实验十 锁存器与触发器基本原理

姓名：吴欣倍

学号：3190103044

专业：软件工程

课程名称：逻辑与计算机设计基础实验

实验时间：2020.11.30.

实验地点：紫金港东4-509

指导老师：蔡铭

1. 实验目的和要求

- 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

2. 实验内容和原理

2.1 实验内容

- 任务1：实现基本SR锁存器，验证功能和存在的时序问题
- 任务2：实现门控SR锁存器，并验证功能和存在的时序问题
- 任务3：实现D锁存器，并验证功能和存在的时序问题
- 任务4：实现SR主从触发器，并验证功能和存在的时序问题
- 任务5：实现D触发器，并验证功能

2.2 实验原理

2.2.1 锁存器

2.2.1.1 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

2.2.1.2 基本锁存器种类

- SR锁存器
- D锁存器

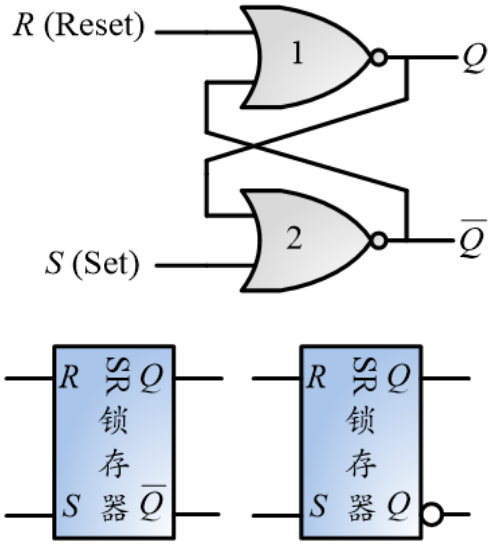
2.2.1.3 锁存器有两个稳定状态，又称双稳态电路

2.2.2 SR 锁存器

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

2.2.2.1 或非门

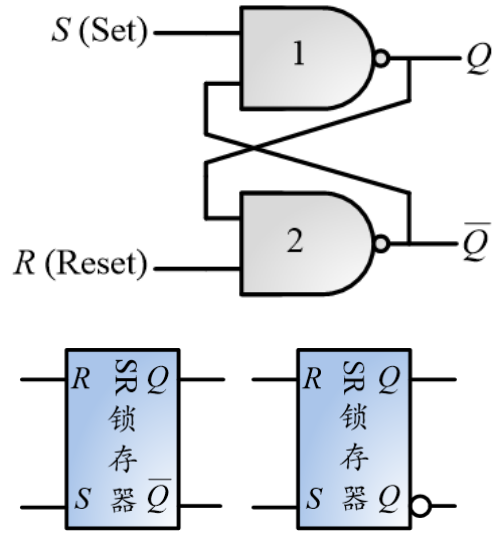
具体原理图如下所示：



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

2.2.2.2 与非门

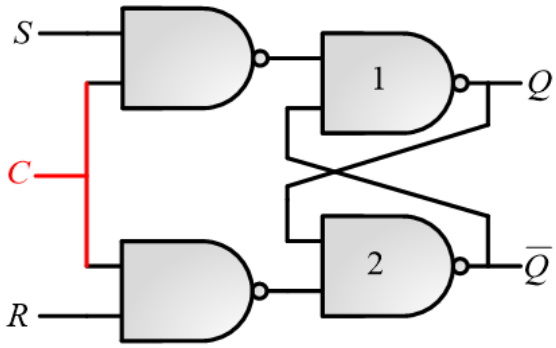
具体原理图如下所示：

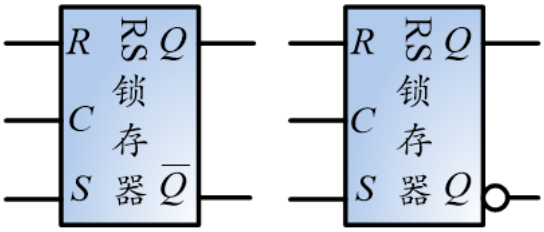


RS	$Q\ \overline{Q}$	说明
0 0	1 1	未定义
0 1	0 1	置0
1 0	1 0	置1
1 1	$Q\ \overline{Q}$	保持

