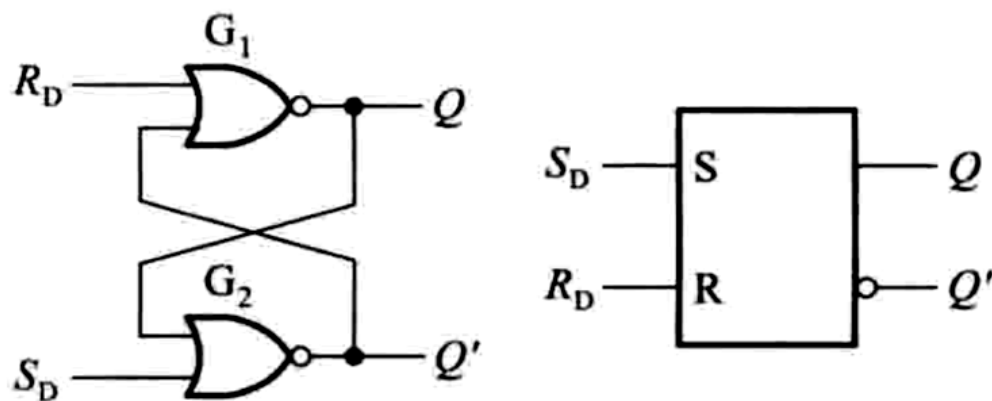
- 掉电不丢失内容的存储器
- 片上只读内存 (ROM) , 片外可编程ROM (Flash)

在逻辑电路中保存过程变量的常用电路

- 锁存器：在特定输入下才能保持状态的状态保持电路
- 触发器：使用触发信号触发锁存的状态保持电路
- 寄存器：使用沿触发信号的触发器

寄存器是现在同步时序电路主要使用的状态保存电路。

SR锁存器—或非门



S	R	Q^*	Q'^*
0	0	Q	Q'
0	1	0	1
1	0	1	0
1	1	0	0

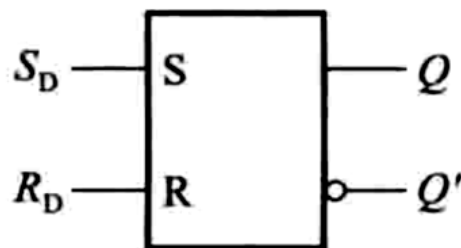
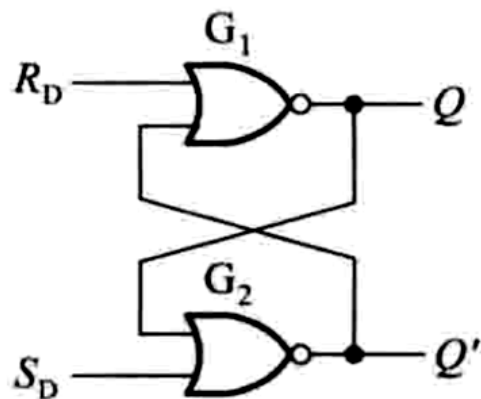
锁存状态

复位状态

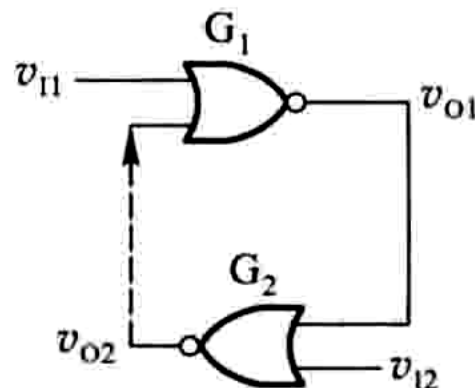
置位状态

非法状态

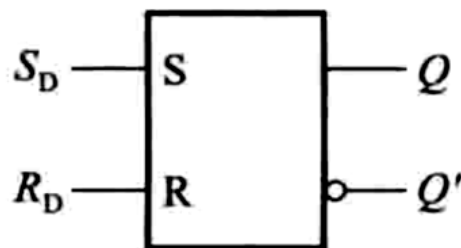
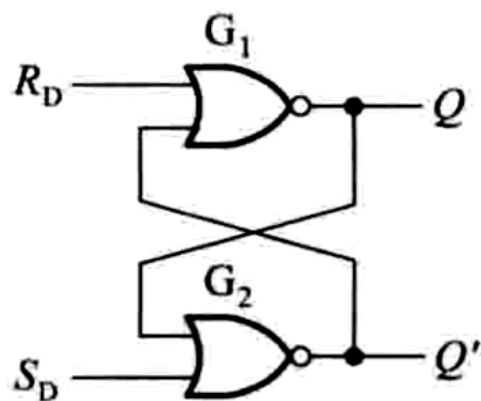
SR锁存器—锁存状态



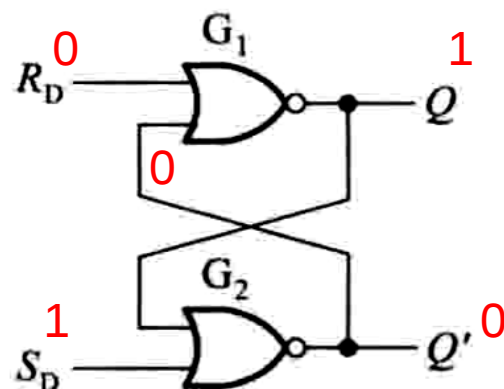
当R和S同时为0时，电路形成一个闭环，电路输出保持之前的状态。



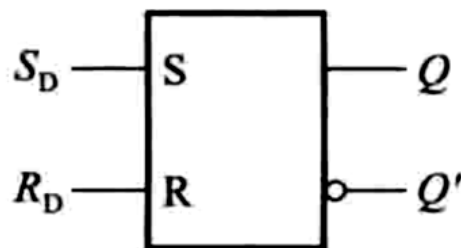
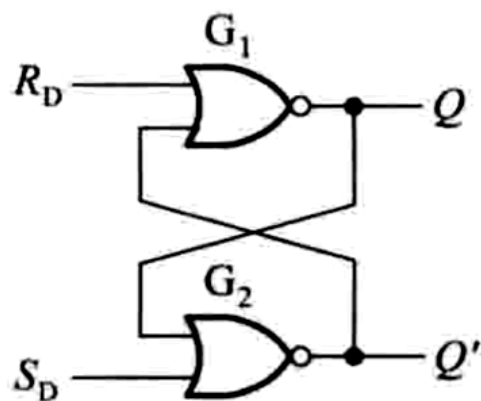
SR锁存器—置位状态 S:set



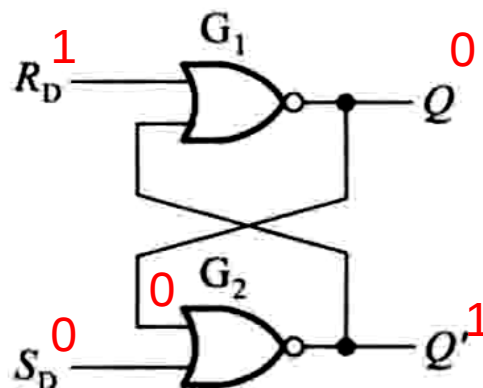
当 $S=1$, $R=0$ 时, Q' 被 S 强制为0, Q' 和 R 都为0, 导致 Q 为1。



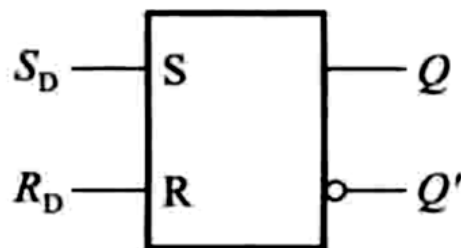
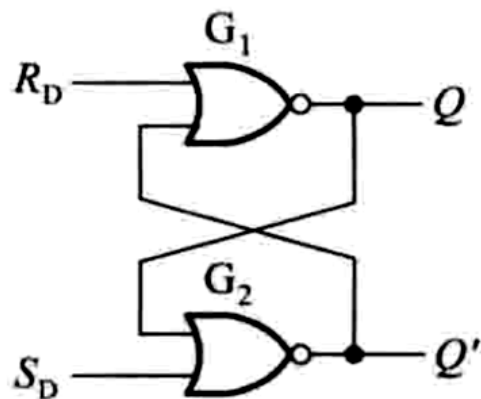
SR锁存器—复位状态 R:reset



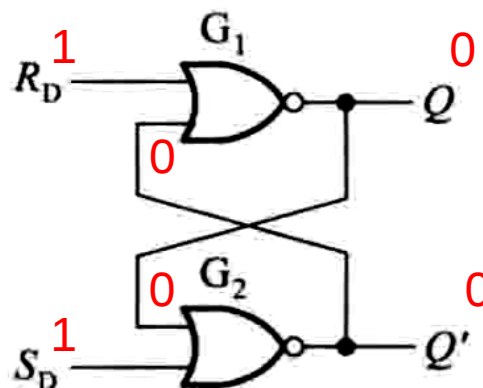
当 $S=0$, $R=1$ 时, Q 被 R 强制为0, Q 和 S 都为0, 导致 Q' 为1。



