

浙江大学

本科实验报告

课程名称:	数字逻辑设计
姓 名:	
学 院:	竺可桢学院
专 业:	混合班
指导教师:	董亚波
报告日期:	2025 年 4 月 17 日

浙江大学实验报告

课程名称： 数字逻辑设计 实验类型： 综合

实验项目名称： 锁存器与触发器基本原理

学生姓名： 学号： 同组学生姓名： 无

实验地点： 紫金港东四 509 室 实验日期： 2025 年 4 月 16 日

一、实验目的

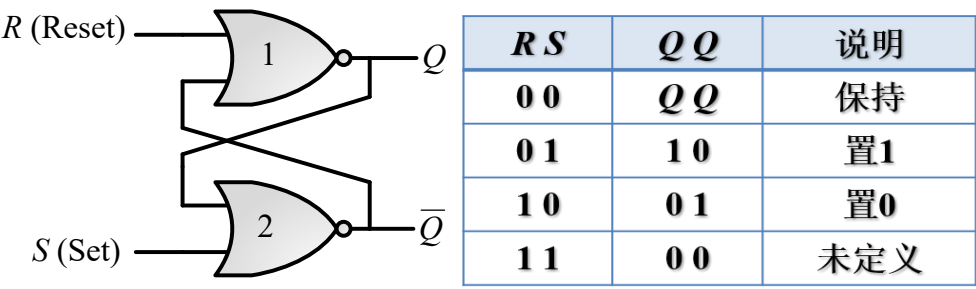
- 掌握锁存器与触发器构成的条件和工作原理
- 掌握锁存器与触发器的区别
- 掌握基本 SR 锁存器、门控 SR 锁存器、D 锁存器、SR 触发器、D 触发器的基本功能
- 掌握基本 SR 锁存器、门控 SR 锁存器、D 锁存器、SR 触发器存在的时序问题

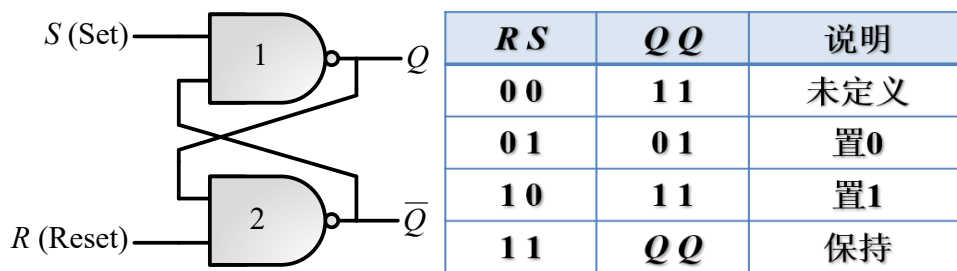
二、操作方法与实验步骤

1. 实现基本SR锁存器，验证功能和存在的时序问题

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
- 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
- 原理图和状态如下（实验中要实现的是SR锁存器）：





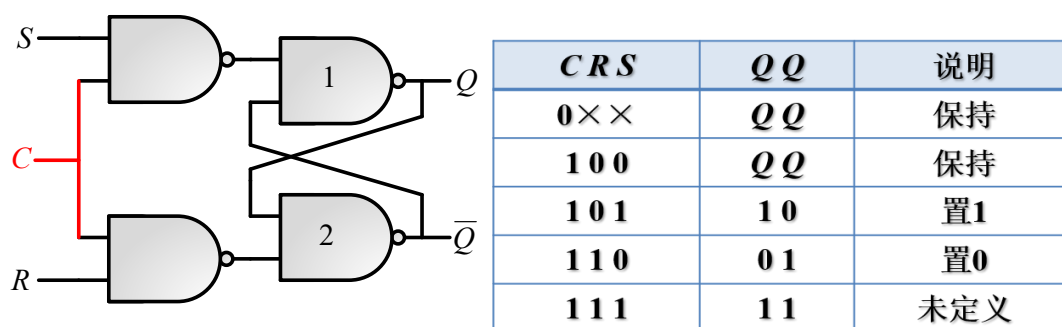
实验步骤

- 根据基本SR锁存器的原理图在 Digital 中设计基本SR锁存器的电路。
- 将基本SR锁存器的电路导出为 verilog 代码
- 设计仿真代码，并比较仿真结果与预期结果，要求仿真出其未定义状态

2. 实现门控 SR 锁存器，并验证功能和存在的时序问题

实验原理

在基本SR锁存器的基础上再增加两个与非门以及一个时钟信号，即可构成门控SR锁存器，原理图和状态如下：



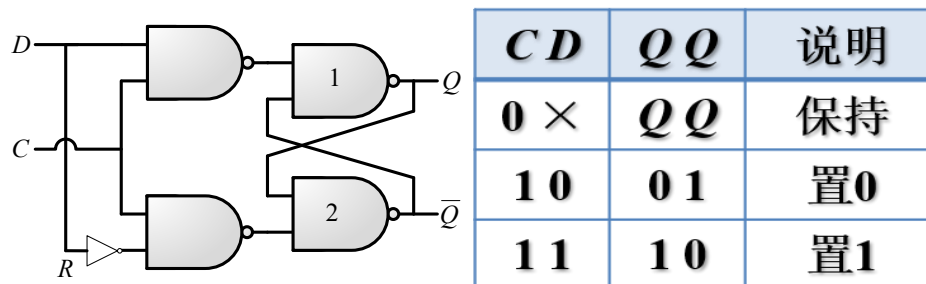
实验步骤

- 根据门控SR锁存器的原理图在 Digital 中设计门控SR锁存器的电路。
- 将门控SR锁存器的电路导出为 verilog 代码
- 设计仿真代码，并比较仿真结果与预期结果，要求仿真出其未定义状态