2.2.2.3 门控 SR 锁存器

具体原理图如下所示：



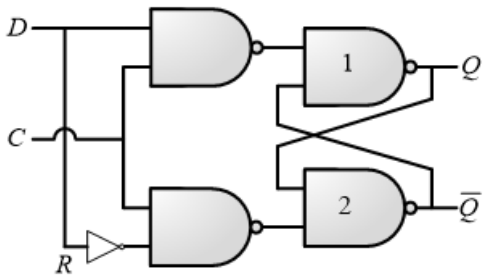


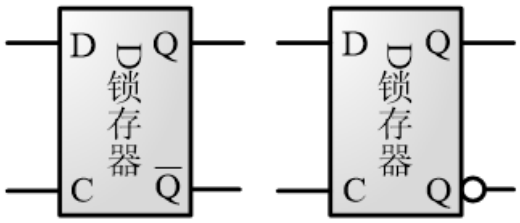
CRS	$Q\ \overline{Q}$	说明
0××	$Q\ \overline{Q}$	保持
1 0 0	$Q\ \overline{Q}$	保持
1 0 1	1 0	置1
1 1 0	0 1	置0
1 1 1	1 1	未定义

2.2.3 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

具体原理图如下所示：





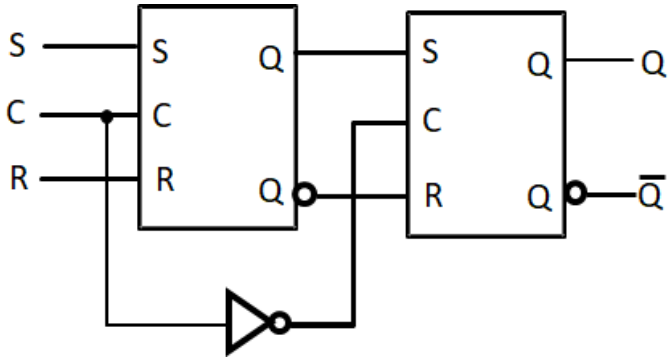
CD	$Q\overline{Q}$	说明
0 ×	$Q\overline{Q}$	保持
1 0	0 1	置0
1 1	1 0	置1

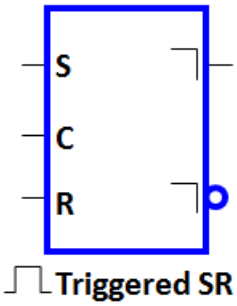
- 只需1个数据输入端 D
- 输出端Q等于输入端D
- 采用电平控制 C

2.2.4 触发器

2.2.4.1 SR 主从触发器

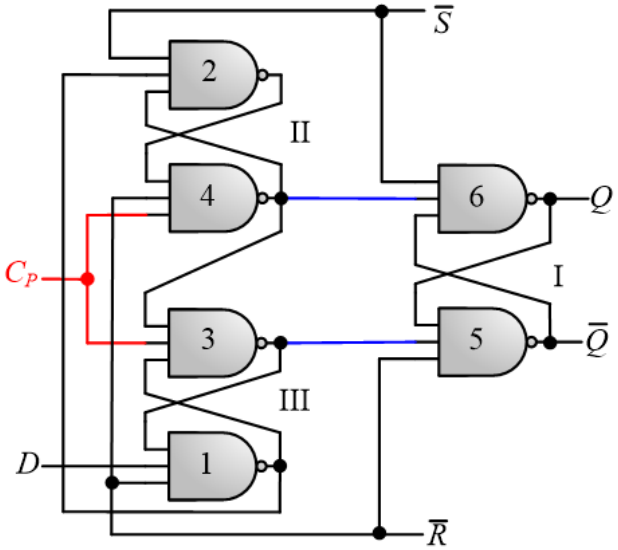
- 由两个门控 S-R 锁存器串联构成，第二个锁存器的时钟通过反相器取反
- 当 C = 1 时，输入信号进入第一个锁存器（主锁存器）
- 当 C = 0 时，第二个锁存器（从锁存器）改变输出
- 从输入到输出的通路被不同的时钟信号值(C = 1 和 C = 0)所断开

