

# 浙江大学

## 本科实验报告

课程名称：数字逻辑设计

姓名：王伟杰

学院：计算机学院

系：软件工程

专业：软件工程

学号：3210106034

指导教师：马德

2022 年 11 月 21 日

# 浙江大学实验报告

课程名称：\_\_\_\_数字逻辑设计\_\_\_\_实验类型：\_\_\_\_

实验项目名称：\_\_\_\_锁存器与触发器基本原理\_\_\_\_

学生姓名：\_\_\_\_王伟杰\_\_\_\_专业：\_\_\_\_软件工程\_\_\_\_学号：\_\_\_\_3210106034\_\_\_\_

同组学生姓名：\_\_\_\_王熠、陈苇远\_\_\_\_指导老师：\_\_\_\_马德\_\_\_\_

实验地点：\_\_\_\_东 4-509\_\_\_\_实验日期：\_\_\_\_2022\_\_\_\_年\_\_\_\_11\_\_\_\_月\_\_\_\_21\_\_\_\_日

## 一、实验目的

- 1.1 掌握锁存器与触发器构成的条件和工作原理。
- 1.2 掌握锁存器与触发器的区别。
- 1.3 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。
- 1.4 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。

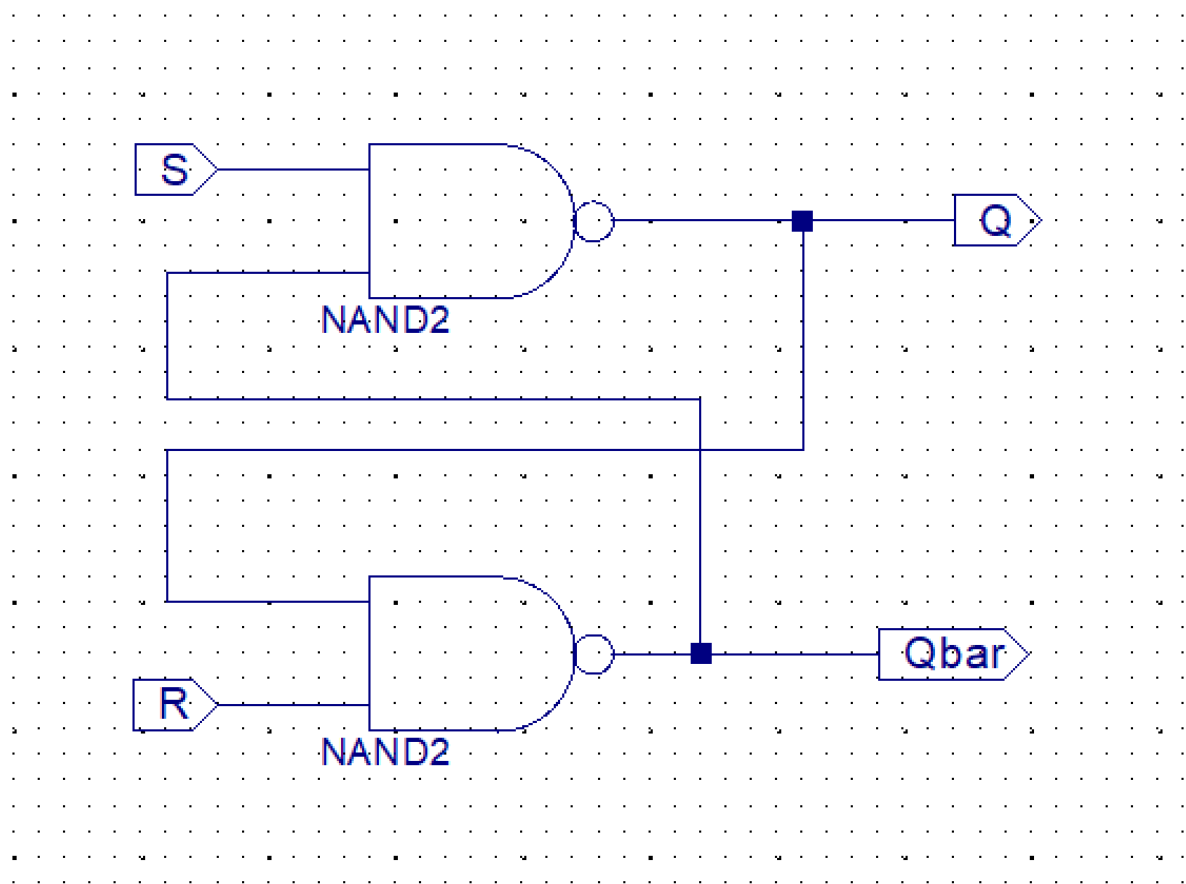
## 二、实验内容

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

## 三、操作方法与实验步骤

### 3.1 SR-LATCH

#### 1. 新建schematic源文件，绘制原理图



## 2. 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  // Inputs
4  reg S;
5  reg R;
6
7  // Output
8  wire Q;
9  wire Qbar;
10
11 // Bidirs
12
13 // Instantiate the UUT
14 SR_LATCH UUT (
15     .Q(Q),
16     .Qbar(Qbar),