3. 实现 D 锁存器，并验证功能和存在的时序问题

实验原理

- 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
- 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
- 原理图和状态如下：



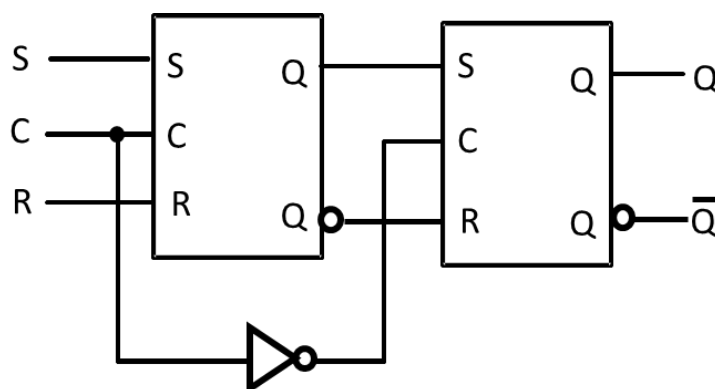
实验步骤

- 根据 D 锁存器的原理图在 Digital 中设计 D 锁存器的电路。
- 将门控 D 锁存器的电路导出为 verilog 代码
- 设计仿真代码，并比较仿真结果与预期结果
- 在 Digital 中设计关于 D 锁存器的空翻测试电路
- 将该测试电路也导出为 verilog 代码
- 设计仿真代码，使得 D 锁存器空翻，记录实验现象

4. 实现 SR 主从触发器，并验证功能和存在的时序问题

实验原理

- 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）
- SR 主从触发器由两个钟控 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
- 原理图如下：



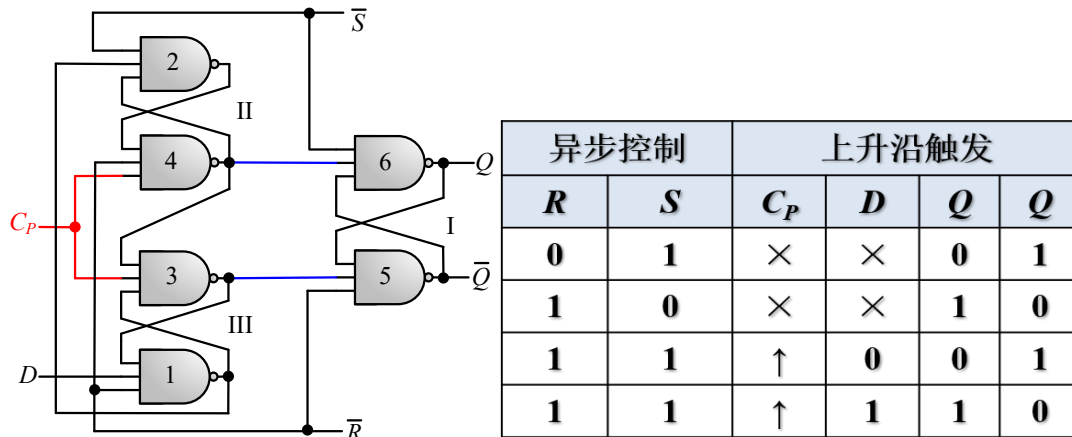
实验步骤

- 使用基本 SR 锁存器，根据原理图在 Digital 中设计 SR 主从触发器的电路。
- 将 SR 主从触发器的电路导出为 verilog 代码
- 设计仿真代码，并比较仿真结果与预期结果，要求仿真出其一次性采样问题

5. 实现 D 触发器，并验证功能

实验原理

D 触发器为边沿触发器，可以有效解决主从触发器存在的一次性采样问题,且 SR 可对触发器进行异步控制，其原理图和状态如下（正边沿触发）：



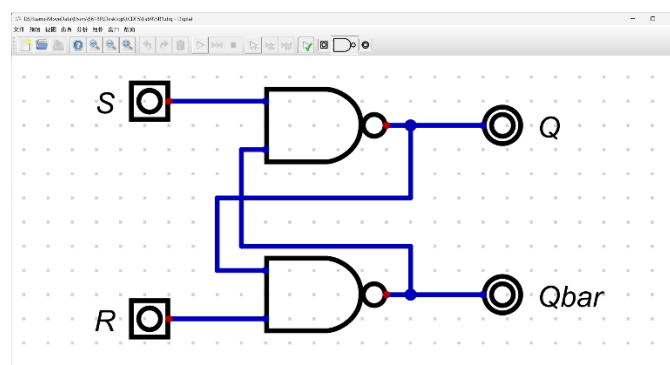
实验步骤

- 根据 D 触发器的原理图在 Digital 中设计 D 触发器的电路
- 将 D 触发器的电路导出为 verilog 代码
- 设计仿真代码，比较仿真结果与预期结果，同时检验其所解决的一次性采样问题以及 SR 的异步控制功能

