

# 实验十 锁存器与触发器基本原理

学生姓名：解雲暄

专业：信息安全

学号：3190105871

同组学生姓名：郭长洁

指导老师：蔡铭

实验地点：紫金港东四 - 509

实验日期：2020 年 11 月 30 日

## 1 实验原理

### 1.1 Latch

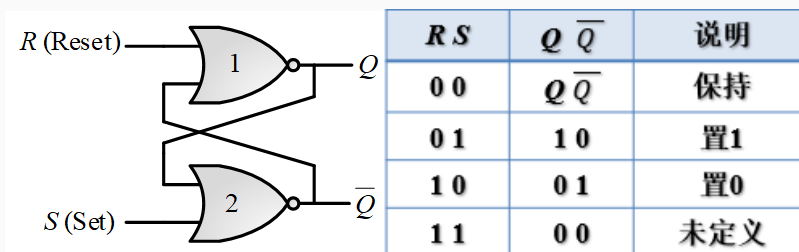
构成锁存器的充分条件有：

- 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。

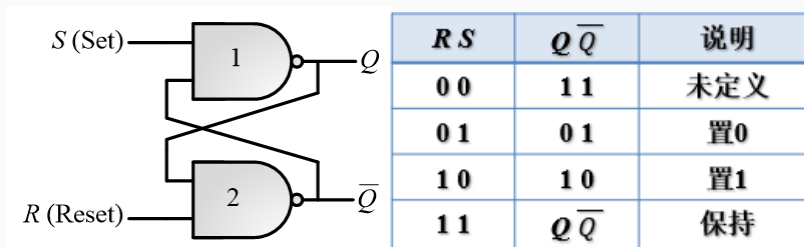
锁存器有两个稳定状态，又称双稳态电路。

### 1.2 The SR-Latch and the Gated SR-Latch

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。我们可以用或非门或与非门实现一个 SR 锁存器：

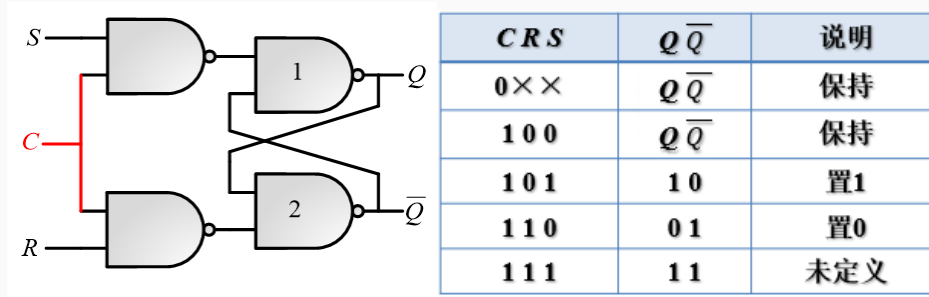


SR 锁存器的或非门实现



SR 锁存器的与非门实现

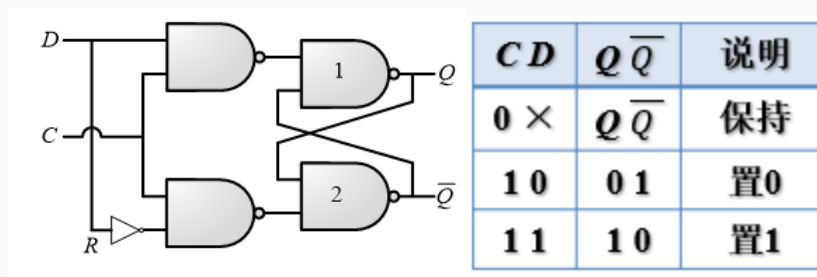
我们可以给 SR 锁存器增加一个使能信号，这样就得到了门控 SR 锁存器：



门控 SR 锁存器