17     .S(S),
18     .R(R)
19 );
20 // Initialize Inputs
21 initial begin
22     R=1;S=1; #50;
23     R=1;S=0; #50;
24     R=1;S=1; #50;
25     R=0;S=1; #50;
26     R=1;S=1; #50;
27     R=0;S=0; #50;

```

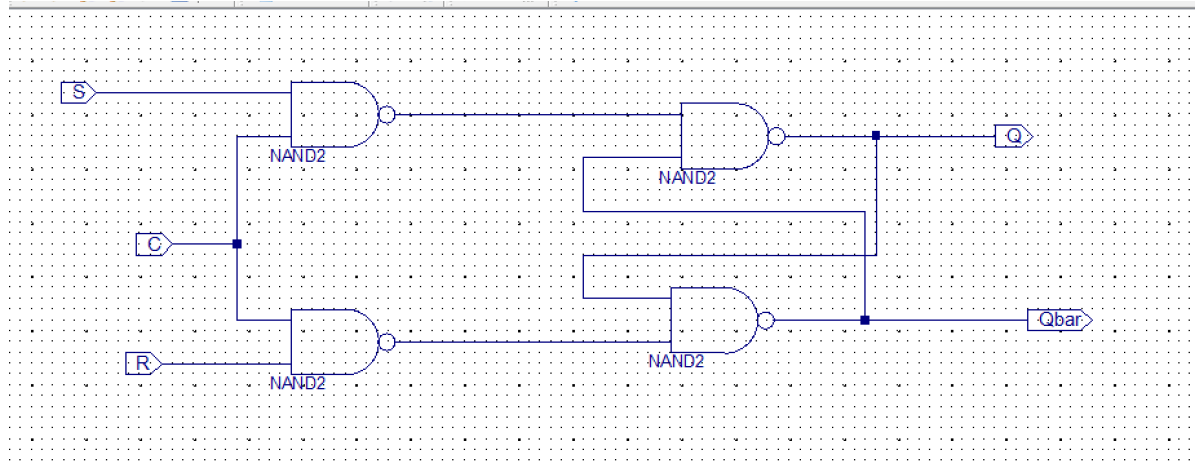
```

28     R=1;S=1; #50;
29     end
30 endmodule

```

## 3.2 CSR-LATCH

### 1. 新建schematic源文件，绘制原理图



### 2. 对模块进行仿真

```

1  module CSR_LATCH_CSR_LATCH_sch_tb();
2
3  // Inputs
4      reg R;
5      reg S;
6      reg C;
7
8  // Output
9      wire Qbar;
10     wire Q;
11
12 // Bidirs
13
14 // Instantiate the UUT
15     CSR_LATCH UUT (
16         .Qbar(Qbar),
17         .Q(Q),
18         .R(R),
19         .S(S),
20         .C(C)
21     );
22 // Initialize Inputs
23 // `ifdef auto_init
24     initial begin
25         C=1;R=1;S=1; #50;
26         R=1;S=0; #50;
27         R=1;S=1; #50;
28         R=0;S=1; #50;
29         R=1;S=1; #50;
30         R=0;S=0; #50;
31         R=1;S=1; #50;
32         C=0;R=1;S=1; #50;

```

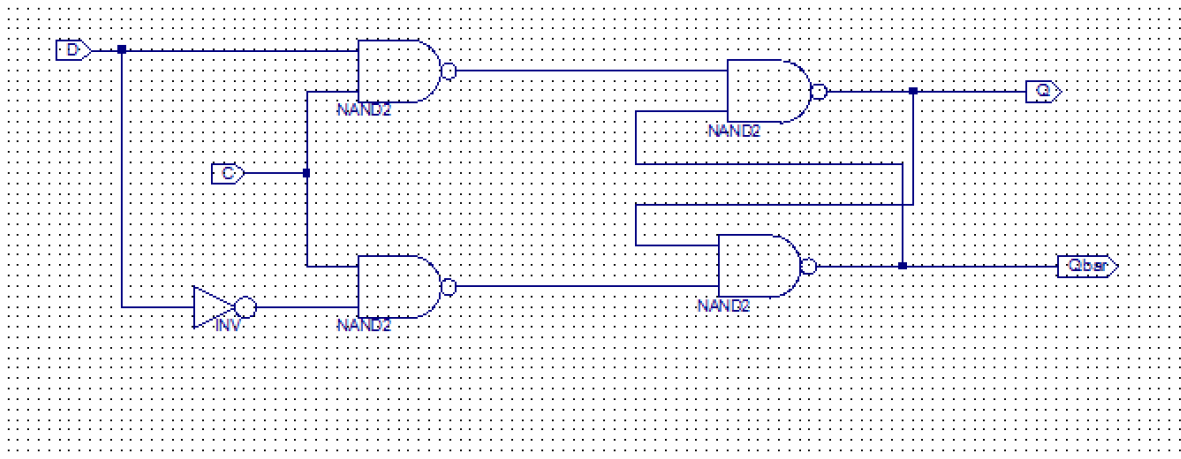
```