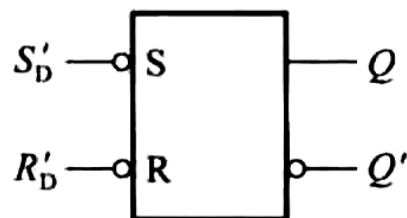
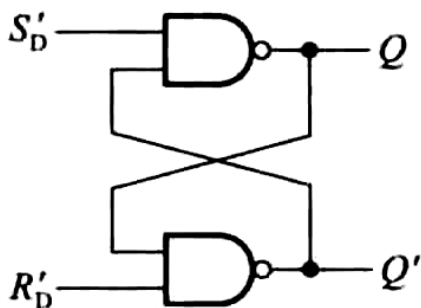
SR锁存器—非法状态



当 $S=1, R=1$ 时， Q 被 R 强制为0， Q' 被 S 强制为0。如果 R 和 S 同时变为0，锁存器的输出不可知，出现不定态（亚稳定状态）。



SR锁存器—与非门



S'	R'	Q^*	Q'^*
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q	Q'

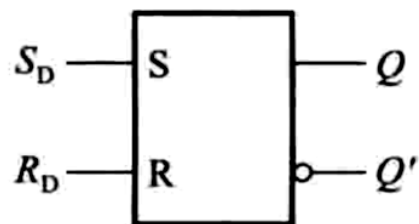
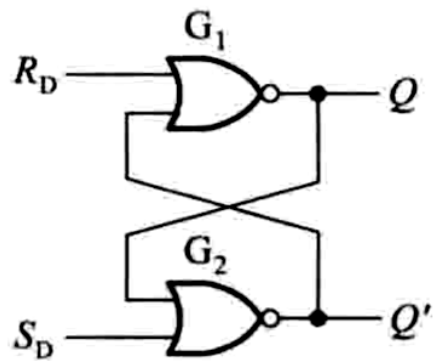
非法状态

置位状态

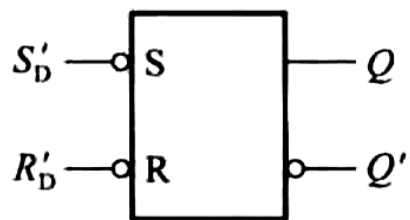
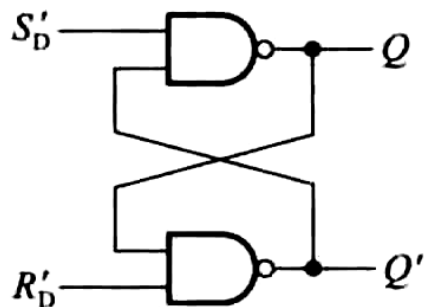
复位状态

锁存状态

两种SR锁存器

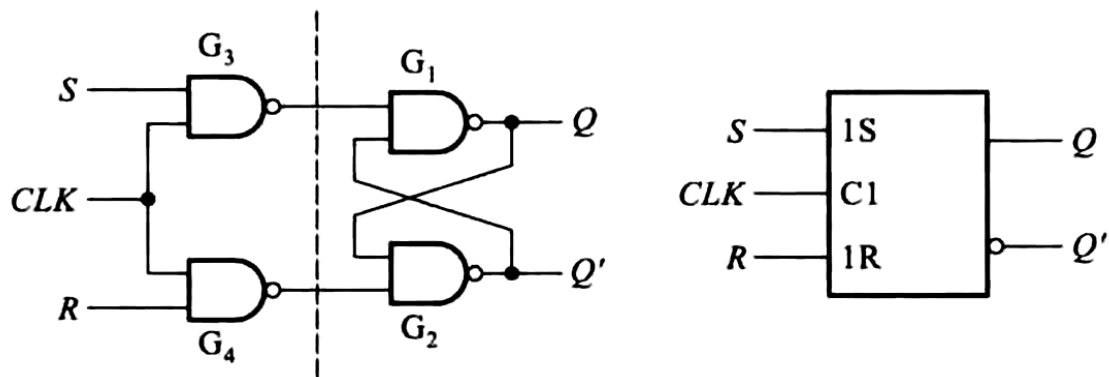


S	R	Q^*	Q'^*
0	0	Q	Q'
0	1	0	1
1	0	1	0
1	1	0	0



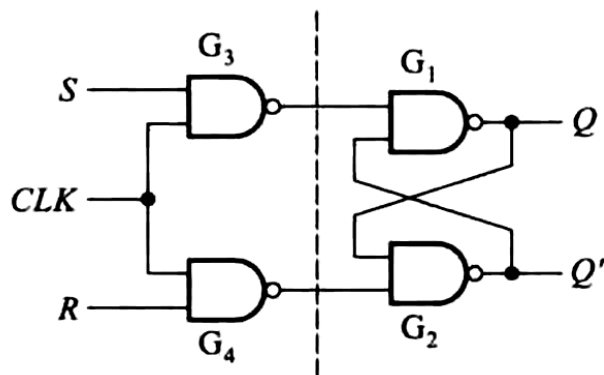
S'	R'	Q^*	Q'^*
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q	Q'

SR触发器

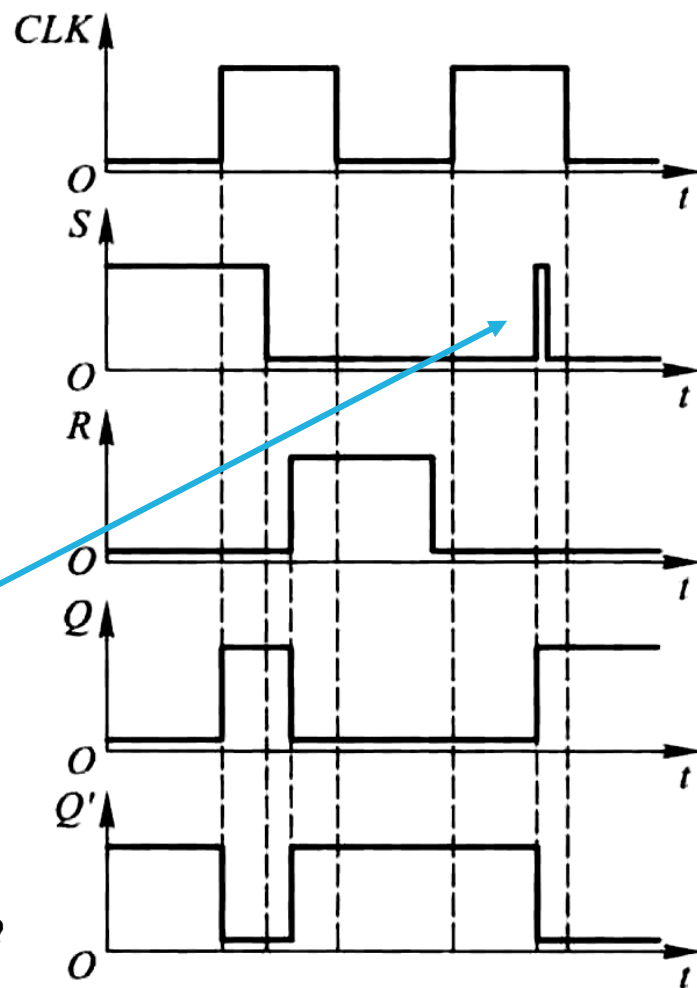
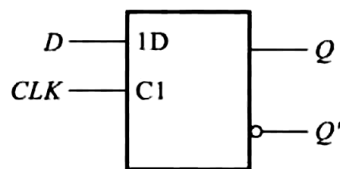
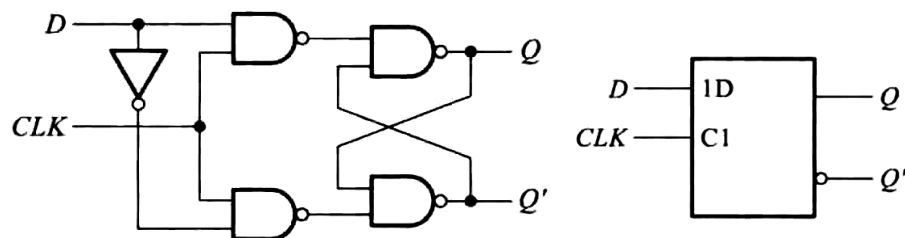


CLK	S	R	Q^*	Q'^*	
0	x	x	Q	Q'	锁存状态
1	0	0	Q	Q'	
1	0	1	0	1	复位状态
1	1	0	1	0	置位状态
1	1	1	0	0	非法状态