### 1.3 The D-Latch

注意到 SR-Latch 和 Gated SR-Latch 都存在未定义的状态，我们设计方法来消除这种未定义（不确定）的状态：



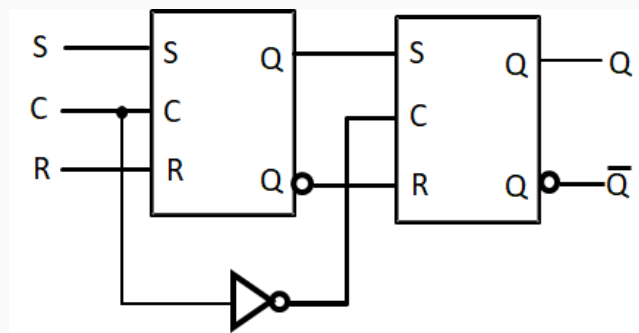
D 锁存器

这样，我们只需要一个数据输入端 D，当门控信号 C 为 0 时，Latch 会保存 D 的内容。

### 1.4 Flip-flop

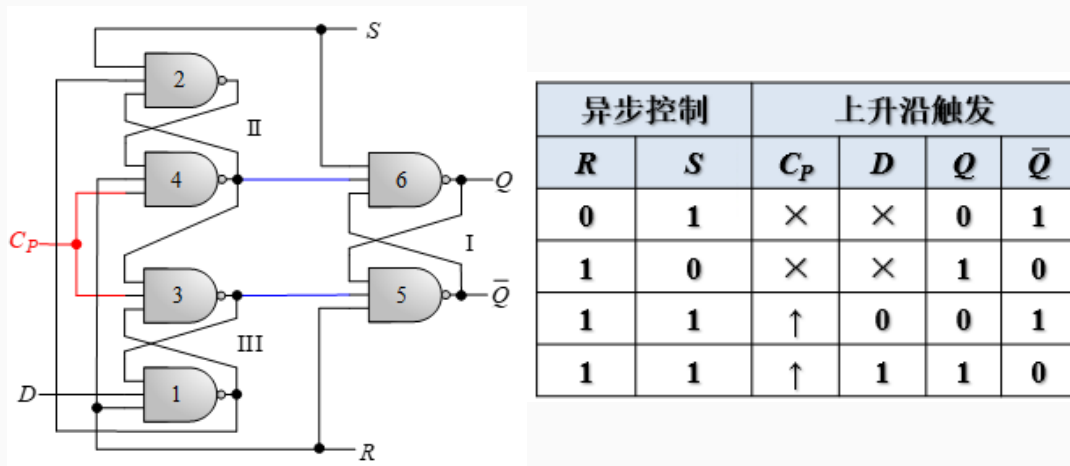
在一个时钟周期内，存储器既要作为组合逻辑电路的输入，又要接受组合逻辑电路的输出。如果存储单元使用 D 锁存器，那么组合逻辑电路的输出会影响锁存器的值，使得这个值被当做输入输给组合逻辑电路，引起问题（因为实际上组合逻辑电路的运行是持续半个时钟周期的循环过程）。

因此，我们需要下图这样的主从触发器（master/slave flip-flop）：在前半个时钟周期开始时，Clock 信号为 1，此时 Master 写入禁止，而 Slave 写入允许。此时 Master 的值进入 Slave 使得 Slave 的值为这个周期中应该使用的值（即当前状态）。后半半个时钟周期开始时，Slave 写入禁止，保持这个周期中应该使用的值，而 Master 写入允许，被替换成下个周期应该使用的值（即下个周期的状态）。