三、实验结果与分析

1. 实现基本SR锁存器，验证功能和存在的时序问题

基本SR锁存器的电路如下：



将该电路导出为 verilog 代码：

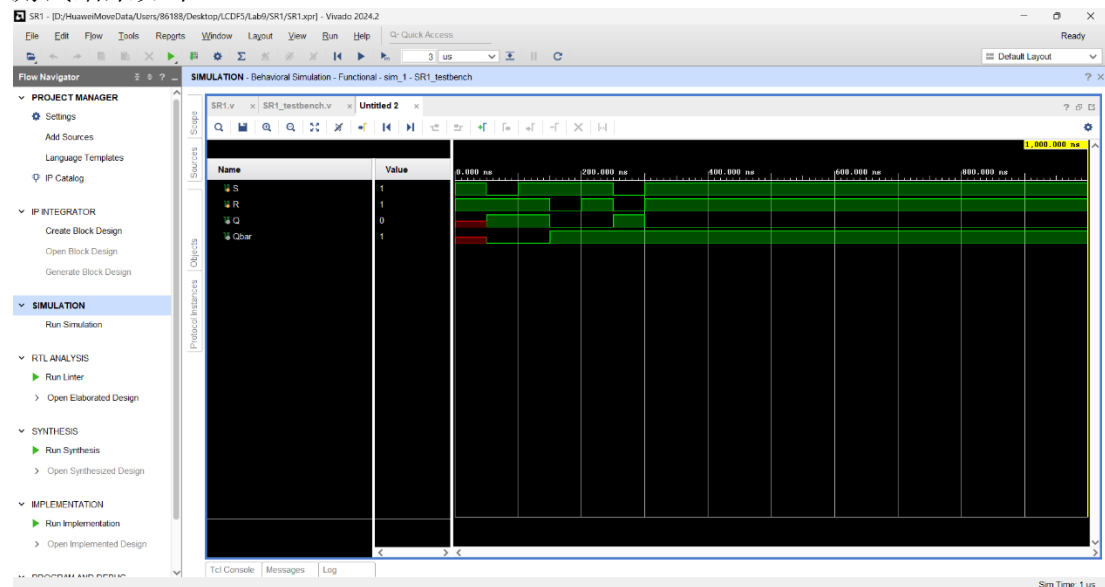
```
module SR1 (  
    input S,  
    input R,  
    output Q,  
    output Qbar
```

```
);
    wire Qbar_temp;
    wire Q_temp;
    assign Q_temp = ~ (S & Qbar_temp);
    assign Qbar_temp = ~ (Q_temp & R);
    assign Q = Q_temp;
    assign Qbar = Qbar_temp;
endmodule
```

对该电路进行仿真测试，仿真代码如下：

```
module SR1_testbench();
    reg S;
    reg R;
    wire Q;
    wire Qbar;
    SR1 uut(.R(R),.S(S),.Q(Q),.Qbar(Qbar));
    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
```

测试结果如下：

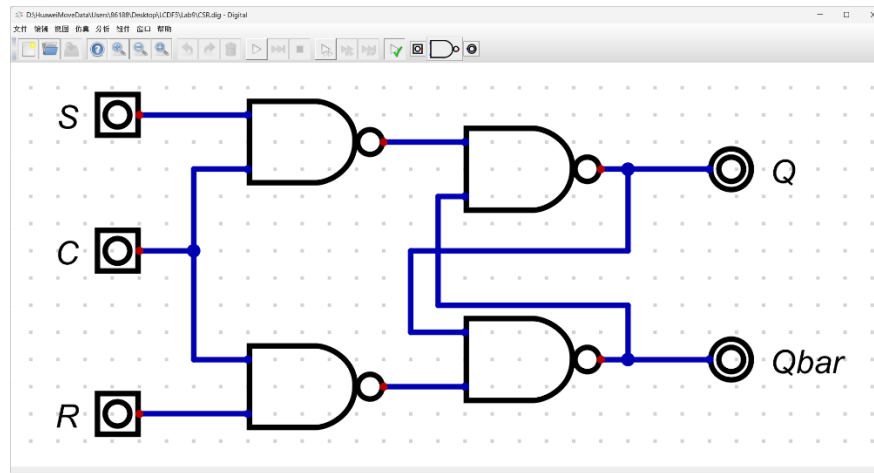


初始状态下，S 和 R 均为 1，为保持状态，而 Q 与 Qbar 的初始状态未知，故为未定态。当 S 变为 0（置 1）时，Q 变为 1，Qbar 变为 0，且当 S 回到 1 时仍然保持该状态；当 R 变为 0（置 0）时，Q 变为 0，Qbar 变为 1，且当 R 回到 1 时仍然保持该状态；当 S,R 同时变为 0 时（异常），Q 与 Qbar 同时变为 1，且当 S,R

同时变回 0 时，Q 与 Qbar 的状态将是未定义的（尽管在图中 $Q=0, Qbar=1$ ）。

2. 实现门控 SR 锁存器，并验证功能和存在的时序问题

门控 SR 锁存器的电路如下：



将该电路导出为 verilog 代码：

```
module CSR (  
    input S,  
    input R,  
    input C,  
    output Q,  
    output Qbar  
);  
    wire Qbar_temp;  
    wire Q_temp;  
    assign Q_temp = ~ (~ (S & C) & Qbar_temp);  
    assign Qbar_temp = ~ (Q_temp & ~ (C & R));  
    assign Q = Q_temp;  
    assign Qbar = Qbar_temp;  
endmodule
```