33     R=1;S=0; #50;
34     R=1;S=1; #50;
35     R=0;S=1; #50;
36     R=1;S=1; #50;
37     R=0;S=0; #50;
38     R=1;S=1; #50;
39     end
40     // `endif
41 endmodule

```

## 3.3 D-LATCH

### 1. 新建schematic源文件，绘制原理图



### 2. 对模块进行仿真

```

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

```

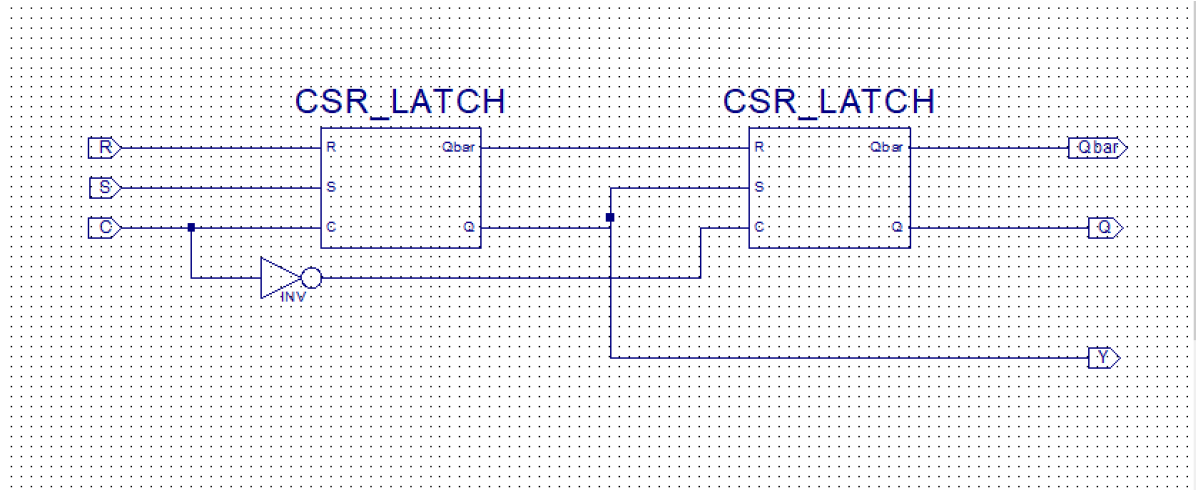
```

27     end
28     // `endif
29 endmodule

```

## 3.4 Master/Slave flip-flop

### 1. 新建schematic源文件，绘制原理图



### 2. 对模块进行仿真