SR 主从触发器

进一步，我们设计正边沿维持阻塞型 D 触发器：



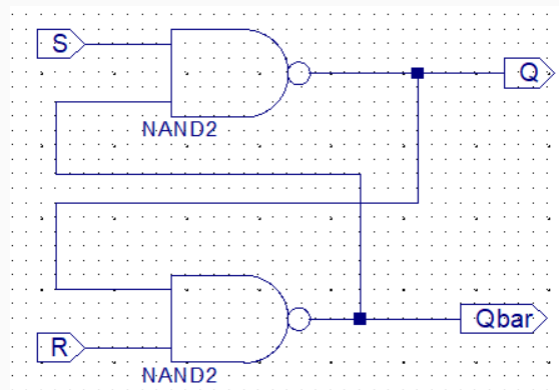
正边沿维持阻塞型 D 触发器

## 2 实验步骤与结果

新建工程 MyLATCHS。

### 2.1 SR-Latch

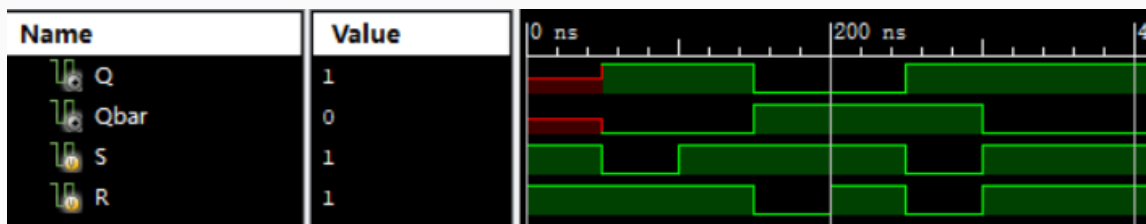
新建 schematic 源文件 SR\_LATCH.sch，绘制原理图如下：



对模块进行仿真，设计激励代码如下：

```
1 module SR_LATCH_SR_LATCH_sch_tb();
2
3     reg S;
4     reg R;
5
6     wire Q;
7     wire Qbar;
8
9     SR_LATCH UUT (
10         .Q(Q),
11         .Qbar(Qbar),
12         .S(S),
13         .R(R)
14     );
15     initial begin
16         R=1;S=1; #50;
17         R=1;S=0; #50;
18         R=1;S=1; #50;
19         R=0;S=1; #50;
20         R=1;S=1; #50;
21         R=0;S=0; #50;
22         R=1;S=1; #50;
23     end
24 endmodule
```

仿真得到波形结果如下：



其中：

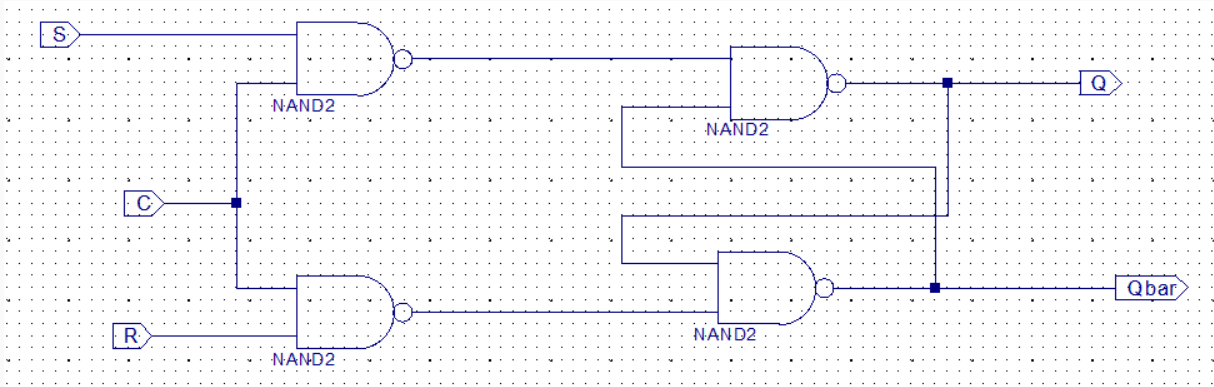
- 0~50ns 没有经过置位， $Q$  和  $\overline{Q}$  的值无意义；
- 50~100ns  $S=0$ ,  $R=1$ ，则  $Q$  被置 1， $\overline{Q}$  被置 0；
- 100~150ns  $S=R=1$ ，锁存器保持原值；
- 150~200ns  $S=1$ ,  $R=0$ ，则  $Q$  被置 0， $\overline{Q}$  被置 1；

