

Lab3

姓名 学号

孙昊哲 PB20000277

实验题目

简单时序逻辑电路

实验目的

- 掌握时序逻辑相关器件的原理及底层结构
- 能够用基本逻辑门搭建各类时序逻辑器件
- 能够使用 Verilog HDL 设计简单逻辑电路

实验环境

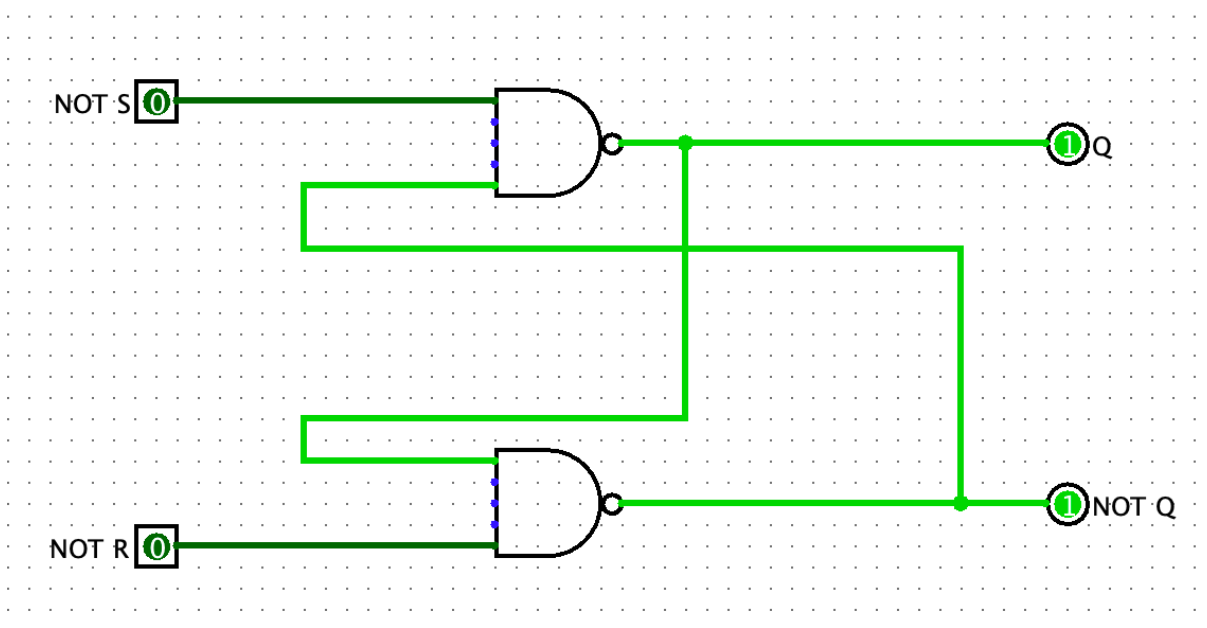
macOS Ventura

Logism 2.7.1

Openjdk 18.0.2.1

实验练习

T1

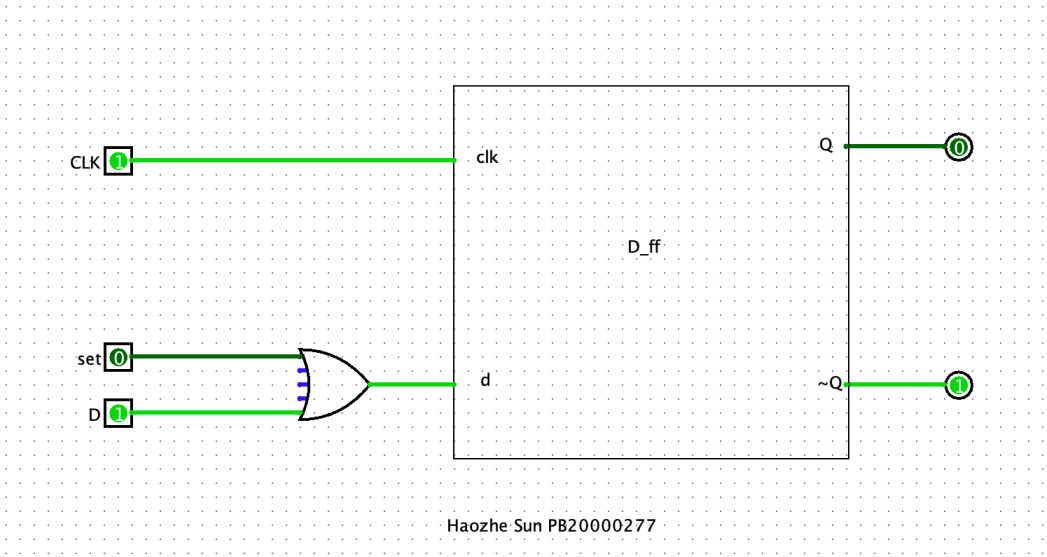


SR锁存器真值表				
S	R	Q	Q_{next}	解释
0	0	0	0	维持
0	0	1	1	维持
0	1	0	0	重置
0	1	1	0	重置
1	0	0	1	设置
1	0	1	1	设置
1	1	0	-	不允许
1	1	1	-	不允许

