

数字逻辑设计实验

实验10、锁存器与触发器基本原理

2023年11月



提 纲

- 实验目的
- 实验设备与材料
- 实验任务
- 实验原理
- 实验内容与步骤



实验目的

- 掌握锁存器与触发器构成的条件和工作原理
- 掌握锁存器与触发器的区别
- 掌握基本SR锁存器、门控SR锁存器、D锁存器、SR锁存器、D触发器的基本功能
- 掌握基本SR锁存器、门控SR锁存器、D锁存器、SR锁存器存在的时序问题



实验设备与材料

□ 实验设备

- 装有Xilinx ISE 14.7的计算机 1台
- SWORD开发板 1套

□ 实验材料

- 无



实验任务

- 实现基本SR锁存器，验证功能和存在的时序问题
- 实现门控SR锁存器，并验证功能和存在的时序问题
- 实现D锁存器，并验证功能和存在的时序问题
- 实现SR主从触发器，并验证功能和存在的时序问题
- 实现D触发器，并验证功能



实验原理

□ 构成锁存器的充分条件

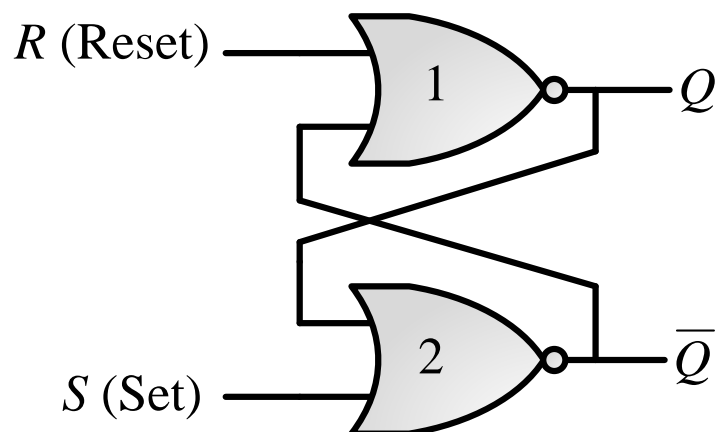
- 能长期保持给定的某个稳定状态
- 有两个稳定状态：0、1
- 在一定条件下能随时改变逻辑状态，即：置1或置0

□ 最基本的锁存器有：SR锁存器、D锁存器

□ 锁存器有两个稳定状态，又称双稳态电路

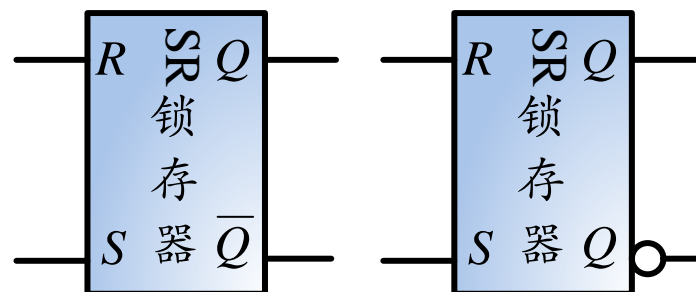
SR锁存器 (1)