- 200~250ns S=R=1，锁存器保持原值；
- 250~300ns S=R=0，是未定义的状态。

可见，在未定义的状态时， $Q$  和  $\bar{Q}$  的值是无意义且不可靠的。

## 2.2 CSR-Latch

新建 schematic 源文件 CSR\_LATCH.sch，绘制原理图如下：



对模块进行仿真，设计激励代码如下：

```

1  `timescale 1ns / 1ps
2
3  module CSR_LATCH_CSR_LATCH_sch_tb();
4
5      reg R;
6      reg S;
7      reg C;
8
9      wire Qbar;
10     wire Q;
11
12     CSR_LATCH UUT (
13         .Qbar(Qbar),
14         .Q(Q),
15         .R(R),
16         .S(S),
17         .C(C)
18     );
19
20     initial begin
21         C=1;R=1;S=1; #50;
22         R=1;S=0; #50;
23         R=1;S=1; #50;

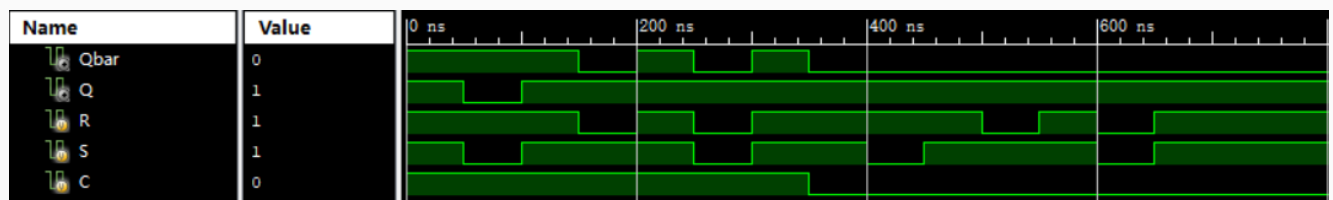
```

```

23     R=0;S=1; #50;
24     R=1;S=1; #50;
25     R=0;S=0; #50;
26     R=1;S=1; #50;
27     C=0;R=1;S=1; #50;
28     R=1;S=0; #50;
29     R=1;S=1; #50;
30     R=0;S=1; #50;
31     R=1;S=1; #50;
32     R=0;S=0; #50;
33     R=1;S=1; #50;
34     end
35 endmodule

```

仿真得到波形结果如下：



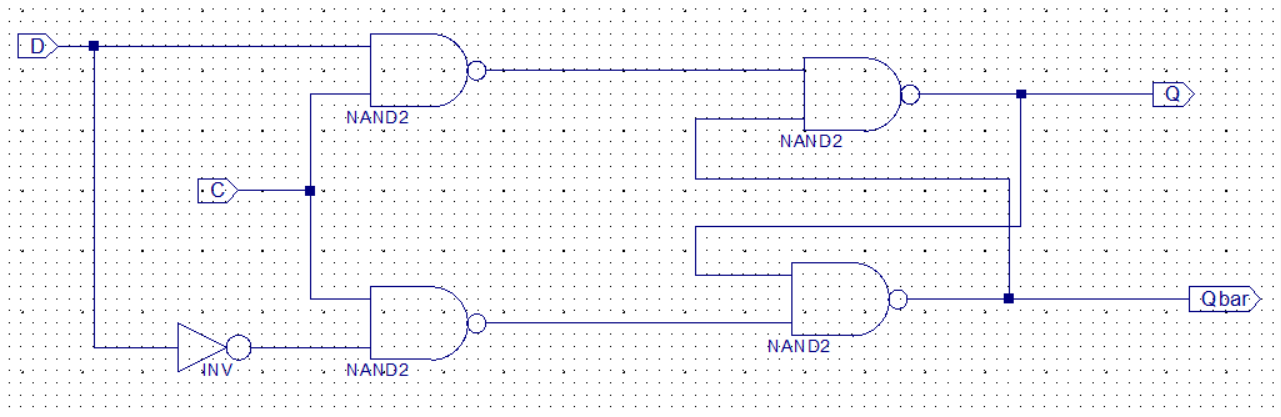
其中：

- 0~50ns 没有经过置位， $Q$  和  $\overline{Q}$  的值无意义；
- 50~100ns  $C=1$   $S=0$   $R=1$ ，则  $Q$  被置 0， $\overline{Q}$  被置 1；
- 100~150ns  $C=S=R=1$ ，是未定义的状态， $Q$  和  $\overline{Q}$  的值无意义；
- 150~200ns  $C=1$   $S=1$   $R=0$ ，则  $Q$  被置 1， $\overline{Q}$  被置 0；
- 200~250ns  $C=S=R=1$ ，是未定义的状态， $Q$  和  $\overline{Q}$  的值无意义；
- 250~300ns  $C=1$   $S=R=0$ ，锁存器保持原值；但由于保持状态并非未定义状态，因此  $Q$  和  $\overline{Q}$  的值应当相反，故此处  $\overline{Q}$  被置 0；
- 350ns 之后， $C=0$ ， $R$  和  $S$  的值不影响  $Q$  和  $\overline{Q}$  的值。