对该电路进行仿真测试，仿真代码如下：

```
module CSR_testbench();  
    reg C;  
    reg S;  
    reg R;  
    wire Q;  
    wire Qbar;  
    CSR uut(.S(S),.R(R),.C(C),.Q(Q),.Qbar(Qbar));  
    initial begin  
        C=1;S=0;R=0;#50;  
        S=0;R=1;#50;  
        S=0;R=0;#50;  
        S=1;R=0;#50;  
    end  
endmodule
```

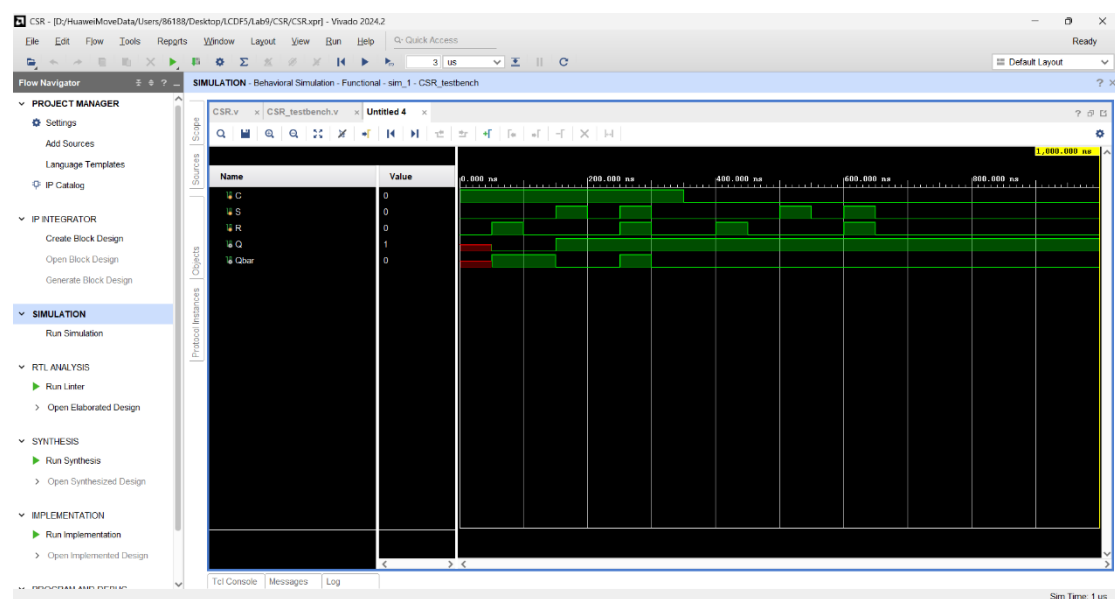
```

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