- 将两个具有2输入端的反向逻辑器件的输出与输入端交叉连起来，另一个输入端作为外部信息输出端，就构成最简单的**SR锁存器**

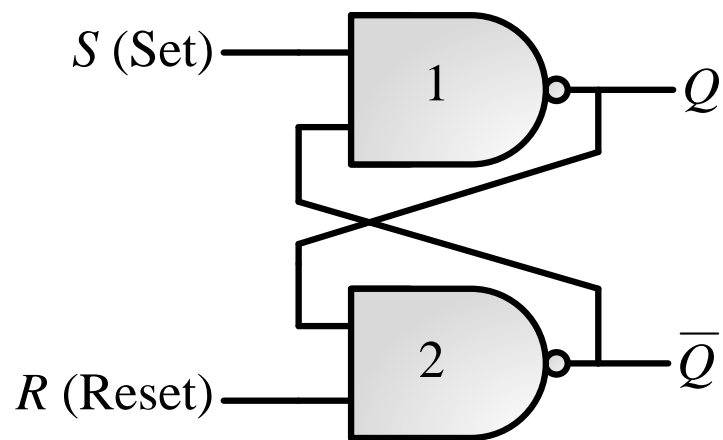


或非门

$R S$	$Q \bar{Q}$	说明
0 0	$Q \bar{Q}$	保持
0 1	1 0	置1
1 0	0 1	置0
1 1	0 0	未定义

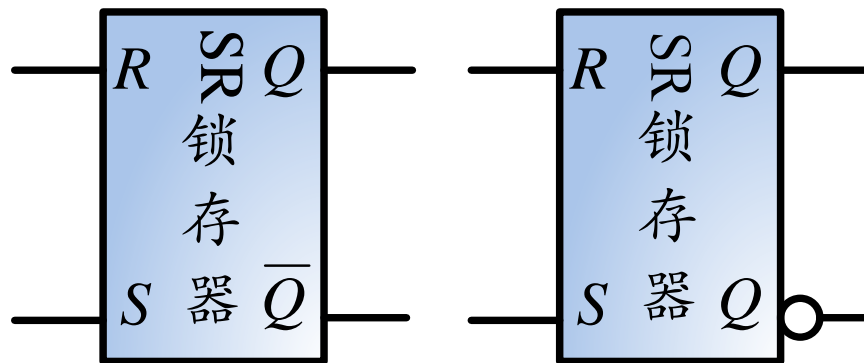


SR锁存器 (2)



与非门

$R S$	$Q \bar{Q}$	说明
0 0	1 1	未定义
0 1	0 1	置0
1 0	1 0	置1
1 1	$Q \bar{Q}$	保持



仿真

R=1;S=1; #50;

R=1;S=0; #50;

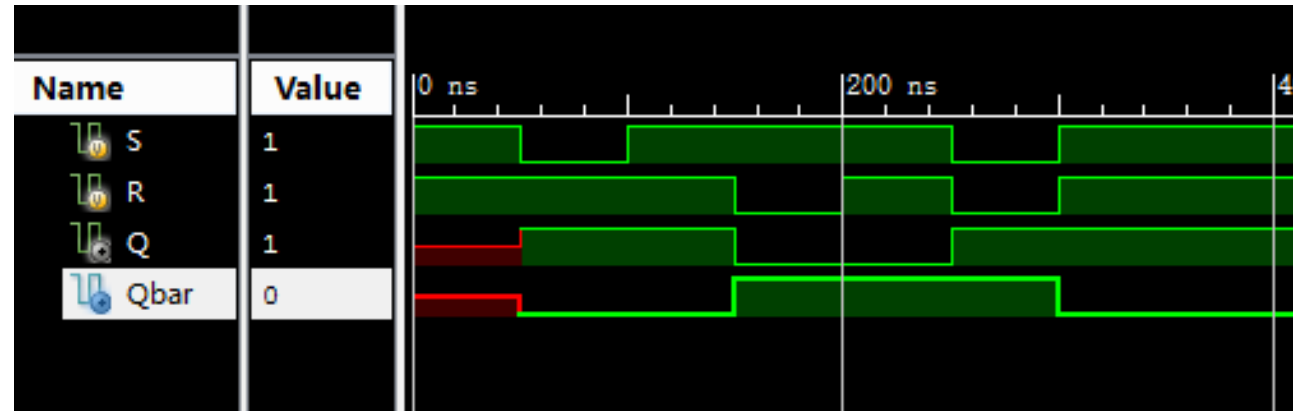
R=1;S=1; #50;

R=0;S=1; #50;