可见，门控  $C$  实现了控制功能，但仍存在未定义状态。

## 2.3 D-Latch

新建 schematic 源文件 D\_LATCH.sch，绘制原理图如下：



对模块进行仿真，设计激励代码如下：

```

1  `timescale 1ns / 1ps
2
3  module D_LATCH_D_LATCH_sch_tb();
4
5  // Inputs
6      reg C;
7      reg D;
8
9  // Output
10     wire Qbar;
11     wire Q;
12
13 // Bidirs
14
15 // Instantiate the UUT
16     D_LATCH UUT (
17         .Qbar(Qbar),
18         .Q(Q),
19         .C(C),
20         .D(D)
21     );
22 // Initialize Inputs
23     initial begin
24         C=1; D=1; #50;
25         D=0; #50;
26         C=0;D=1; #50;
27         D=0;

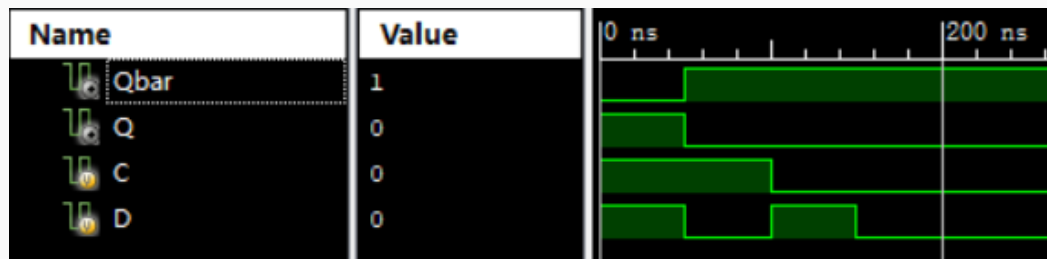
```

```

28     end
29 endmodule

```

仿真得到波形结果如下：



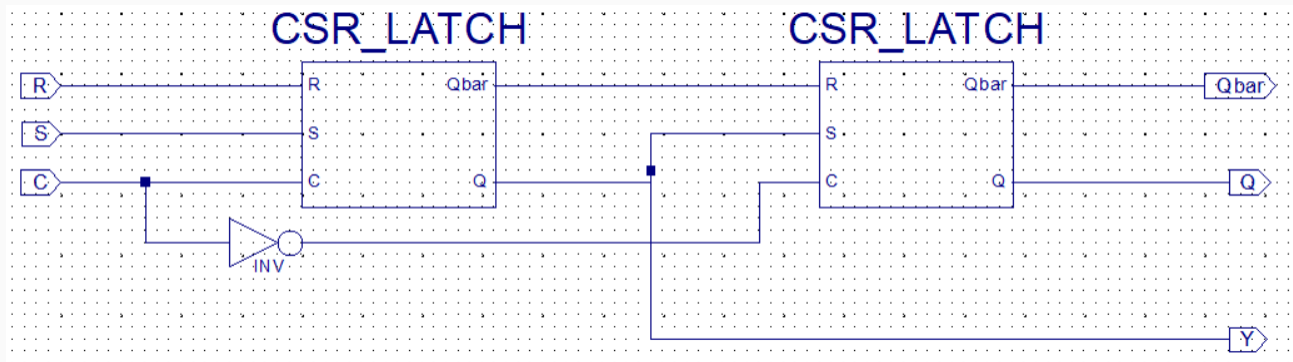
其中：

- 0~50ns C=D=1 则  $Q$  被置 1,  $\overline{Q}$  被置 0；
- 50~100ns C=1 D=0 则  $Q$  被置 0,  $\overline{Q}$  被置 1；
- 100ns 之后, C=0, D 的值不影响  $Q$  和  $\overline{Q}$  的值。

可见, D 锁存器解决了未定义状态的问题。

## 2.4 Master/Slave flip-flop

新建 schematic 源文件 MS\_FLIPFLOP.sch, 绘制原理图如下：



对模块进行仿真, 设计激励代码如下：

```

1 module MS_FLIPFLOP_MS_FLIPFLOP_sch_tb();
2
3 // Inputs
4     reg S;
5     reg R;
6     reg C;
7
8 // Output