```

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 // `ifdef auto_init
26     initial begin
27         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;

```



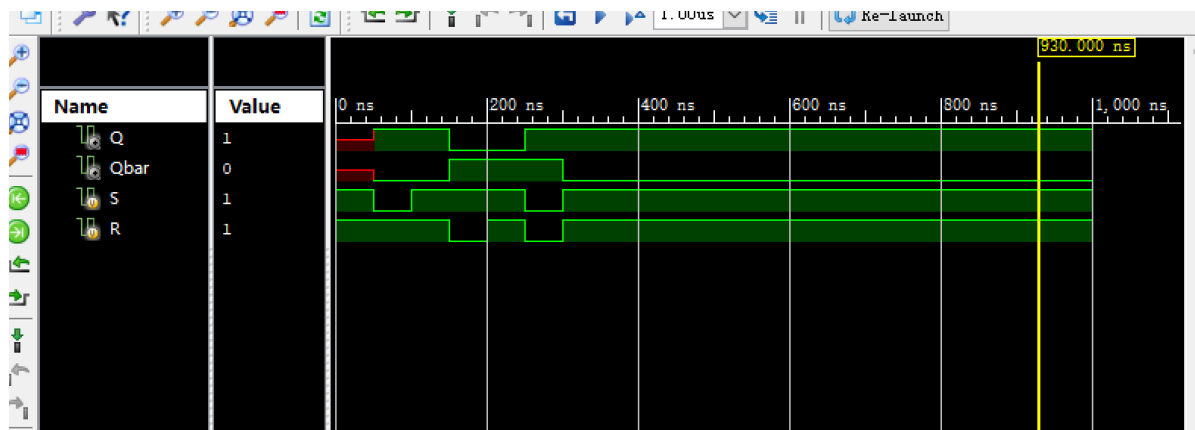
```