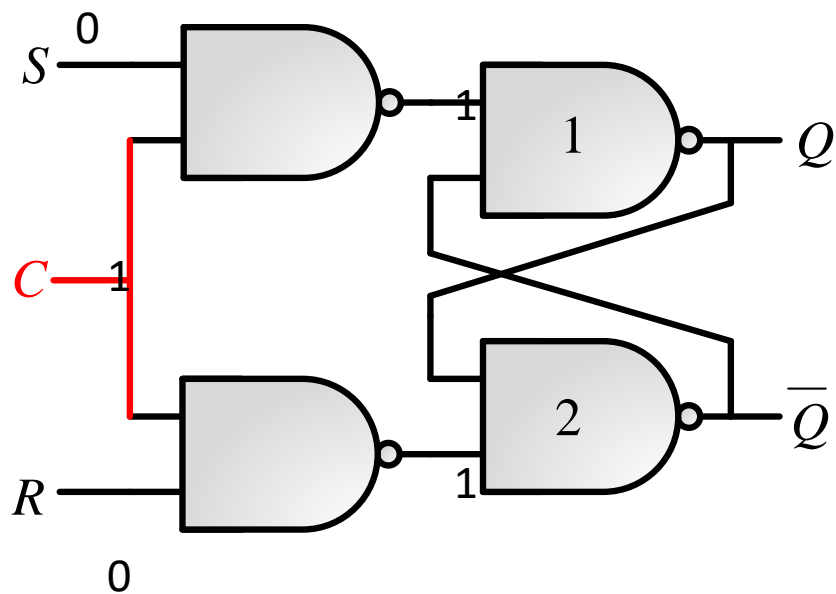
R=1;S=1; #50;

R=0;S=0; #50;

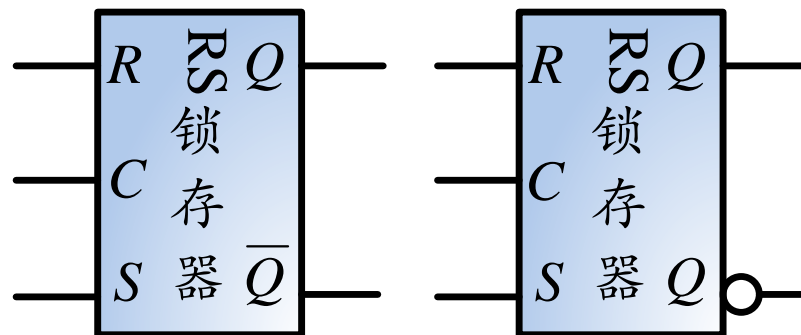
R=1;S=1; #50;



门控SR锁存器



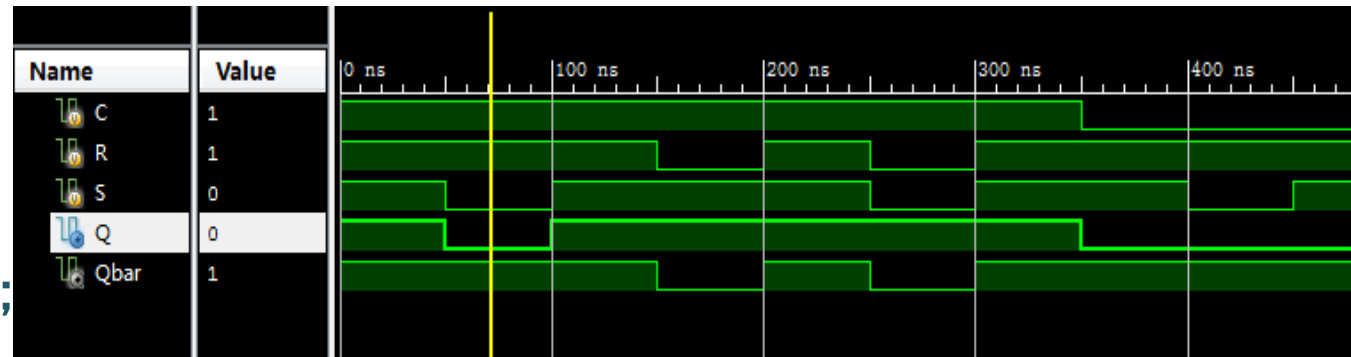
$C R S$	$Q \bar{Q}$	说明
$0 \times \times$	$Q \bar{Q}$	保持
$1 0 0$	$Q \bar{Q}$	保持
$1 0 1$	$1 0$	置1
$1 1 0$	$0 1$	置0
$1 1 1$	$1 1$	未定义