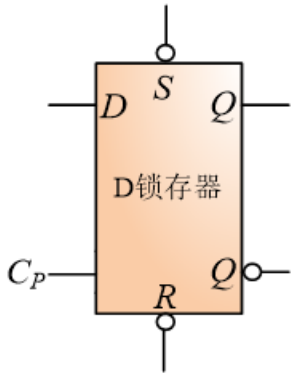




2.2.4.2 正边沿维持阻塞型D触发器

具体原理图如下所示：





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

3. 主要仪器设备

- 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

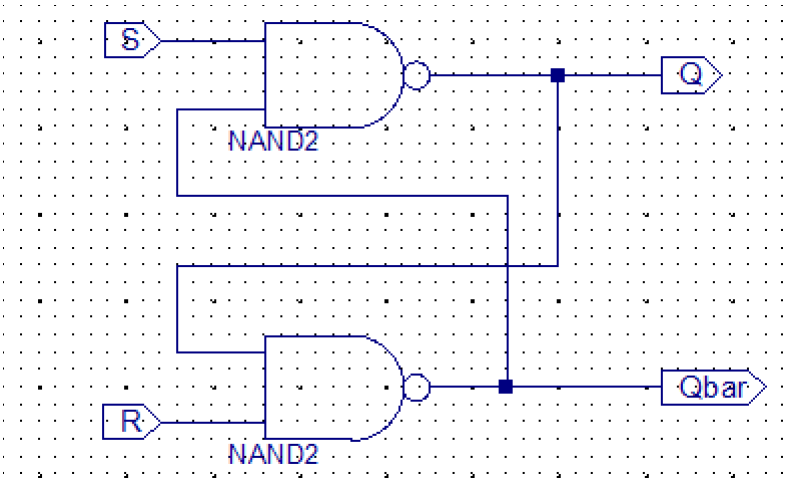
  无

4. 操作方法和实验步骤

4.1 基本SR锁存器

1. 新建工程，名为 MyLATCHS_3190103044，Top Level Source Type 为 HDL

2. 新建类型为 schematic 的源文件，命名为 SR_LATCH
3. 用原理图方式设计，具体如下所示：

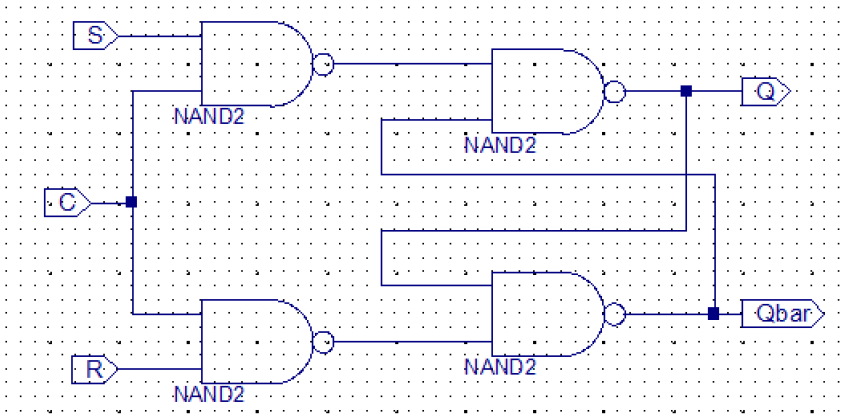


4. 生成逻辑符号和.vf文件：点击 Process 窗口下 Design Utilities -> Create schematic symbol，在工程文件夹里可以找到相应的.sym文件
5. 进行波形仿真，具体代码如下：

```
// Verilog test fixture created from schematic D:\Coding\Verilog\lab9-ALU\MyLATCHES_3190103044\SR_LATCH.sch - Mon Nov 30 13:33:14 2020

`timescale 1ns / 1ps

module SR_LATCH_SR_LATCH_sch_tb();

// Inputs
   reg S;
   reg R;

// Output
   wire Q;
   wire Qbar;

// Bidirs

// Instantiate the UUT
   SR_LATCH UUT (
        .S(S),
        .R(R),
        .Q(Q),
        .Qbar(Qbar)
   );
// Initialize Inputs
      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
endmodule
```