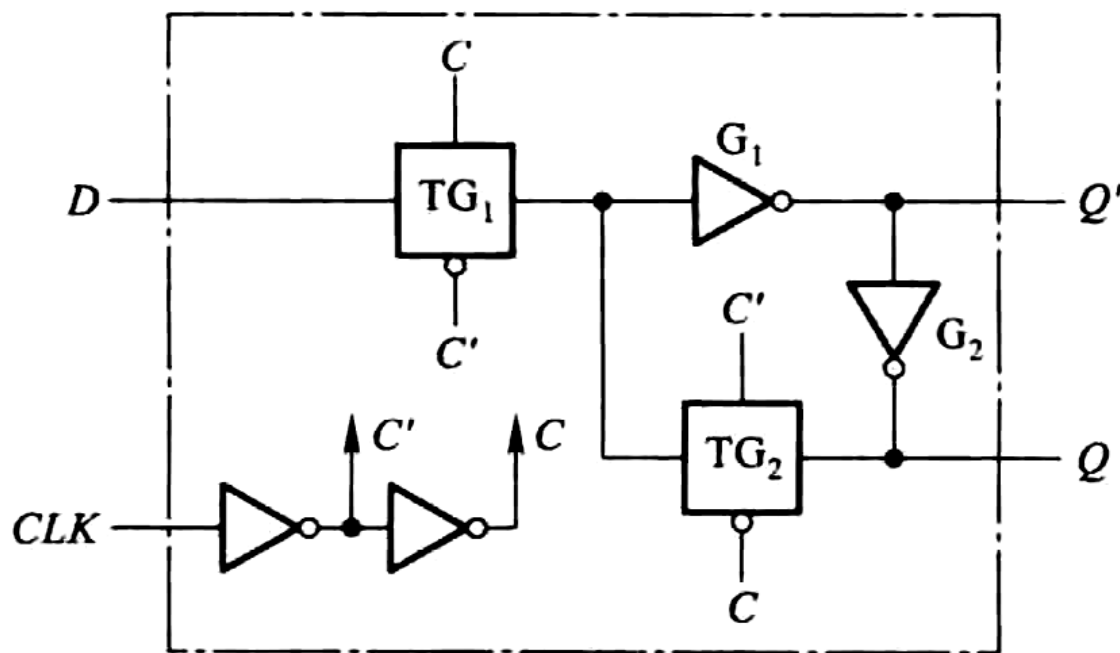
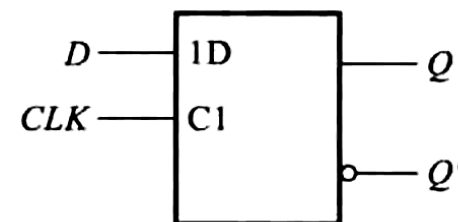
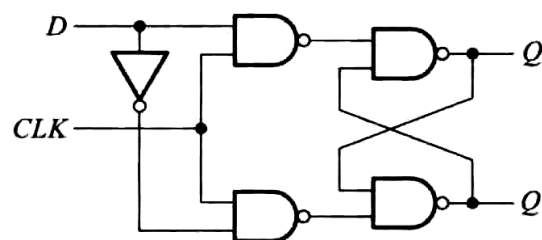
SR触发器的使用



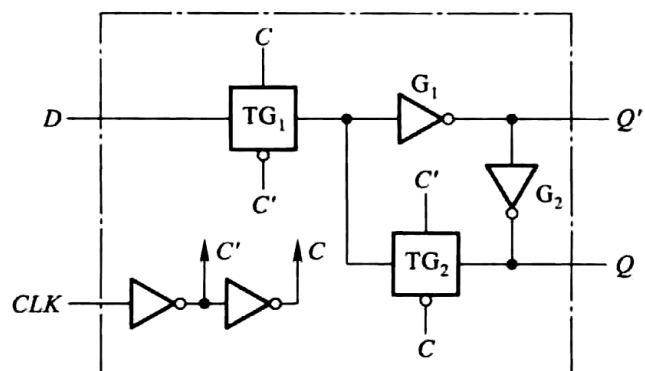
- 在 $CLK=0$ 的时候保持状态
- 在 $CLK=1$ 的时候，接受输入
- 但是可能被毛刺触发



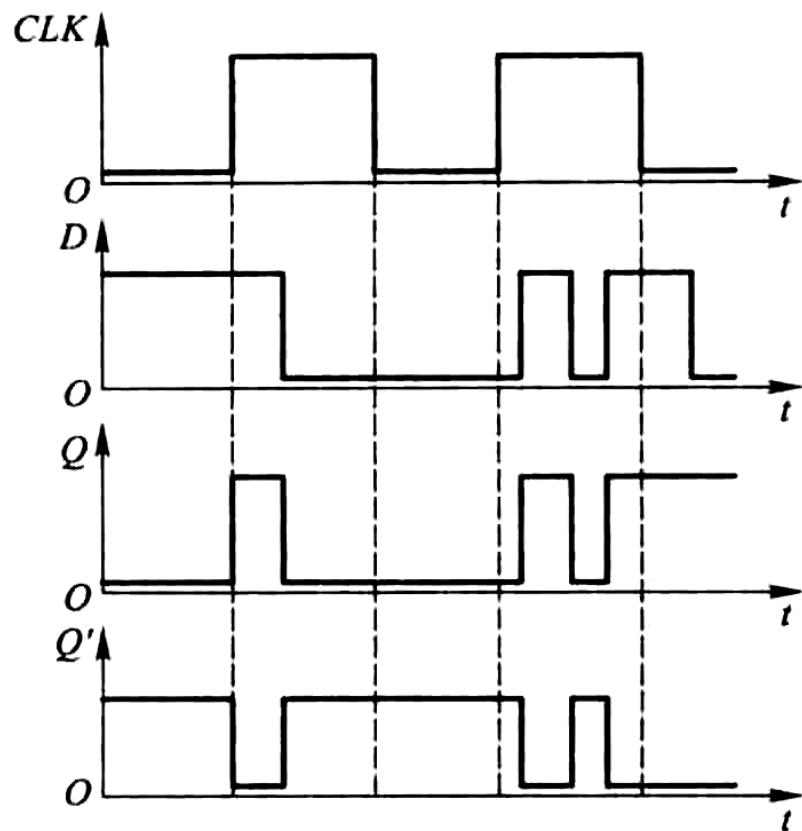
D触发器的传输门实现



D触发器的使用



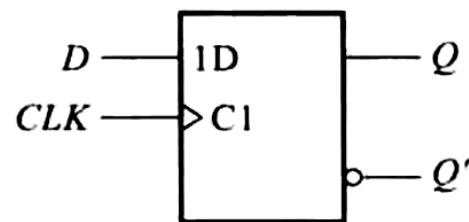
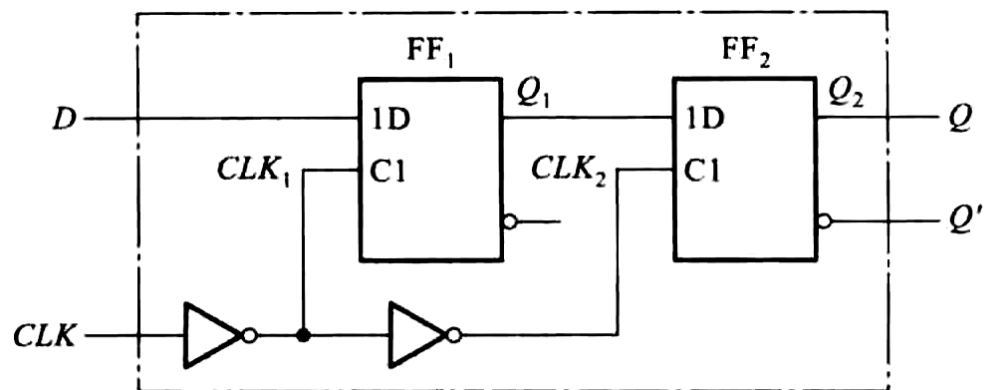
- 在 $CLK=0$ 的时候保持状态
- 在 $CLK=1$ 的时候，接受输入



电平触发的触发器总结

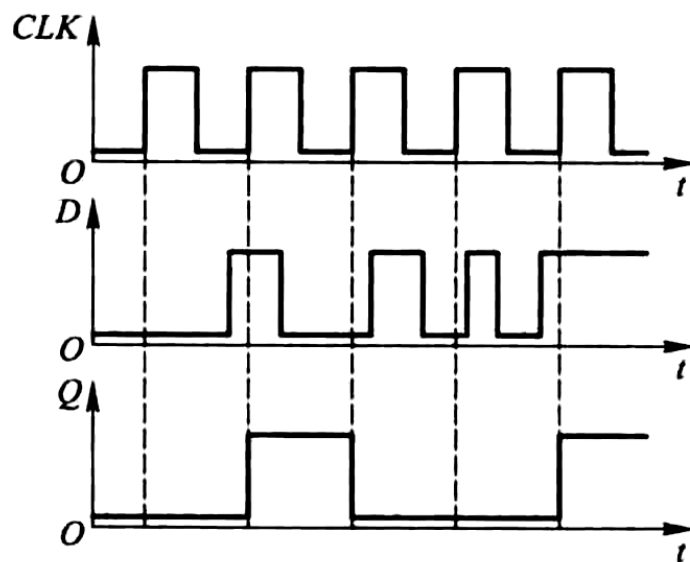
- 触发器：
 - 有触发信号的状态保存器
 - 触发信号一般为时钟信号 (clock, CLK)
- 电平触发：
 - 当触发信号有效时 (一般为高电平), 输出受输入控制
 - 当触发信号复位时, 输出保留有效时最后的输出状态
- RS锁存器：
 - 基于或非门或与非门的RS锁存器结构
- RS触发器：
 - 带触发端的RS锁存器
- D触发器和T触发器：
 - 利用输入反门迫使 $R = \bar{S}$ 的RS触发器
- 利用传输门搭建的D触发器
 - 现在的大规模集成电路多使用该方式实现D触发器