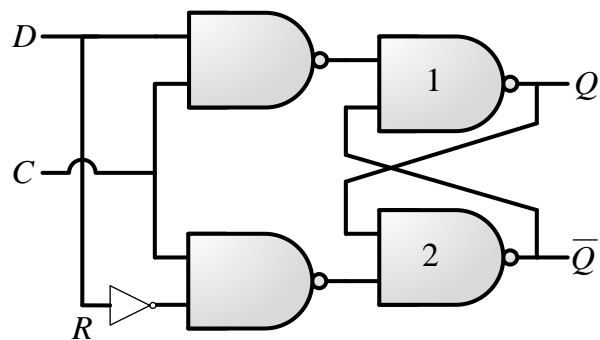
仿真

```
C=1;R=1;S=1; #50;  
R=1;S=0; #50;  
R=1;S=1; #50;  
R=0;S=1; #50;  
R=1;S=1; #50;  
R=0;S=0; #50;  
R=1;S=1; #50;  
C=0;R=1;S=1; #50;  
R=1;S=0; #50;  
R=1;S=1; #50;  
R=0;S=1; #50;  
R=1;S=1; #50;  
R=0;S=0; #50;  
R=1;S=1; #50;
```

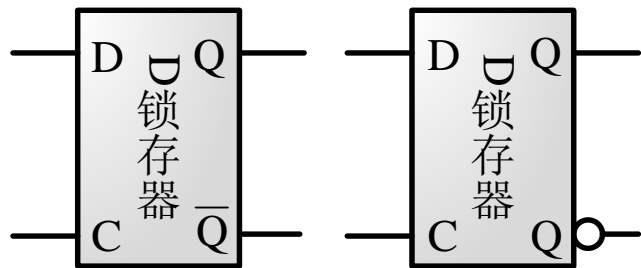


D锁存器

- 基本SR锁存器缺点：存在不确定状态
- 解决方法：消除不确定状态



$C D$	$Q \bar{Q}$	说明
$0 \times$	$Q \bar{Q}$	保持
$1 0$	$0 1$	置0
$1 1$	$1 0$	置1



- 只需1个数据输入端 D
- 输出端Q等于输入端D
- 采用电平控制 C

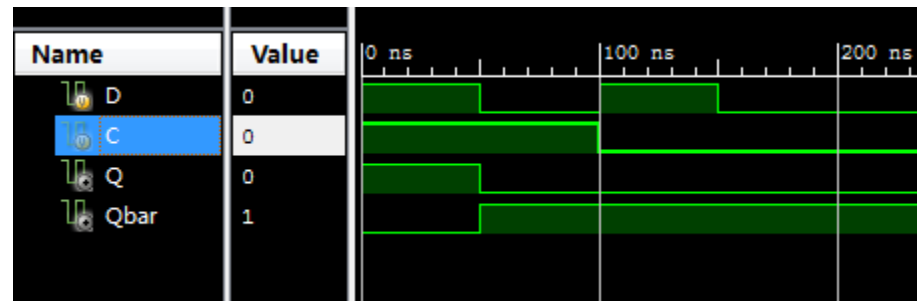
仿真

C=1;D=1; #50;

D=0; #50;

C=0;D=1; #50;