4.2 门控SR锁存器

1. 新建类型为 schematic 的源文件，命名为 CSR_LATCH
2. 用原理图方式设计，具体如下所示：



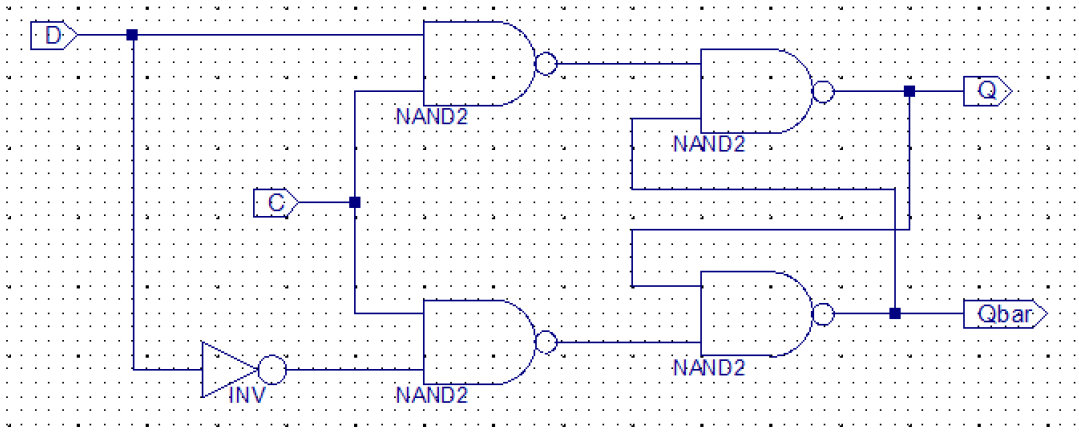
3. 生成逻辑符号和.vf文件：点击 Process 窗口下 Design Utilities -> Create schematic symbol，在工程文件夹里可以找到相应的.sym文件
4. 进行波形仿真，具体代码如下：

```
// Verilog test fixture created from schematic D:\Coding\Verilog\lab9-ALU\MyLATCHES_3190103044\CSR_LATCH.sch - Mon Nov 30 13:37:06 2020

`timescale 1ns / 1ps

module CSR_LATCH_CSR_LATCH_sch_tb();

// Inputs
   reg C;
   reg S;
   reg R;

// Output
   wire Q;
   wire Qbar;

// Bidirs

// Instantiate the UUT
   CSR_LATCH UUT (
        .C(C),
        .S(S),
        .R(R),
        .Q(Q),
        .Qbar(Qbar)
   );
// Initialize Inputs
    initial begin
        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;
    end

endmodule
```

4.3 D锁存器

1. 新建类型为 schematic 的源文件，命名为 D_LATCH
2. 用原理图方式设计，具体如下所示：



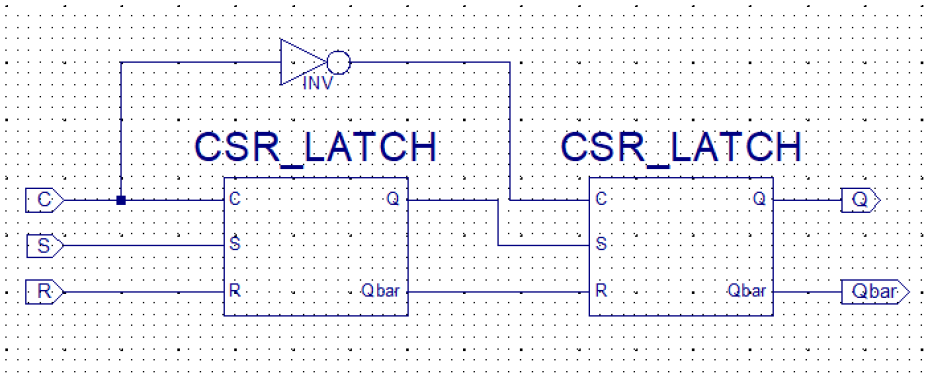
3. 生成逻辑符号和.vf文件：点击 Process 窗口下 Design Utilities -> Create schematic symbol，在工程文件夹里可以找到相应的.sym文件
4. 进行波形仿真，具体代码如下：

```
// Verilog test fixture created from schematic D:\Coding\Verilog\lab9-ALU\MyLATCHES_3190103044\D_LATCH.sch - Mon Nov 30 13:42:47 2020

`timescale 1ns / 1ps

module D_LATCH_D_LATCH_sch_tb();

// Inputs
   reg C;
   reg D;

// Output
   wire Q;
   wire Qbar;

// Bidirs

// Instantiate the UUT
   D_LATCH UUT (
        .C(C),
        .Q(Q),
        .Qbar(Qbar),
        .D(D)
   );
// Initialize Inputs
```

```
      initial begin
            C=1;D=1; #50;
            D=0; #50;
            C=0;D=1; #50;
            D=0;
        end
endmodule
```