```

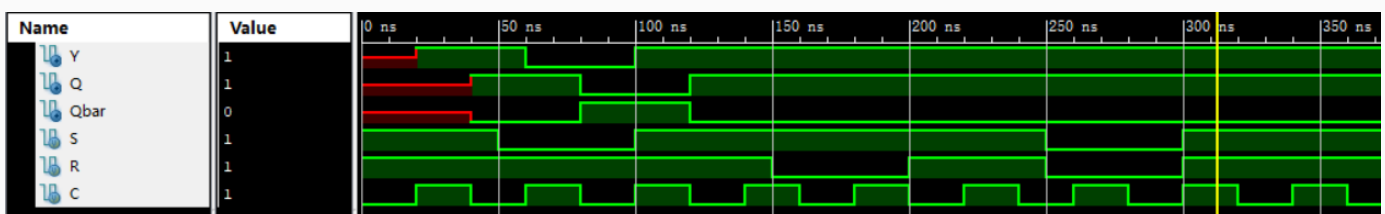


```

9    wire Y;
10   wire Q;
11   wire Qbar;
12
13 // Bidirs
14
15 // Instantiate the UUT
16   MS_FLIPFLOP UUT (
17       .S(S),
18       .R(R),
19       .C(C),
20       .Y(Y),
21       .Q(Q),
22       .Qbar(Qbar)
23   );
24 // Initialize Inputs
25 initial begin
26     R=1;S=1; #50;
27     R=1;S=0; #50;
28     R=1;S=1; #50;
29     R=0;S=1; #50;
30     R=1;S=1; #50;
31     R=0;S=0; #50;
32     R=1;S=1; #50;
33 end
34 always begin
35     C=0;#20;
36     C=1;#20;
37 end
38
39 endmodule

```

仿真得到波形结果如下：



根据设计图，只有 C 置为 0 时  $Q$  和  $\overline{Q}$  的值才能被改变，这与仿真结果是一致的。

40 ns 时，C=0 Y=1， $Q$  被置 1， $\overline{Q}$  被置 0；

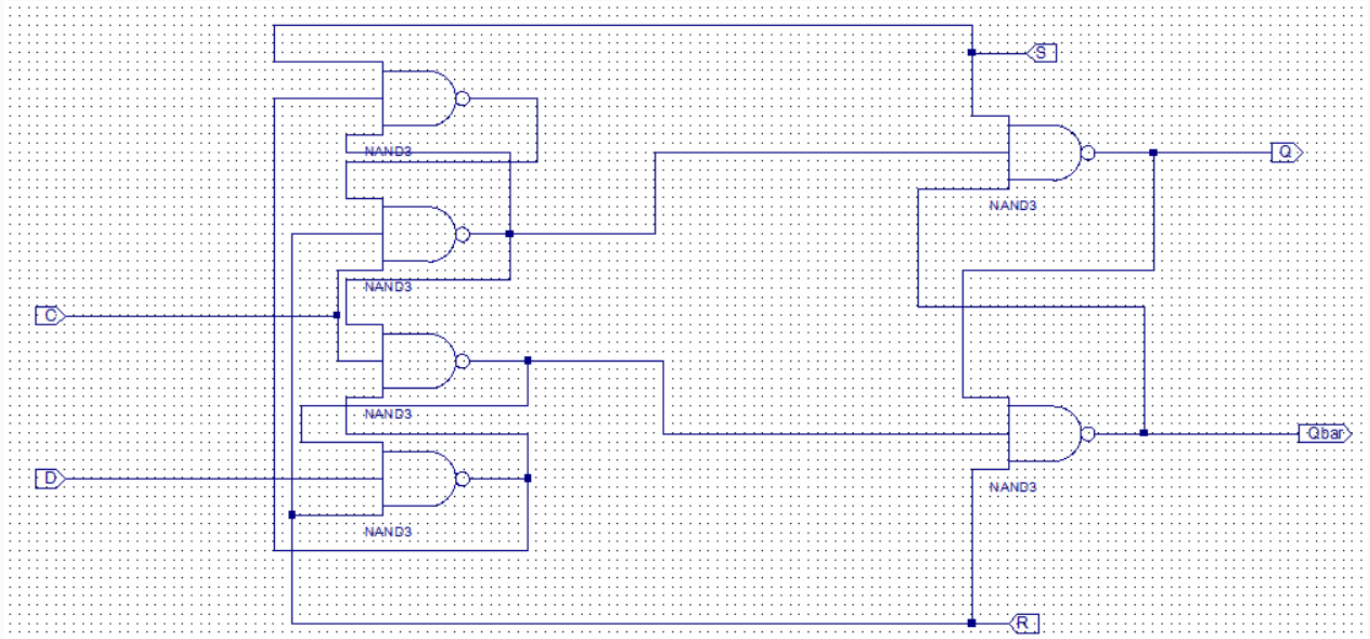
40~80 ns 时，C=1，Y 的值不影响  $Q$  和  $\overline{Q}$  的值；