D=0



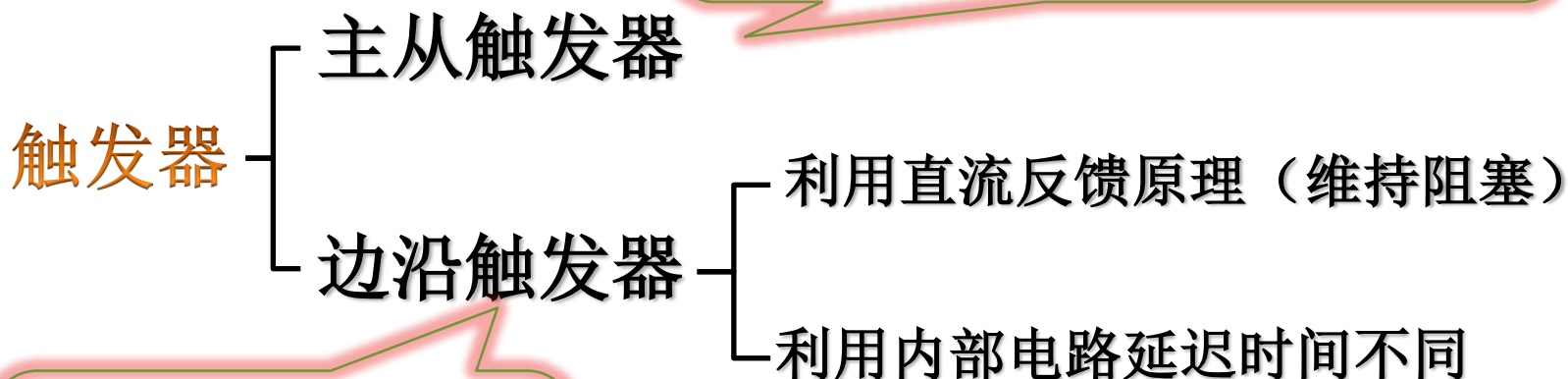


触发器

- D锁存器的缺点：存在**空翻**现象——如果D锁存器直接用在时序电路中作为状态存储元件，当使能控制信号有效时，会导致该元件内部的状态值随时多次改变，而不是保持所需的原始状态值
- 解决方法：消除空翻现象，使每次触发仅使锁存器的内部状态仅改变一次
- **触发**：外部输入使锁存器状态改变的瞬间状态
- **触发器**：在锁存器的基础上使每次触发仅使状态改变一次的锁存电路（双稳态）

触发器

用两个锁存器，主锁存器在脉冲控制下接收输入数据，从锁存器在脉冲结束后改变并保持状态。

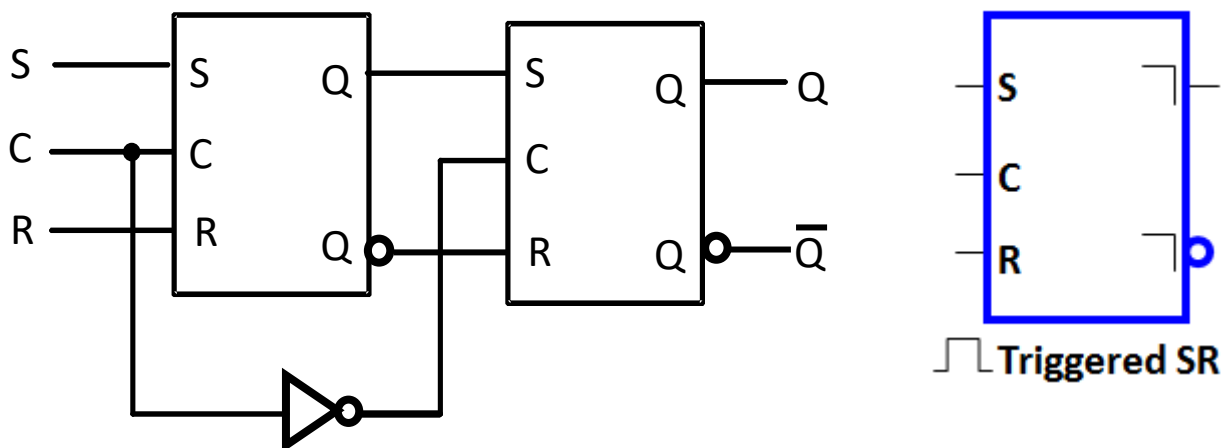


利用时钟上升沿或下降沿变换状态，其他时间保持状态

常见的触发器有：主从SR触发器、D触发器、JK触发器、T

SR主从触发器

- 由两个门控S-R锁存器串联构成，第二个锁存器的时钟通过反相器取反
- 当 $C=1$ 时，输入信号进入第一个锁存器（主锁存器）
- 当 $C=0$ 时，第二个锁存器（从锁存器）改变输出
- 从输入到输出的通路被不同的时钟信号值 ($C = 1$ 和 $C = 0$) 所断开



仿真



initial begin

R=1;S=1; #50;

R=1;S=0; #50;

R=1;S=1; #50;

R=0;S=1; #50;

R=1;S=1; #50;