4.4 SR主从触发器

1. 新建类型为 schematic 的源文件，命名为 MS_FLIPFLOP
2. 用原理图方式设计，具体如下所示：



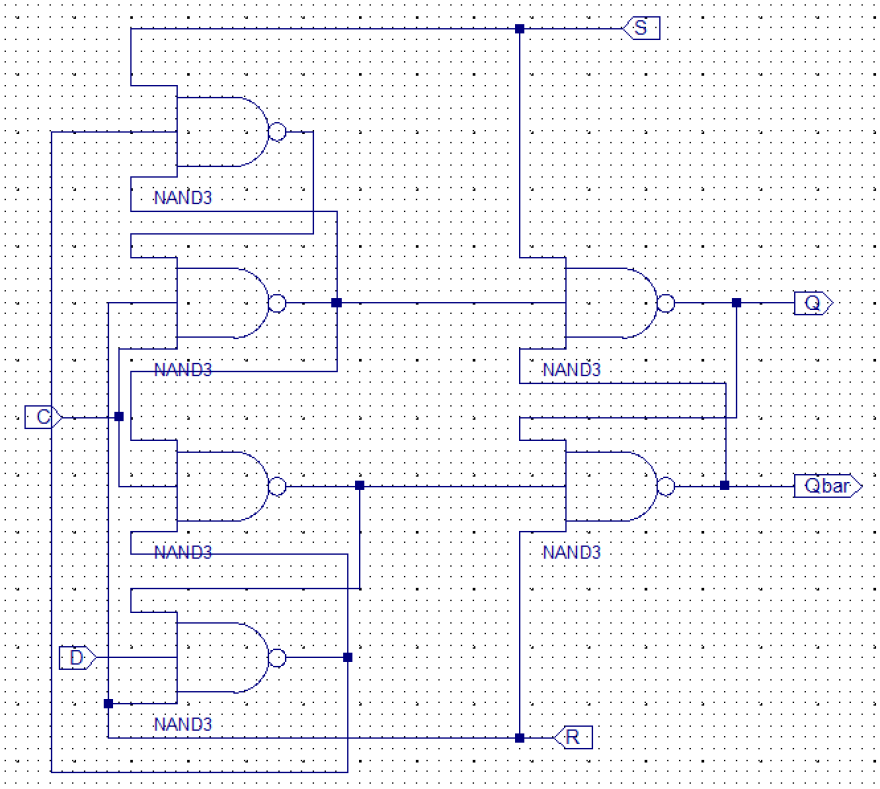
3. 生成逻辑符号和.vf文件：点击 Process 窗口下 Design Utilities -> Create schematic symbol，在工程文件夹里可以找到相应的.sym文件
4. 进行波形仿真，具体代码如下：

```
// Verilog test fixture created from schematic D:\Coding\Verilog\lab9-
ALU\MyLATCHES_3190103044\MS_FLIPFLOP.sch - Mon Nov 30 13:54:26 2020

`timescale 1ns / 1ps

module MS_FLIPFLOP_MS_FLIPFLOP_sch_tb();

// Inputs
   reg R;
   reg S;
   reg C;

// Output
   wire Q;
   wire Qbar;

// Bidirs

// Instantiate the UUT
   MS_FLIPFLOP UUT (
        .Q(Q),
        .Qbar(Qbar),
        .R(R),
        .S(S),
        .C(C)
   );
// Initialize Inputs
    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

endmodule
```

4.5 D触发器

1. 新建类型为 schematic 的源文件，命名为 MS_FLIPFLOP
2. 用原理图方式设计，具体如下所示：



3. 生成逻辑符号和.vf文件：点击 Process 窗口下 Design Utilities -> Create schematic symbol，在工程文件夹里可以找到相应的.sym文件
4. 进行波形仿真，具体代码如下：