沿触发D触发器

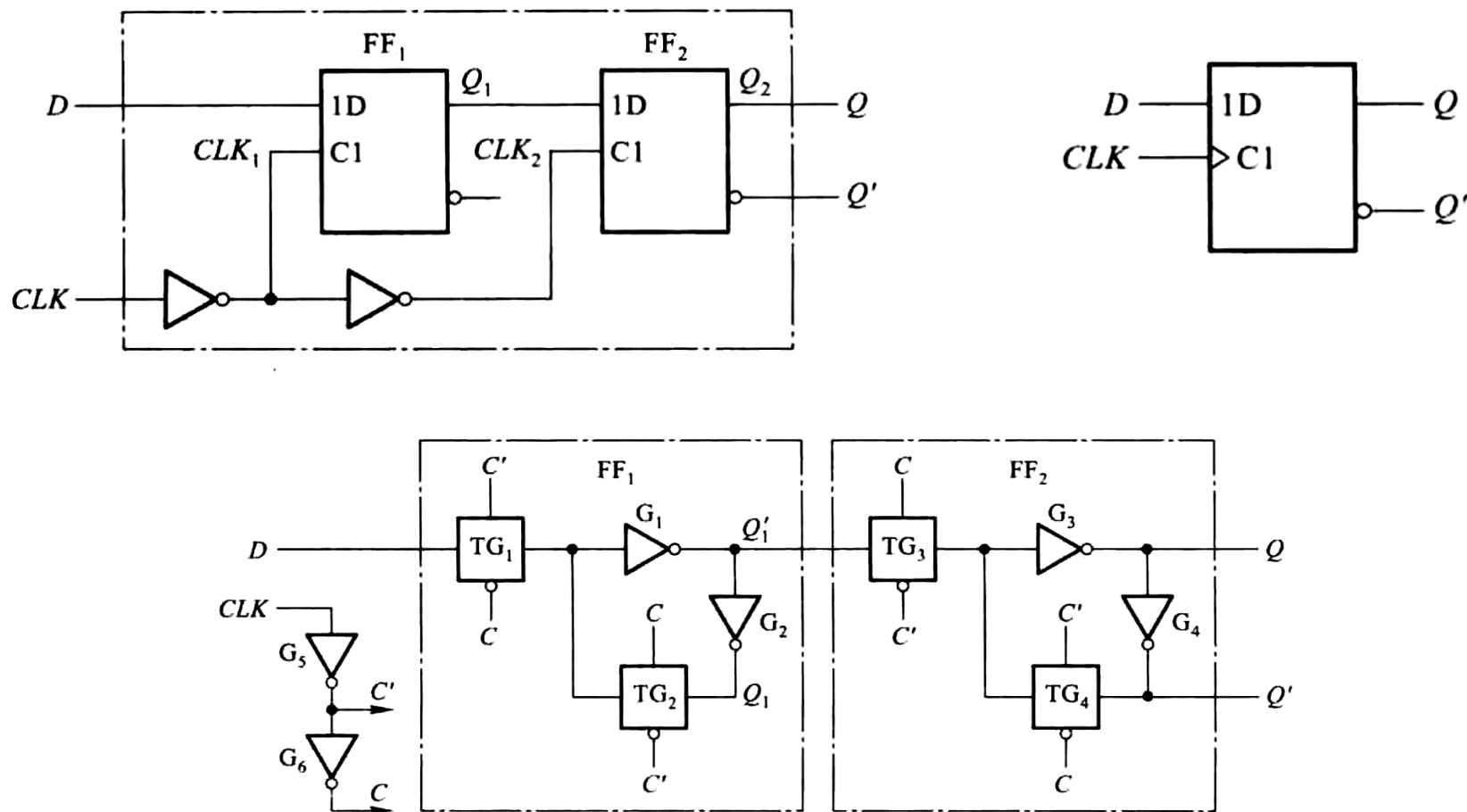


CLK	D	Q^*	Q'^*
x	x	Q	Q'
$\uparrow$	0	0	1
$\uparrow$	1	1	0

滤去了时钟信号为高电平时的组合逻辑噪声（毛刺）。回想在处理组合逻辑竞争冒险时使用的选通脉冲方法。这也是同步时序逻辑处理组合逻辑竞争冒险的主要方式。

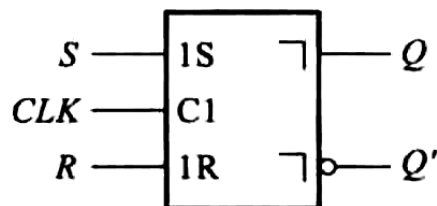
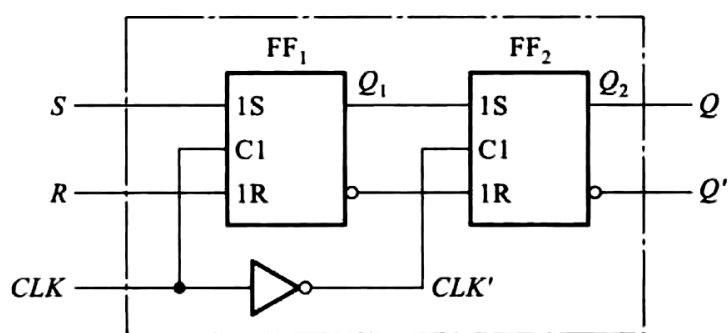


沿触发D触发器的实现



沿触发的D触发器是现代大规模集成电路的常用触发器，也被称为DFF（D flip-flop），其实现方式即为上图所示的传输门实现。

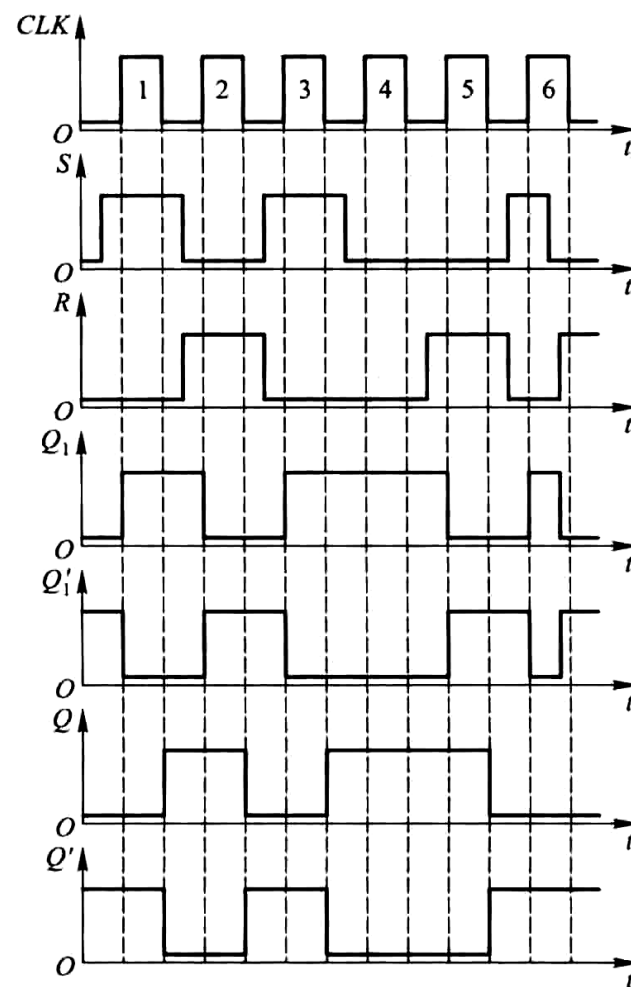
脉冲触发SR触发器



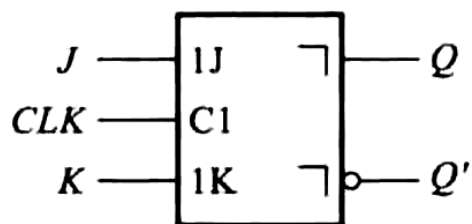
CLK	S	R	Q^*	Q'^*
X	X	X	Q	Q'
$\downarrow$	0	0	Q	Q'
$\downarrow$	1	0	1	0
$\downarrow$	0	1	0	1
$\downarrow$	1	1	1	1