```

endmodule

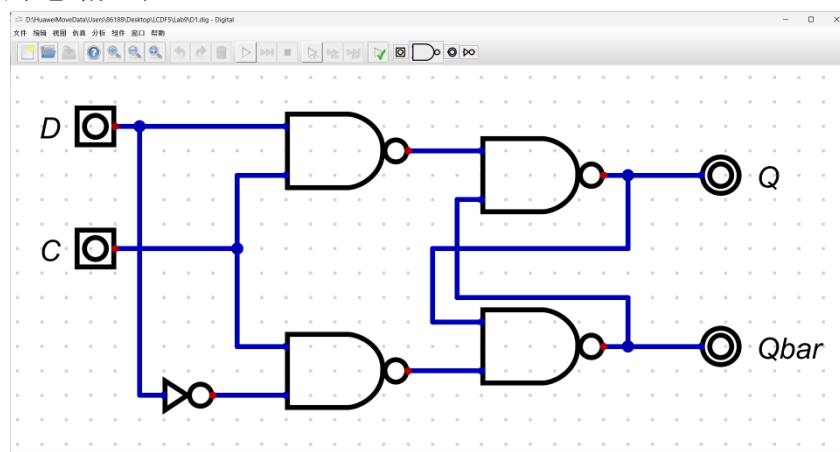
测试结果如下：



当 C 为 1 时，测试过程与 SR 锁存器完全相同，故不再赘述结果；当 C 为 0 时，无论 S,R 的状态如何，均无法对 Q 和 Qbar 起到控制作用。

3. 实现 D 锁存器，并验证功能和存在的时序问题

D 锁存器的电路如下：



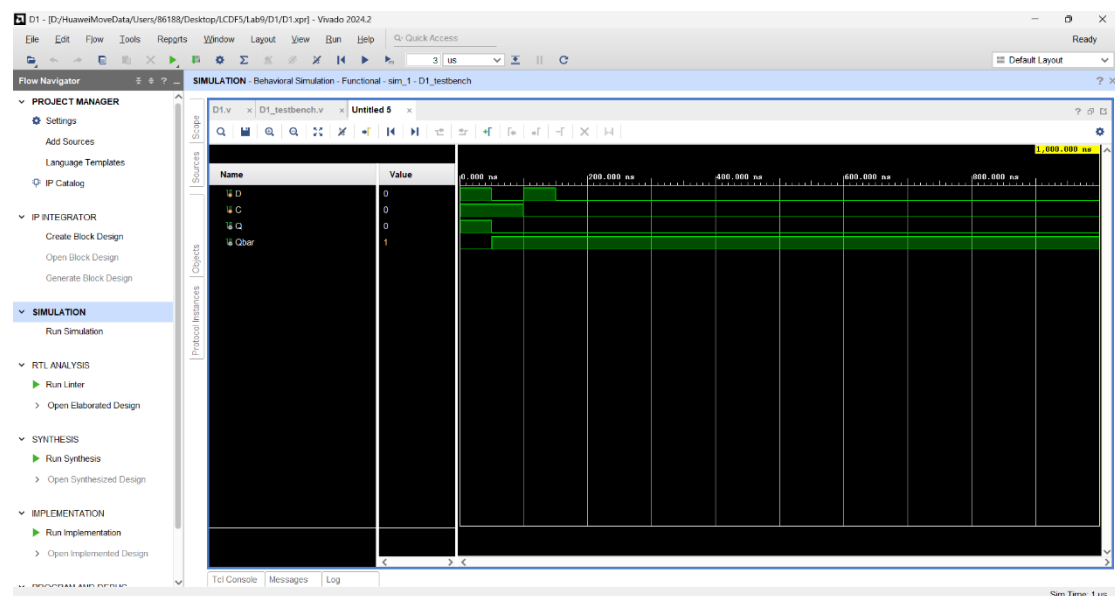
将该电路导出为 verilog 代码：

```
module D1 (  
    input D,  
    input C,  
    output Q,  
    output Qbar  
);  
    wire Qbar_temp;  
    wire Q_temp;  
    assign Q_temp = ~ (~ (D & C) & Qbar_temp);  
    assign Qbar_temp = ~ (Q_temp & ~ (C & ~ D));  
    assign Q = Q_temp;  
    assign Qbar = Qbar_temp;  
endmodule
```

对该电路进行仿真测试，仿真代码如下：

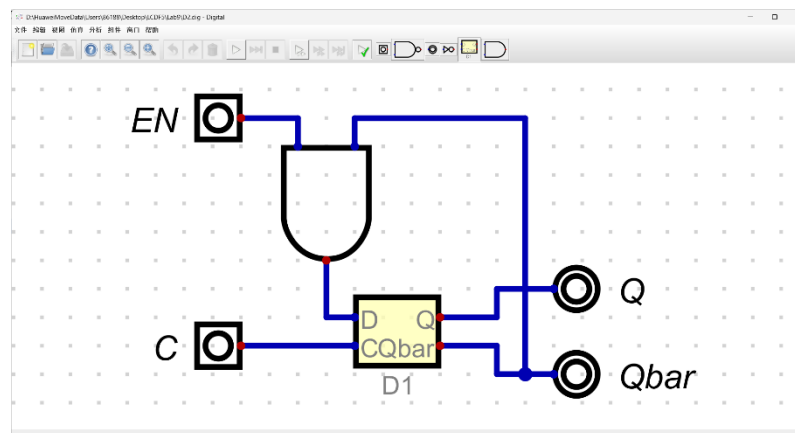
```
module D1_testbench();  
    reg D;  
    reg C;  
    wire Q;  
    wire Qbar;  
    D1 uut(.D(D),.C(C),.Q(Q),.Qbar(Qbar));  
    initial begin  
        D=1;C=1;#50;  
        D=0;C=1;#50;  
        D=1;C=0;#50;  
        D=0;C=0;#50;  
    end  
endmodule
```

测试结果如下：



当 C 为 1 时，D 的状态与 Q 的状态完全相同；当 C 为 0 时，无论 D 的状态如何变化，对 Q 的状态均无影响。

由于 D 锁存器存在空翻问题，因此搭建了对 D 锁存器进行空翻测试的电路：



同样将其导出为 verilog 代码：

```
module D1 (
    input D,
    input C,
    output Q,
    output Qbar
);
    wire Qbar_temp;
    wire Q_temp;
    assign Q_temp = ~ (~ (D & C) & Qbar_temp);
    assign Qbar_temp = ~ (Q_temp & ~ (C & ~ D));
    assign Q = Q_temp;
    assign Qbar = Qbar_temp;
endmodule

module D2 (
    input EN,
    input C,
    output Q,
    output Qbar
);
    wire s0;
    wire Qbar_temp;
    D1 D1_i0 (
        .D( s0 ),
        .C( C ),
        .Q( Q ),
        .Qbar( Qbar_temp )
    );
    assign s0 = (Qbar_temp & EN);
endmodule
```

```

    assign Qbar = Qbar_temp;
endmodule