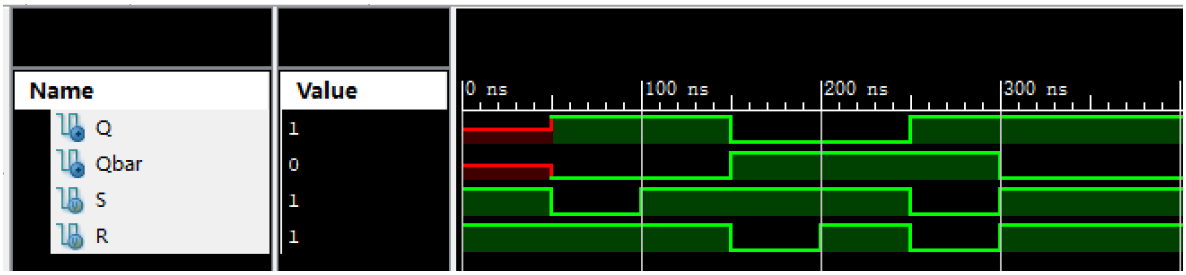
```
// Verilog test fixture created from schematic D:\Coding\Verilog\lab9-ALU\MyLATCHES_3190103044\D_FLIPFLOP.sch - Mon Nov 30 16:43:40 2020

`timescale 1ns / 1ps

module D_FLIPFLOP_D_FLIPFLOP_sch_tb();

// Inputs
   reg C;
   reg R;
   reg S;
```

```
    reg D;

// Output
    wire Qbar;
    wire Q;

// Bidirs

// Instantiate the UUT
    D_FLIPFLOP UUT (
          .C(C),
          .R(R),
          .Qbar(Qbar),
          .Q(Q),
          .S(S),
          .D(D)
    );
// Initialize Inputs
initial begin
    S = 1;
    R = 1;
     D = 0; #150;
     D = 1; #150;
end

always begin
     C=0; #50;
     C=1; #50;
end

endmodule
```