非法状态

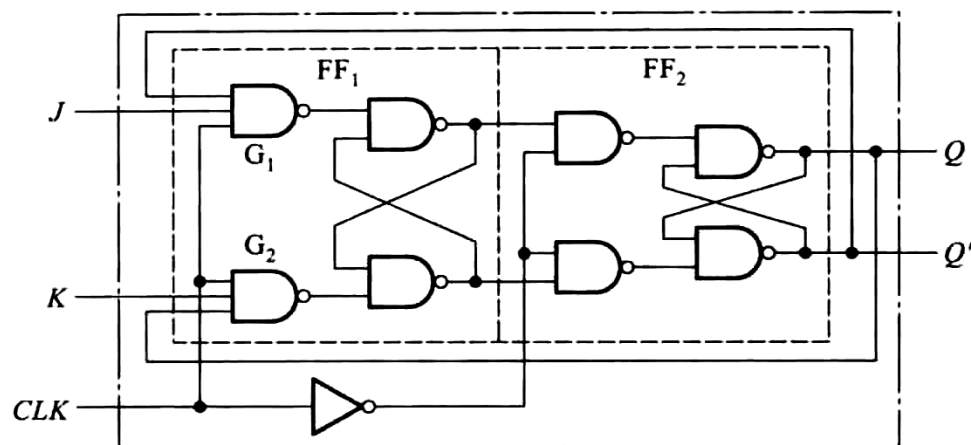
注意CLK的连接，和DFF相反，实际上是下沿触发。假设CLK的高电平时间较短，高电平期间S/R没有变化，所以也称为脉冲触发。由于本质是SR触发器，仍存在非法状态。



脉冲触发JK触发器

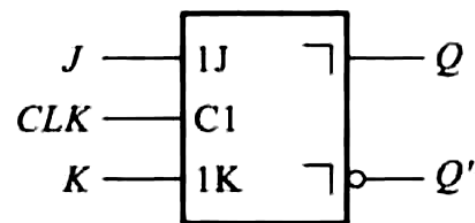
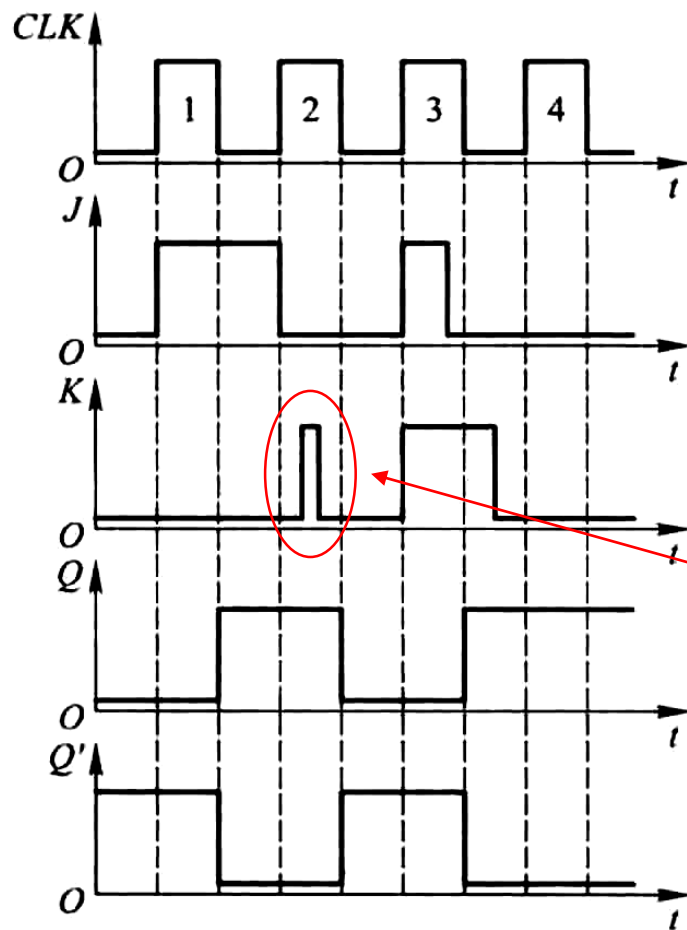


CLK	J	K	Q^*	Q'^*
X	X	X	Q	Q'
$\downarrow$	0	0	Q	Q'
$\downarrow$	1	0	1	0
$\downarrow$	0	1	0	1
$\downarrow$	1	1	Q'	Q



JK触发器将输出反馈到输入端，当J和K输入均为1时，输出端驱动输入，导致输出状态翻转，形成一个T (toggle) 触发器。其他输入时和脉冲触发的SR触发器一致。去除了SR触发器的非法状态。

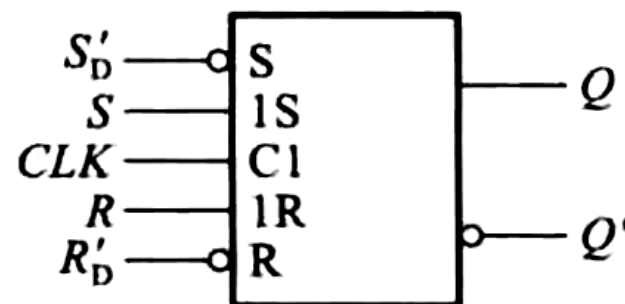
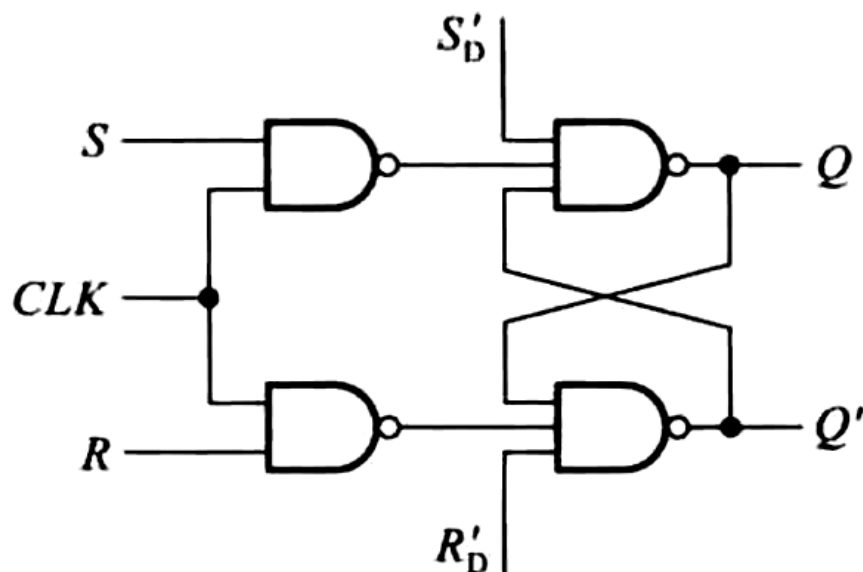
脉冲触发JK触发器的波形