```

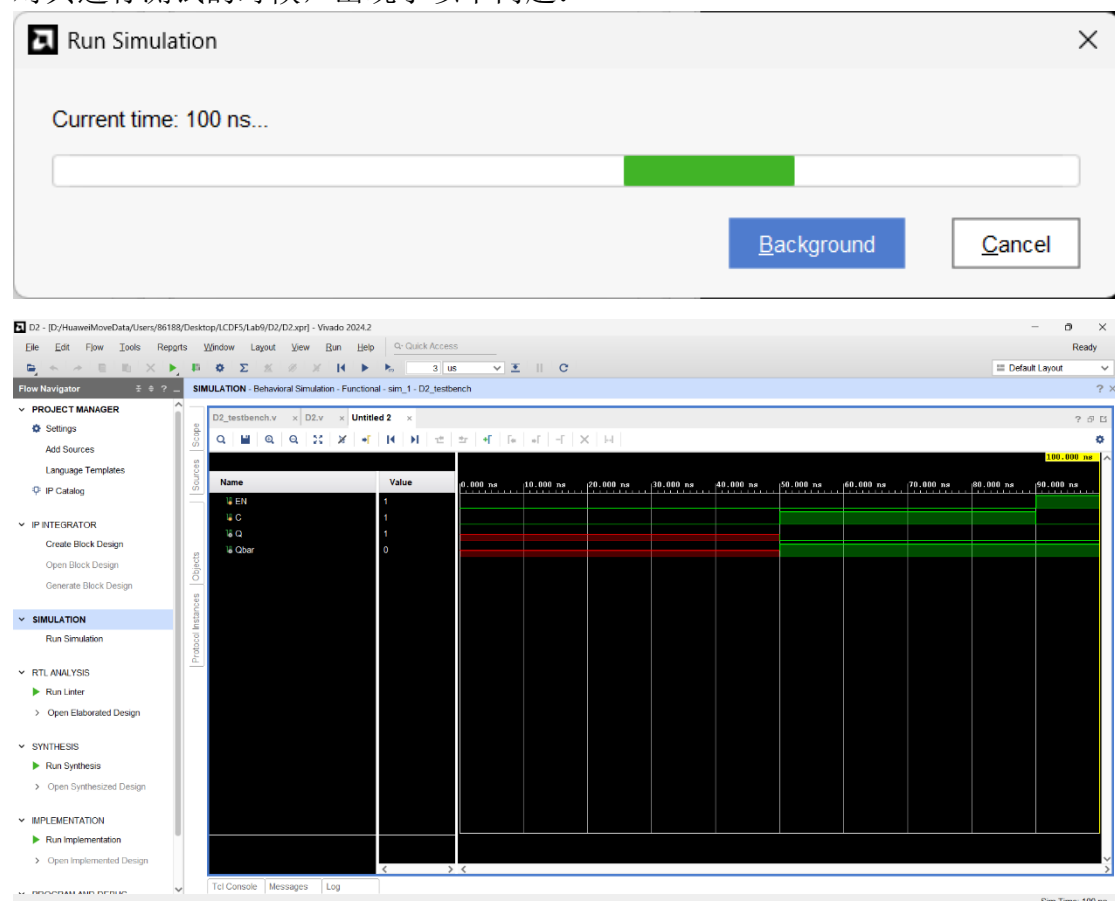
并编写仿真代码：

```

module D2_testbench();
    reg EN;
    reg C;
    wire Q;
    wire Qbar;
    D2 uut(.EN(EN),.C(C),.Q(Q),.Qbar(Qbar));
    initial begin
        EN=0;C=0;#50;
        EN=0;C=1;#40;
        C=0;EN=1;#10;
        EN=1;C=1;#0.001;
        EN=1;C=0;#50;
    end
endmodule

```

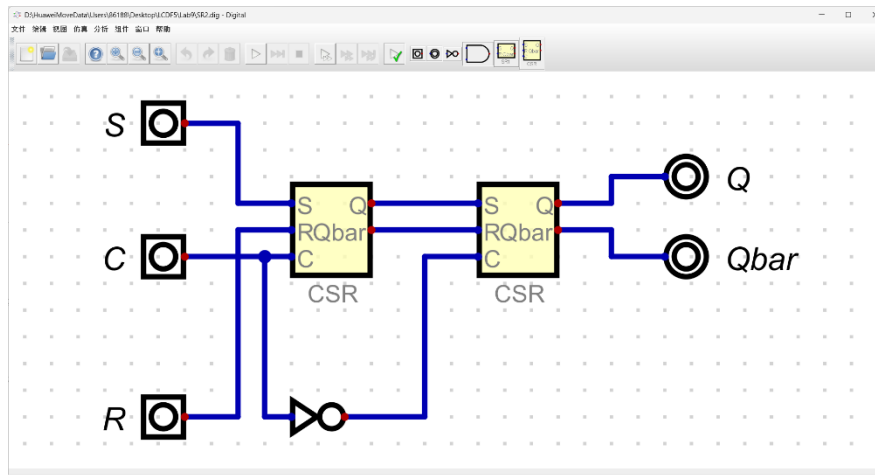
对其进行测试的时候，出现了以下问题：



也就是出现了仿真时间卡在了开始空翻的时间(100ns)，一直无法结束。

4. 实现 SR 主从触发器，并验证功能和存在的时序问题

SR 主从触发器的电路如下：



将该电路导出为 verilog 代码:

```
module CSR (
    input S,
    input R,
    input C,
    output Q,
    output Qbar
);
    wire Qbar_temp;
    wire Q_temp;
    assign Q_temp = ~ (~ (S & C) & Qbar_temp);
    assign Qbar_temp = ~ (Q_temp & ~ (C & R));
    assign Q = Q_temp;
    assign Qbar = Qbar_temp;
endmodule
```

```
module SR2 (
    input S,
    input R,
    input C,
    output Q,
    output Qbar
);
    wire s0;
    wire s1;
    wire s2;
    CSR CSR_i0 (
        .S( S ),
        .R( R ),
        .C( C ),
        .Q( s0 ),
        .Qbar( s1 )
    );
    wire s2;
    CSR CSR_i1 (
        .S( s0 ),
        .R( s1 ),
        .C( C ),
        .Q( Q ),
        .Qbar( Qbar )
    );
endmodule
```

```

);
assign s2 = ~ C;
CSR CSR_i1 (
    .S( s0 ),
    .R( s1 ),
    .C( s2 ),
    .Q( Q ),
    .Qbar( Qbar )
);
endmodule