5. 实验结果和分析

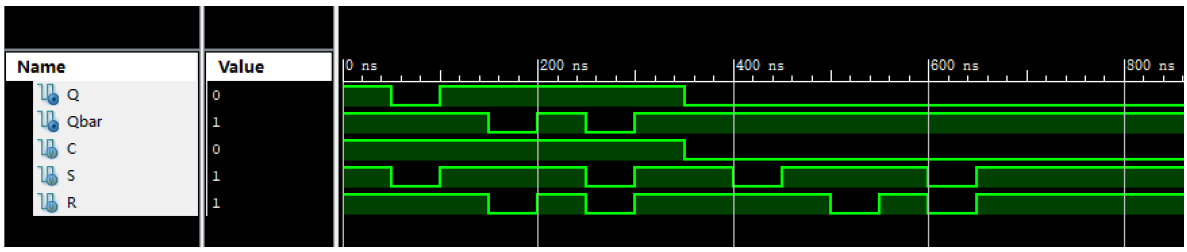
5.1 SR锁存器波形仿真结果

具体仿真结果如下图所示，与理论预估结果一致，故元件逻辑功能正确。



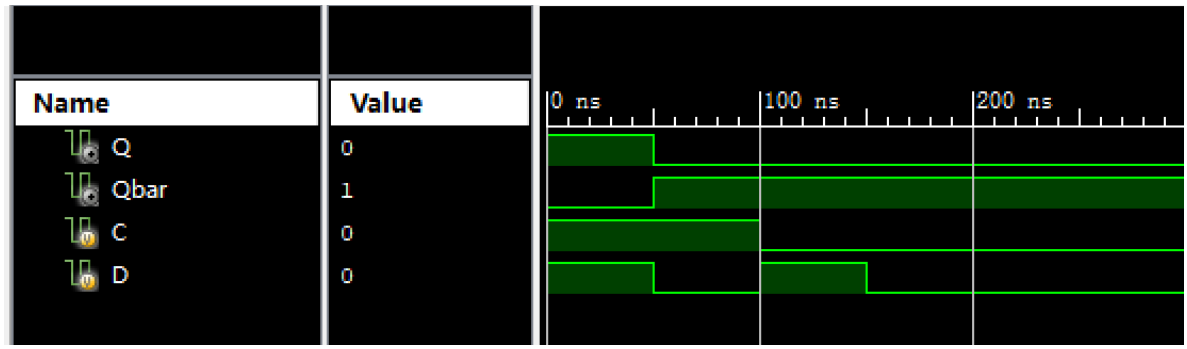
5.2 门控SR锁存器波形仿真结果

具体仿真结果如下图所示，与理论预估结果一致，故元件逻辑功能正确。



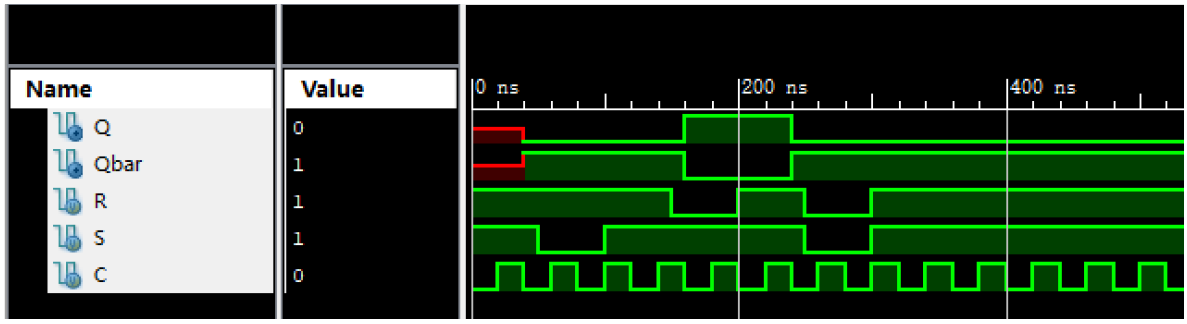
5.3 D锁存器波形仿真结果

具体仿真结果如下图所示，与理论预估结果一致，故元件逻辑功能正确。



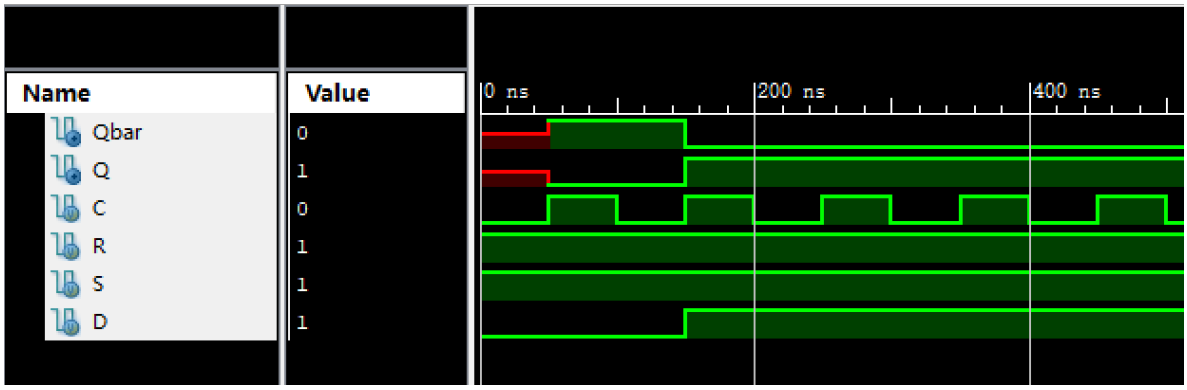
5.4 SR主从触发器波形仿真结果

具体仿真结果如下图所示，与理论预估结果一致，故元件逻辑功能正确。



5.5 D触发器波形仿真结果

具体仿真结果如下图所示，与理论预估结果一致，故元件逻辑功能正确。



6. 讨论和心得

6.1 波形仿真

波形仿真的过程中，如果修改了元件，那就需要重新建立一个仿真文件。而单纯将原仿真文件 remove 还不够，还要在文件目录下删除，防止报错。仿真过程中需要将波形仿真应用关闭方能顺利进行，不然也会报错。

6.2 D锁存器

D锁存器绘制的时候要注意，虽然实验原理提供的 S、R的输入是 Sbar、Rbar，但在绘制过程中不需要添加 inv 元件，不然会导致逻辑功能错误，输出的 Q 和 Qbar 值相同而非相反。