R=0;S=0; #50;

R=1;S=1; #50;

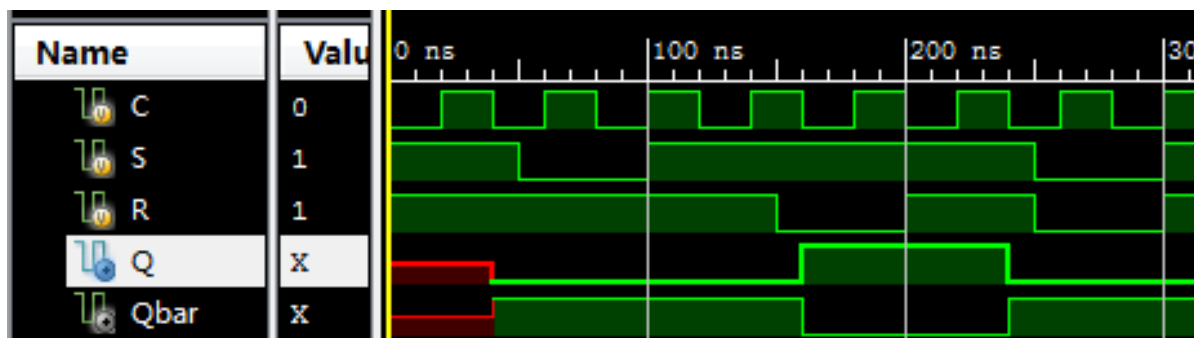
end

always begin

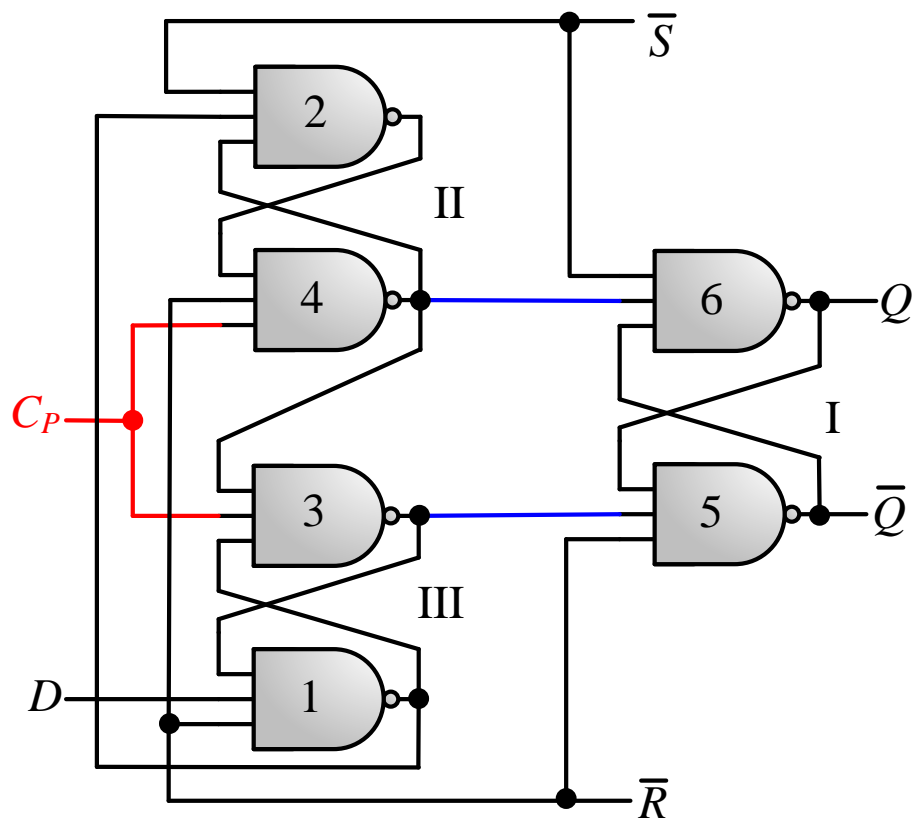
C=0;#20;

C=1;#20;