```

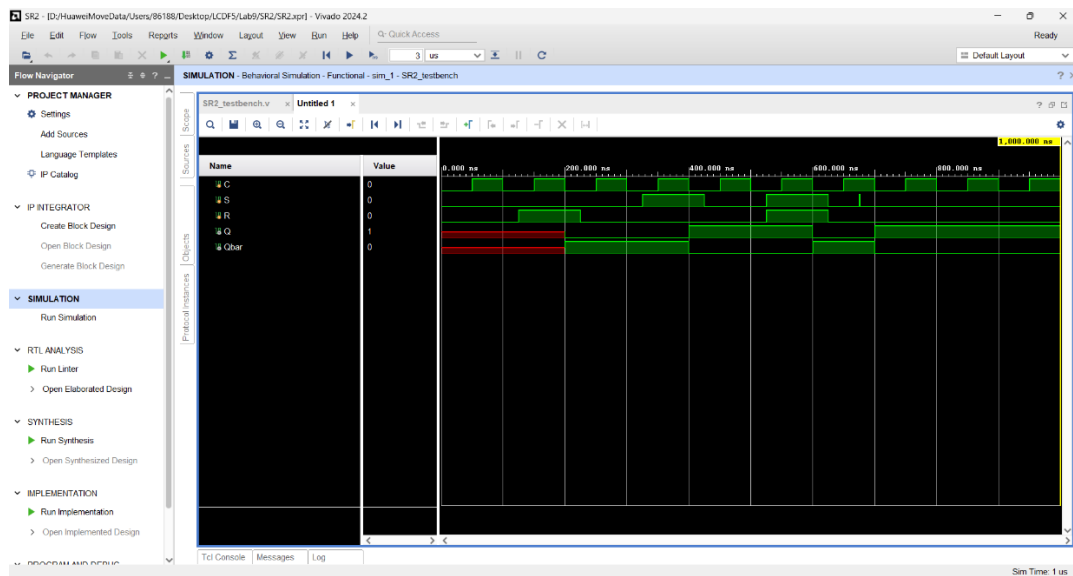
对该电路进行仿真测试，仿真代码如下（同时验证其一次性采样问题）：

```

module SR2_testbench();
    reg C;
    reg S;
    reg R;
    wire Q;
    wire Qbar;
    SR2 uut( .S(S), .R(R), .C(C), .Q(Q), .Qbar(Qbar));
    initial begin
        S=0;R=0;#125;
        S=0;R=1;#100;
        S=0;R=0;#100;
        S=1;R=0;#100;
        S=0;R=0;#100;
        S=1;R=1;#100;
        S=0;R=0;#50;
        S=1;R=0;#2;
        S=0;R=0;
    end
    always begin
        C=0;#50;
        C=1;#50;
    end
end
endmodule

```

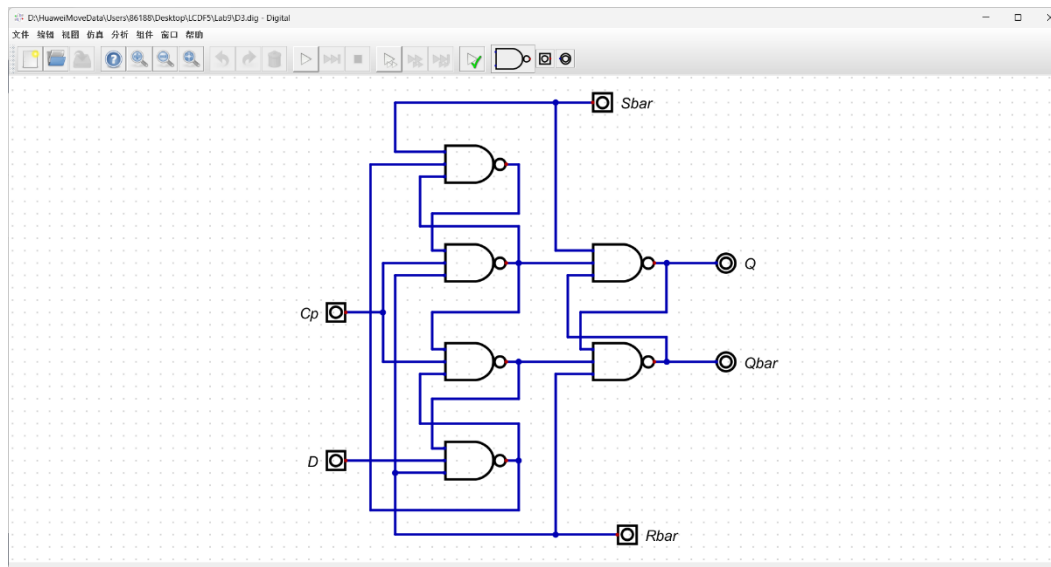
测试结果如下：



在该仿真中，每 50ns，时钟信号改变一次。在前三次 S,R 值变化时，Q 与 Qbar 的信号将相对 S,R 的信号滞后一个时钟周期发生变化（第三次较特殊，为禁用状态，Q 与 Qbar 的实际状态应当是不确定的）。在第四次变化（S 在 C=1 的时候跳动了以下）后，Q 与 Qbar 的状态再次发生了改变，即验证了一次性采样问题。