80 ns 时，C=1 Y=0， $Q$  被置 0， $\overline{Q}$  被置 1；

120 ns 时，C=0 Y=1， $Q$  被置 1， $\overline{Q}$  被置 0。

## 2.5 正边沿维持阻塞型 D 触发器

新建 schematic 源文件 D\_FLIPFLOP.sch，绘制原理图如下：



对模块进行仿真，设计激励代码如下：

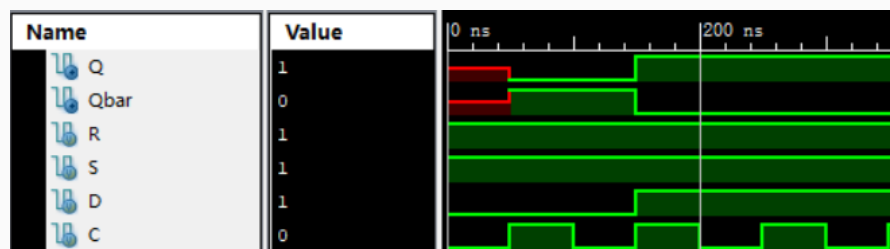
```
1 `timescale 1ns / 1ps
2
3 module D_FLIPFLOP_D_FLIPFLOP_sch_tb();
4
5 // Inputs
6     reg R;
7     reg S;
8     reg D;
9     reg C;
10
11 // Output
12     wire Q;
13     wire Qbar;
```

```

14
15 // Bidirs
16
17 // Instantiate the UUT
18   D_FLIPFLOP UUT (
19       .Q(Q),
20       .Qbar(Qbar),
21       .R(R),
22       .S(S),
23       .D(D),
24       .C(C)
25   );
26 // Initialize Inputs
27 initial begin
28     S = 1;
29     R = 1;
30     D = 0; #150;
31     D = 1; #150;
32 end
33
34 always begin
35     C=0; #50;
36     C=1; #50;
37 end
38
39 endmodule

```

仿真得到波形结果如下：



根据设计图，只有  $C=1$  时  $Q$  和  $\overline{Q}$  的值才能被改变，这与仿真结果是一致的。

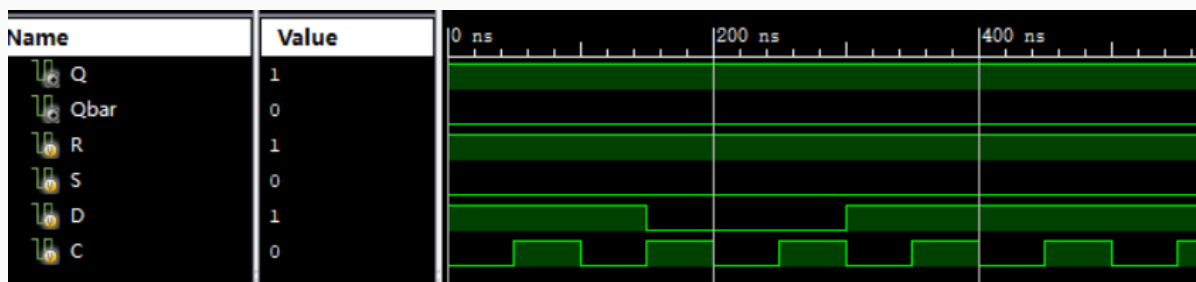
50 ns 时， $C=1$   $D=0$ ， $Q$  被置 0， $\overline{Q}$  被置 1；

150 ns 时， $C=D=1$ ， $Q$  被置 1， $\overline{Q}$  被置 0。

其他时间  $Q$  和  $\overline{Q}$  的值不会改变。

尝试 S=0 R=1 的情况：

```
1 initial begin
2     S = 0;
3     R = 1;
4     D = 1; #150;
5     D = 0; #150;
6     D = 1; #150;
7 end
8
9 always begin
10     C=0; #50;
11     C=1; #50;
12 end
```



此时，保持  $Q$  被置 1， $\overline{Q}$  被置 0。

S=1 R=0 的情况是恰好相反的。此略。

### 3 讨论与心得

本次实验完成了多种锁存器和触发器的设计和仿真，进一步熟悉了其结构特点与设计思路，并了解了它们的功能特点。

通过本次实验，我更好地了解了各种锁存器和触发器存在或者解决的问题，为日后时序电路的设计打好基础。