20         .S(S),
21         .D(D),
22         .C(C)
23     );
24     // Initialize Inputs
25     // `ifdef auto_init
26     initial begin
27         S = 1;
28         R = 1;
29         //D = 1; #150;
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     // `endif
39 endmodule

```

## 四、实验结果与分析

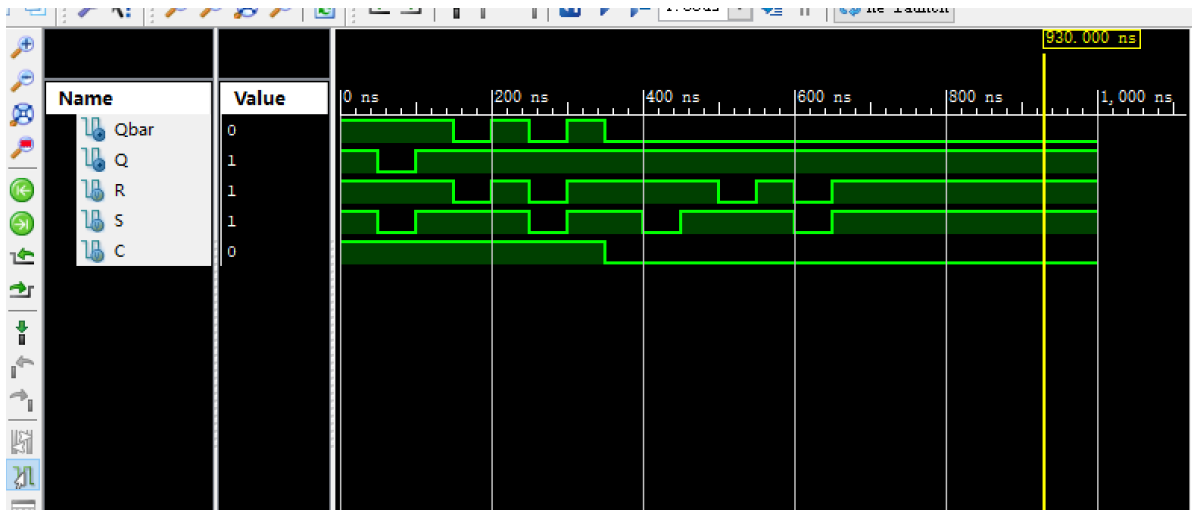
### 4.1 SR-LATCH



- 0-50ns: 没有置位
- 50-100ns:  $Q$ 置1,  $\overline{Q}$ 为0
- 100-150ns: 保持原来的值
- 150-200ns:  $Q$ 置0,  $\overline{Q}$ 为1

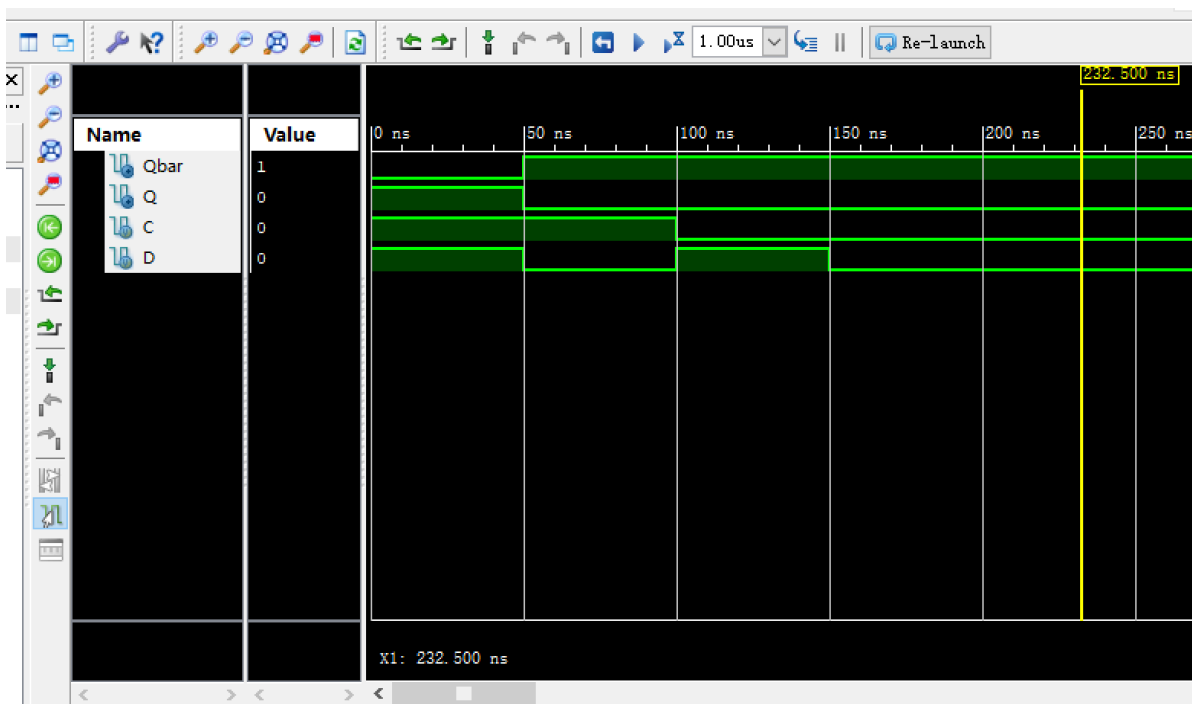
### 4.2 CSR-LATCH





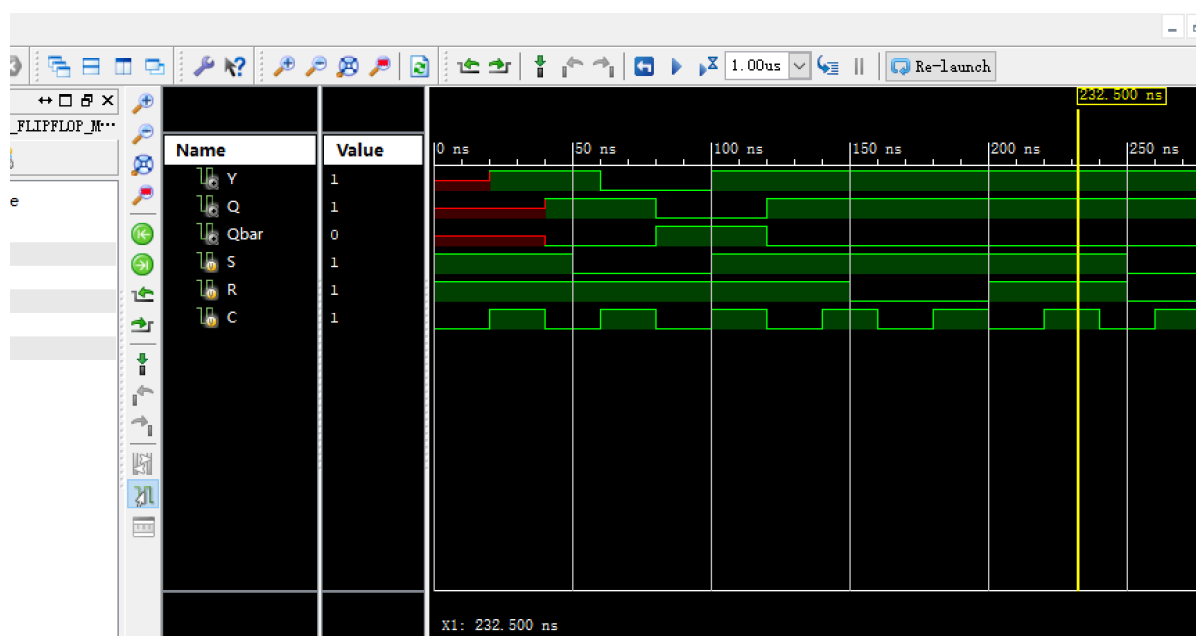
- 0-50ns: 没有置位
- 50-100ns:  $Q$ 置0,  $\overline{Q}$ 为1
- 100-150ns: 无意义
- 150-200ns:  $Q$ 置1,  $\overline{Q}$ 为0