5. 实现 D 触发器，并验证功能

D 触发器的电路如下：



将该电路导出为 verilog 代码：

```
module D3 (
    input D,
    input Cp,
    input Sbar,
    input Rbar,
    output Q,
    output Qbar
);
```

```

wire s0;
wire s1;
wire s2;
wire Qbar_temp;
wire Q_temp;
assign s2 = ~ (s0 & Cp & s1);
assign s0 = ~ (~ (Sbar & s1 & s0) & Cp & Rbar);
assign s1 = ~ (s2 & D & Rbar);
assign Q_temp = ~ (Sbar & s0 & Qbar_temp);
assign Qbar_temp = ~ (Q_temp & s2 & Rbar);
assign Q = Q_temp;
assign Qbar = Qbar_temp;
endmodule

```

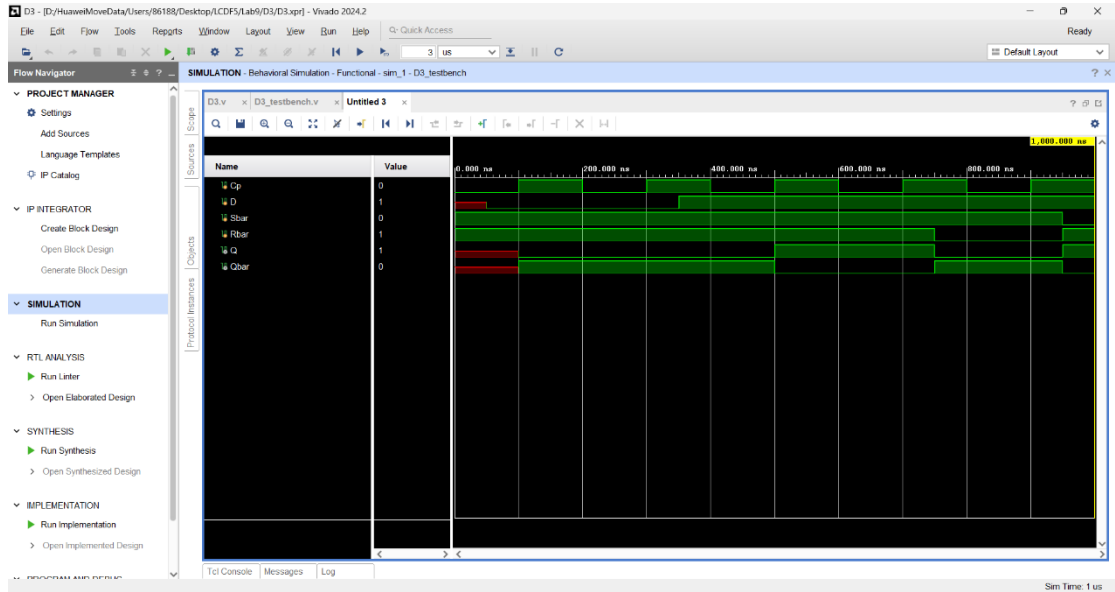
对该电路进行仿真测试，仿真代码如下：

```

module D3_testbench();
    reg Cp;
    reg D;
    reg Sbar;
    reg Rbar;
    wire Q;
    wire Qbar;
    D3 uut(.D(D),.Cp(Cp),.Sbar(Sbar),.Rbar(Rbar),.Q(Q),.Qbar(Qbar));
    initial begin
        Sbar=1;
        Rbar=1;
        #50 D=0;
        #300 D=1;
        #400 Rbar=0;
        #200 Rbar=1;Sbar=0;
    end
    always begin
        Cp=0;#100;
        Cp=1;#100;
    end
endmodule

```

测试结果如下：



Cp 每 100ns 改变一次状态。当 D 信号在 Cp 上升沿时为 0 时，Q 将接收到 D 的信号，因此 Q 与 Qbar 将在 Cp 的上升沿时分别确定为 0 和 1；当 D 信号在 Cp 固定或在下降沿时为 1 时，对 Q 的信号没有影响；直到 Cp 到达下一个上升沿时，Q 与 Qbar 的状态才发生改变；同时，当 S,R 的状态发生改变（不为保持状态）时，Q 与 Qbar 的值将立刻发生改变，无视时钟信号。因此，这样能够完成对 D 触发器行为的仿真。

四、讨论、心得

这一次实验中，我进一步掌握了基本 SR 锁存器、门控 SR 锁存器、D 锁存器、SR 主从触发器以及 D 触发器的工作原理，同时也对它们存在的问题进行了一一验证，理解了其出现问题的原因，这将对后续复杂时序电路的设计打下良好的基础。