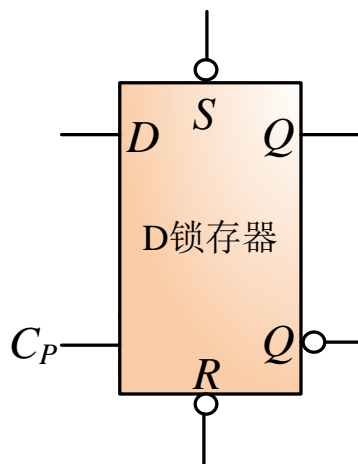
end



正边沿维持阻塞型D触发器



异步控制		上升沿触发			
R	S	C_P	D	Q	\bar{Q}
0	1	×	×	0	1
1	0	×	×	1	0
1	1	↑	0	0	1
1	1	↑	1	1	0



仿真

```
initial begin
```

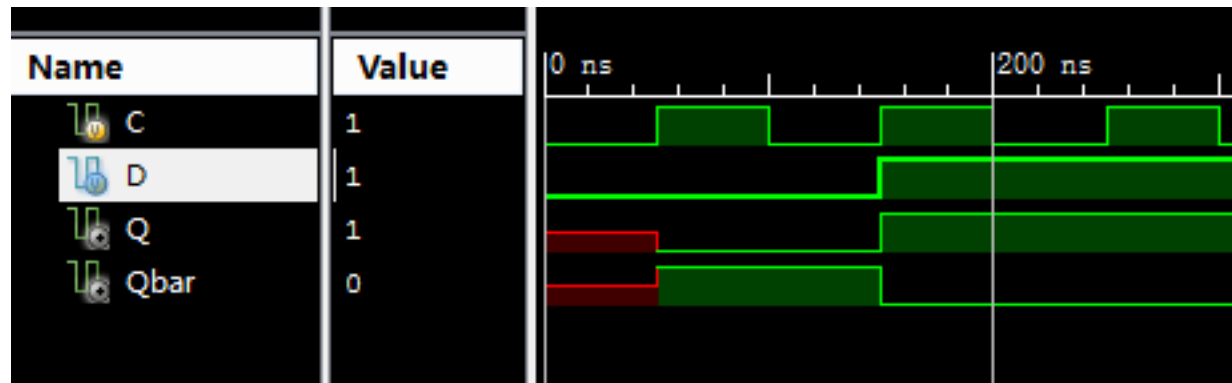
```
    S = 1;
```

```
    R = 1;
```

```
    D = 0; #150;
```

```
    D = 1; #150;
```

```
end
```



```
always begin
```

```
    C=0; #50;
```

```
    C=1; #50;
```

```
end
```

还能试试S=1,R=0和
S=0,R=1的情况



实验内容与步骤

- 实现基本SR锁存器，验证功能和存在的时序问题
- 实现门控SR锁存器，并验证功能和存在的时序问题
- 实现D锁存器，并验证功能和存在的时序问题
- 实现SR主从触发器，并验证功能和存在的时序问题
- 实现D触发器，并验证功能



实验内容与步骤

- 寄存器锁存器均可使用verilog代码来结构化实现，
由于代码较为简单，这里仅展示基本SR锁存器

```
module SR(  
    input wire S,R,  
    output wire Q,InvQ  
);  
    nand m0(Q,S,InvQ);  
    nand m1(InvQ,Q,R);  
  
endmodule
```



基本SR锁存器

- 新建工程MyLATCHS
- 新建源文件SR_LATCH. sch
- 用原理图方式设计
- 用NAND2实现

- 仿真



门控SR锁存器

- 新建源文件CSR_LATCH. sch
- 用原理图方式设计
- 用NAND2实现
- 仿真
- 生成自定义符号的CSR_LATCH. sym



D锁存器

- 新建源文件D_LATCH. sch
- 用原理图方式设计
- 用NAND2实现

- 仿真



SR主从触发器

- 新建源文件MS_FLIPFLOP. sch
- 用原理图方式设计
- 调用CSR_LATCH实现

- 仿真（包含一次性采样）



D触发器

- 新建源文件D_FLIPFLOP. sch
- 用原理图方式设计
- 调用NAND3实现

- 仿真



Thank You !