- 200-250ns: 无意义
- 250-300ns: 锁存器保持原值, 但由于保持状态并非未定义状态, 所以 $\overline{Q}$ 为0
- 350ns之后:  $C=0$ , 不影响输出值

### 4.3 D-LATCH



- 0-50ns:  $Q$ 置1,  $\overline{Q}$ 为0
- 50-100ns:  $Q$ 置0,  $\overline{Q}$ 为1
- 100ns之后:  $C=0$ , 不影响输出

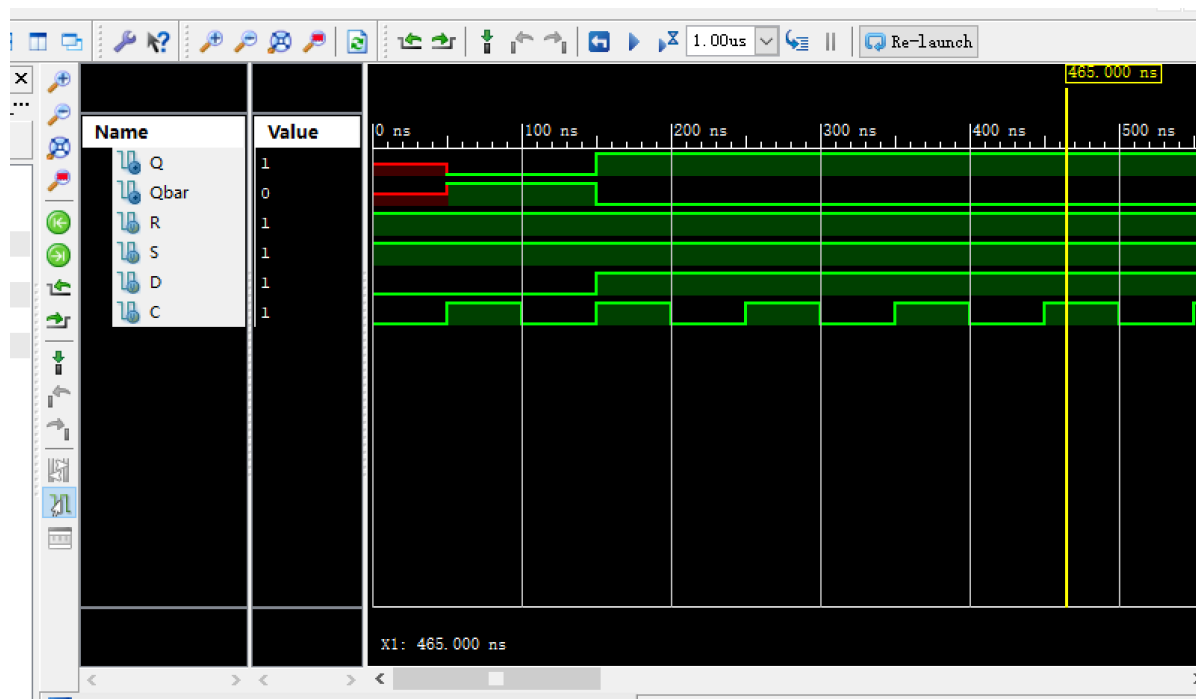
### 4.4 Master/Slave flip-flop



C=0时可以改变Q的值。

- 40ns时:  $Q$ 置1,  $\overline{Q}$ 为0
- 40-80ns: C=1, Y不影响 $Q$ 和 $\overline{Q}$
- 80ns时:  $Q$ 置0,  $\overline{Q}$ 为1
- 120ns时:  $Q$ 置1,  $\overline{Q}$ 为0

## 4.5 D触发器



C=1时可以改变Q的值。

- 50ns时:  $Q$ 置0,  $\overline{Q}$ 为1
- 150ns时: C=D=1,  $Q$ 置1,  $\overline{Q}$ 为0

总体来说, 五种情况都符合预期结果。

## 五、讨论、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，为后面时序电路的设计打好基础。