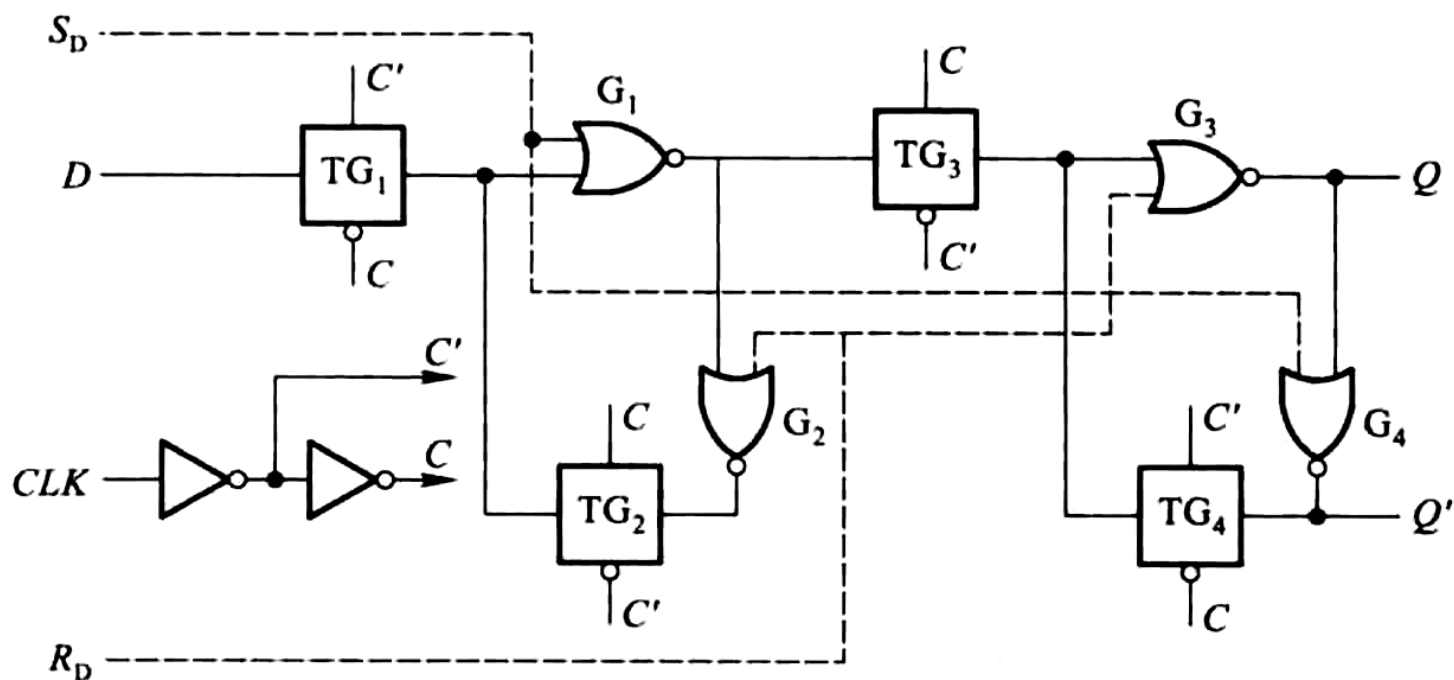
和沿触发的触发器不同，
CLK=1期间的脉冲仍然能改变输出状态。

异步复位/置位

- 受时钟控制的信号为同步信号
- 不受时钟控制的信号为异步信号
- 不受时钟控制的行为为异步行为
- 异步复位/置位：不受时钟影响，随时复位/置位



带异步复位的沿触发D触发器



触发器总结

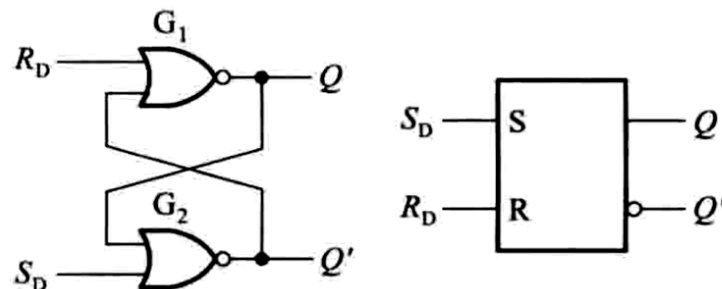
- RS触发器: $Q^* = S + \bar{R}Q$
 - 电平触发, 沿触发, 脉冲触发
- D触发器: $Q^* = D$
 - 迫使 $S = \bar{R}$
 - 电平触发, 沿触发, 脉冲触发
- JK触发器: $Q^* = J\bar{Q} + \bar{K}Q$
 - 当RS触发器处于非法状态时, 将输出反接至输入, 倒置输出翻转
 - 电平触发, 沿触发, 脉冲触发
- T触发器: $Q^* = \bar{Q}$
 - JK触发器的特殊状态, 输出在每次触发后翻转

RS锁存器的Verilog HDL表述

○RS锁存器

```
reg q, qb;  
always @(r, s) begin  
    if(s == 1) begin  
        qb = 0;  
        if(r == 0) q = 1;  
    end  
    if(r == 1) begin  
        q = 0;  
        if(s == 0) qb = 1;  
    end  
    // 当s == 0, r == 0 时, q 和 qb 保持原值, 不用描述  
end
```

并列if应该互斥

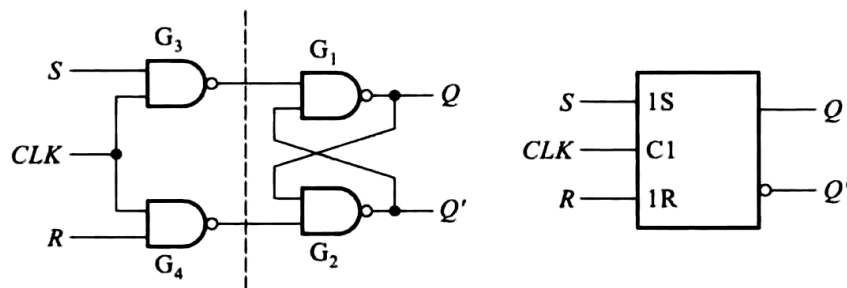


实际电路中RS锁存器并不常见，这里只是指出，在always块中，如果变量不被赋值，则保持原值。

电平触发的RS触发器的Verilog HDL表述

○高电平触发的RS触发器

```
reg q, qb;  
always @(clk, r, s) begin  
    if(clk == 1) begin  
        if(s == 1) begin  
            qb = 0;  
            if(r == 0) q = 1;  
        end  
        if(r == 1) begin  
            q = 0;  
            if(s == 0) qb = 1;  
        end  
    end  
end  
end
```

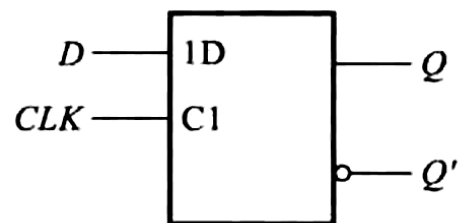
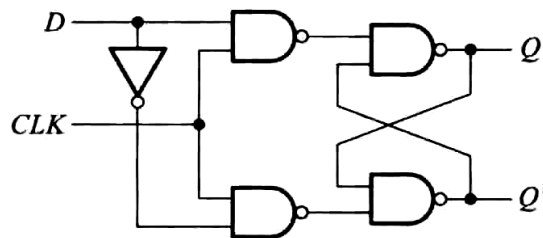


增加了clk信号，只有在clk==1的时候才改变输出状态。

电平触发的D触发器的Verilog HDL表述

○高电平触发的D触发器

```
reg q, qb;  
always @(clk, d) begin  
    if(clk == 1) begin  
        q = d;  
        qb = ~d;  
    end  
end
```



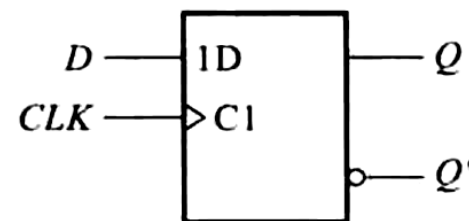
反而D触发器的Verilog描述非常简单。

其实这也是大规模集成电路当中常见的电路，一般我们称其为 Latch (锁存器，注意术语区别)。

沿触发的D触发器的Verilog HDL表述

○上升沿触发的D触发器

```
reg q, qb;  
always @(posedge clk) begin  
    q <= d;  
    qb <= ~d;  
end
```



注意这里的posedge和<=赋值，同时clk不出现在if的条件判断里。

大规模集成电路当中最常见的触发器形式，一般我们称其为register（寄存器）或FF (Flip-flop，注意术语区别)。

阻塞和非阻塞赋值

○阻塞赋值：普通意义上的赋值

$q = d;$

$qb = \sim d;$

$q = d;$

$qb = \sim q;$

等效

不等效

$qb = \sim q;$

$q = d;$

○非阻塞赋值：赋值给 Q^*

$q \leq d;$

$qb \leq \sim d;$

$q \leq d;$

$qb \leq \sim q;$

不等效

等效

$qb \leq \sim q;$

$q \leq d;$

$Q^* = \bar{Q}$

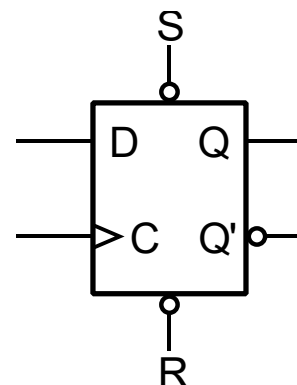
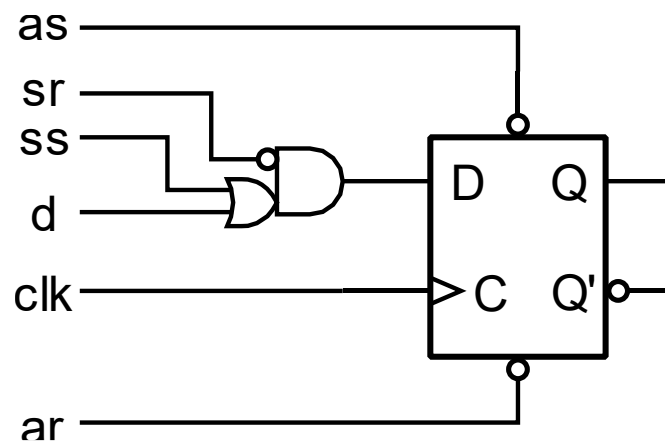
非阻塞赋值只出现在always块中，非阻塞赋值语句与顺序无关，更合乎硬件思维。

同步和异步复位/置位

带同步异步复位的DFF

```
reg q, qb;  
wire d; // 输入信号  
wire clk; // 时钟  
wire as, ar; // 异步置位/复位  
wire ss, sr; // 同步置位/复位
```

```
always @(posedge clk or negedge as or negedge ar) begin  
    if(~ar)      begin q <= 0; qb <= 1; end  
    else if(~as) begin q <= 1; qb <= 0; end  
    else begin  
        if(sr)      begin q <= 0; qb <= 1; end  
        else if(ss) begin q <= 1; qb <= 0; end  
        else        begin q <= d; qb <= ~d; end  
    end  
end
```



沿触发的T触发器

○带异步复位的沿触发T触发器 利用沿触发的DFF搭建。

```
reg q, qb;
```

```
always @(posedge clk or negedge rst) begin
```

```
    if(~rst) begin
```

```
        q <= 0;
```

```
        qb <= 1;
```

```
    end else begin
```

```
        if(t) begin // 如果t为高, 翻转输出状态
```

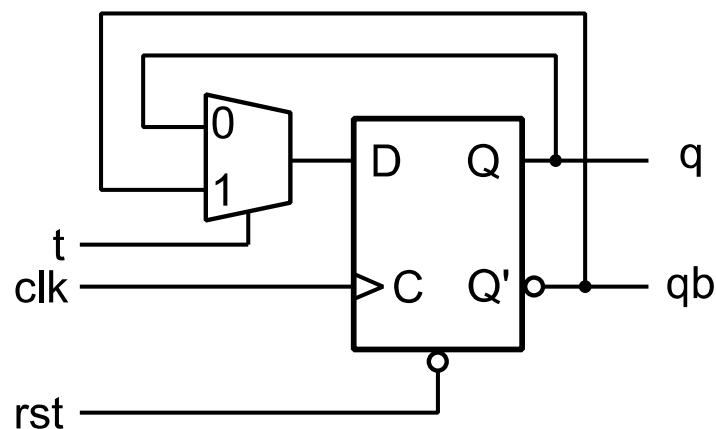
```
            q <= qb;
```

```
            qb <= q;
```

```
        end
```

```
    end
```

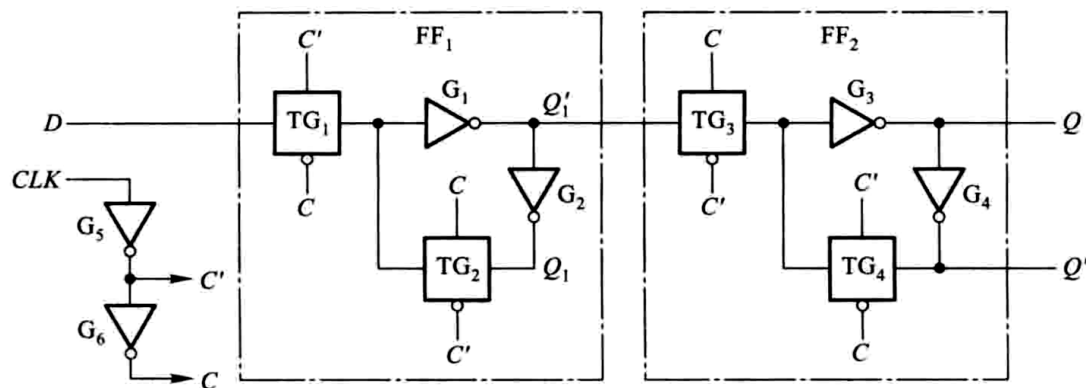
```
end
```



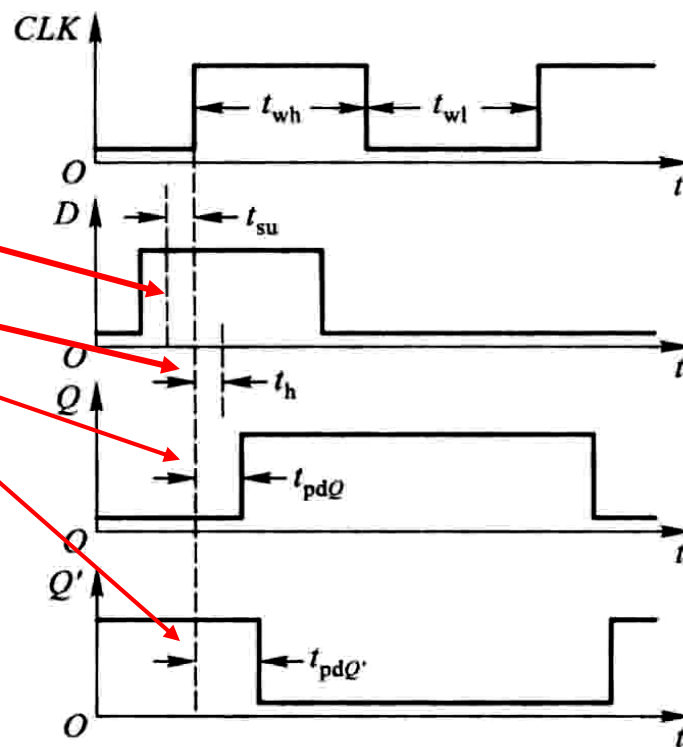
触发器Verilog HDL描述总结

- Verilog HDL描述D触发器
 - 电平触发的D触发器 (latch)
 - 沿触发的D触发器(flip-flop)
- 阻塞和非阻塞赋值
- 同步和异步复位/置位
- 用DFF构造T触发器
- Verilog的新语法：
 <=, posedge, negedge

沿触发的D触发器(DFF)的动态特性

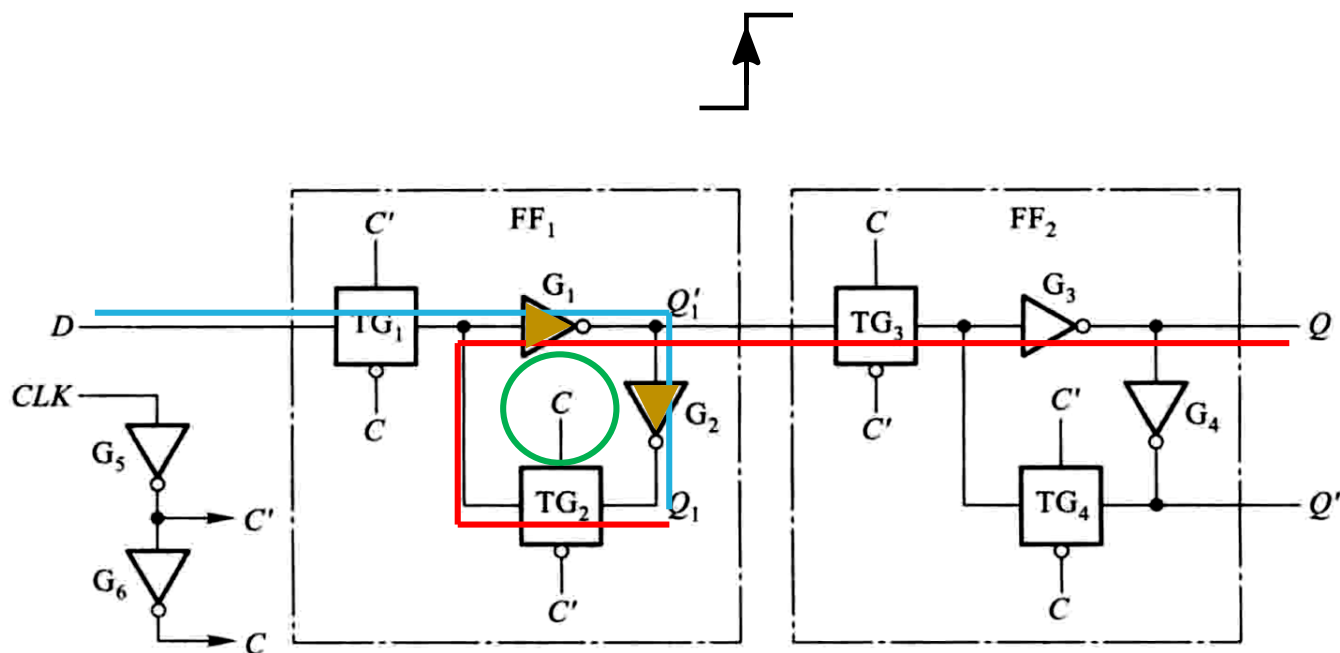


- 建立时间 (setup time) t_{su}
- 保持时间 (hold time) t_h
- 传输延迟 (propagation delay) t_{pd}



DFF的建立时间

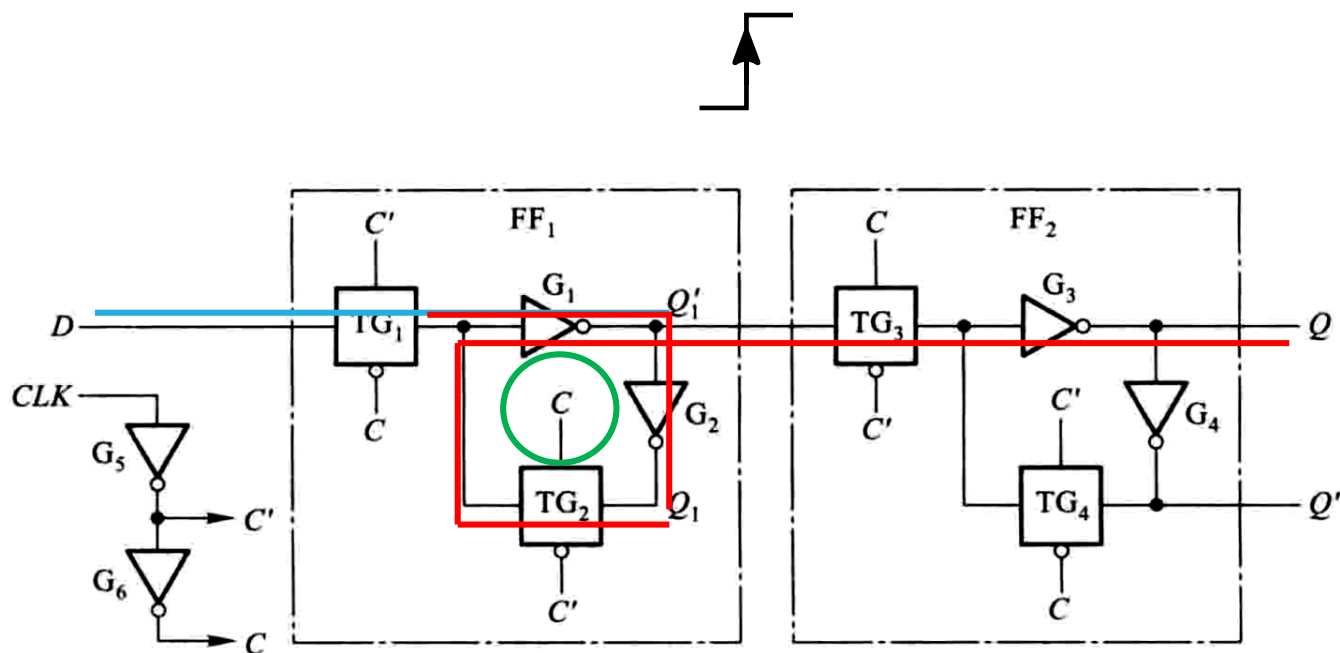
建立时间 t_{su} 是时钟沿到来之前数据输入需要保持稳定的最小时间。



在时钟沿到来之前，输入信号 D 应该已经传输到 Q_1 ，否则输入级的反馈环不能正确建立。所以， $t_{su} \approx t_{TG1} + t_{G1} + t_{G2}$

DFF的保持时间

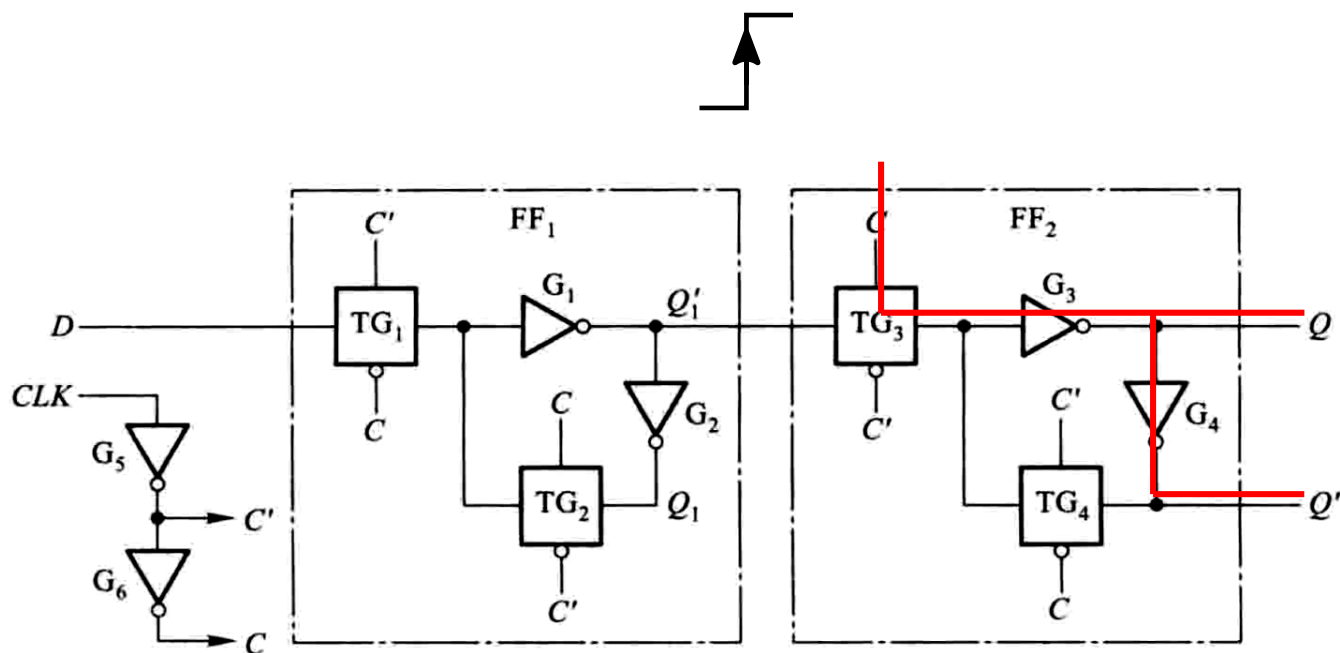
保持时间 t_h 是时钟沿到来之后数据输入需要保持稳定的最小时间。



在时钟沿到来之后，输入信号 D 应该在 TG_1 彻底关闭之前保持稳定。所以， $t_h \approx t_{TG1}$ 。（这个和课本上有出入，欢迎讨论）

DFF的传输时间

传输时间 t_{pd} 是从时钟沿到输出状态变化的最小时间。



在时钟沿到来之后:

$$t_{pd} \approx t_{TG3} + t_{G3}$$
$$t'_{pd} \approx t_{TG3} + t_{G3} + t_{G4}$$

总结：考试范围

- 各种触发器：逻辑、实现、特性
RS触发器，D触发器，JK触发器，T触发器
- Verilog HDL：设计
 - D-Latch：D型电平触发器
 - DFF：D型沿触发器
 - 由DFF构成T触发器
- DFF的动态特性（概念）
 - 建立时间
 - 保持时间
 - 传输时间

任何问题?

第五章习题：

4, 5, 8, 9, 15, 17, 19, 20, 21。

○用Verilog HDL描述第五章习题P5.20的电路

(其中的 R'_D 为异步复位信号)

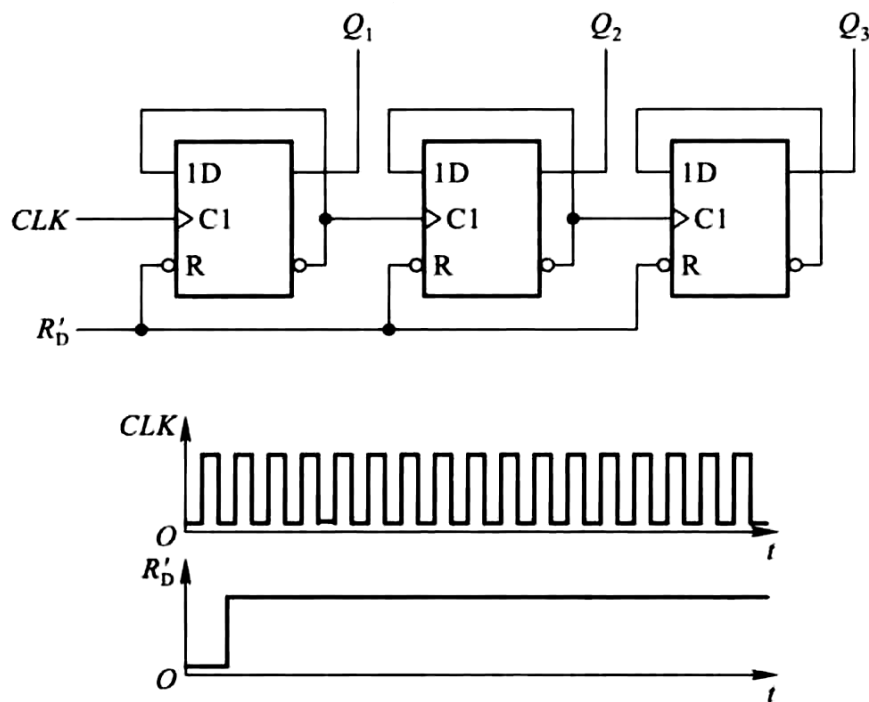


图 P5. 20