电路图功能：S 和 R 都为 1 时保持，仅 S 为 0 时输出 Q 设为 1，仅R 为 0 时输出 Q 设为 0，S 和 R 都为 0 时状态未定义。

T2

我们设计一个或门即可，在set置为1时，即可使得或门的结果置为1



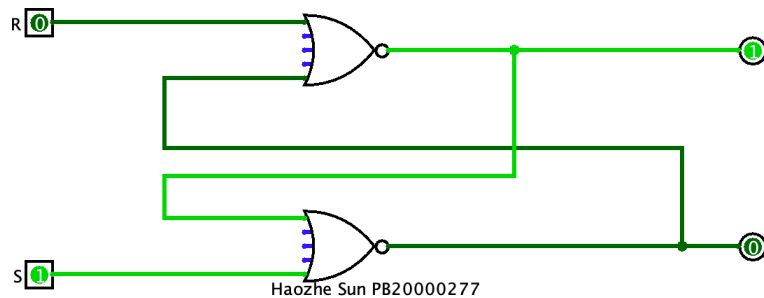
```

module d_ff_s(input clk, set, d, output reg q);
  always@(posedge clk)
  begin
    if(set==1)
      q<=1'b1;
    else
      q<=d;
    end
  endmodule

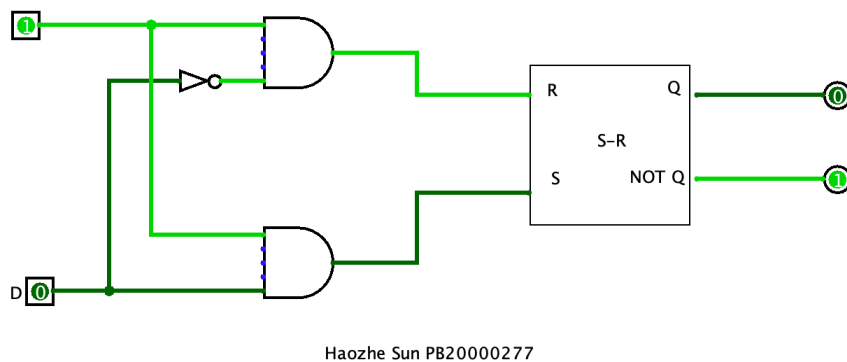
```

T3

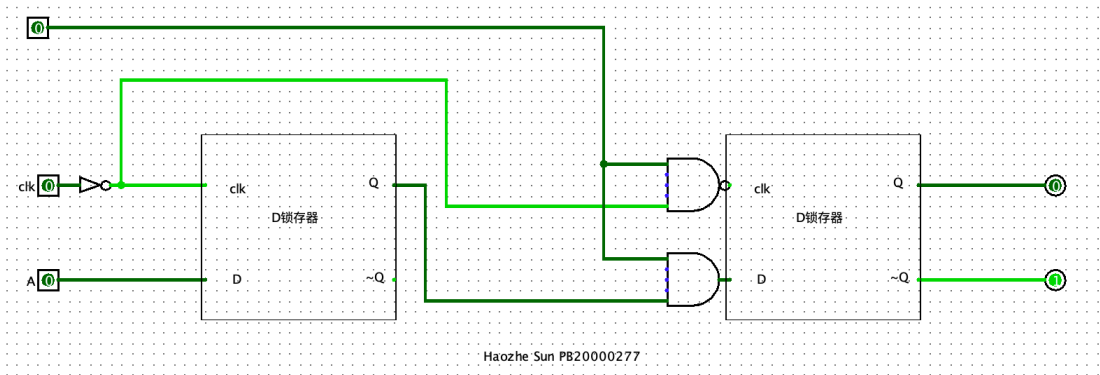
1. 我们首先搭建一个SR锁存器



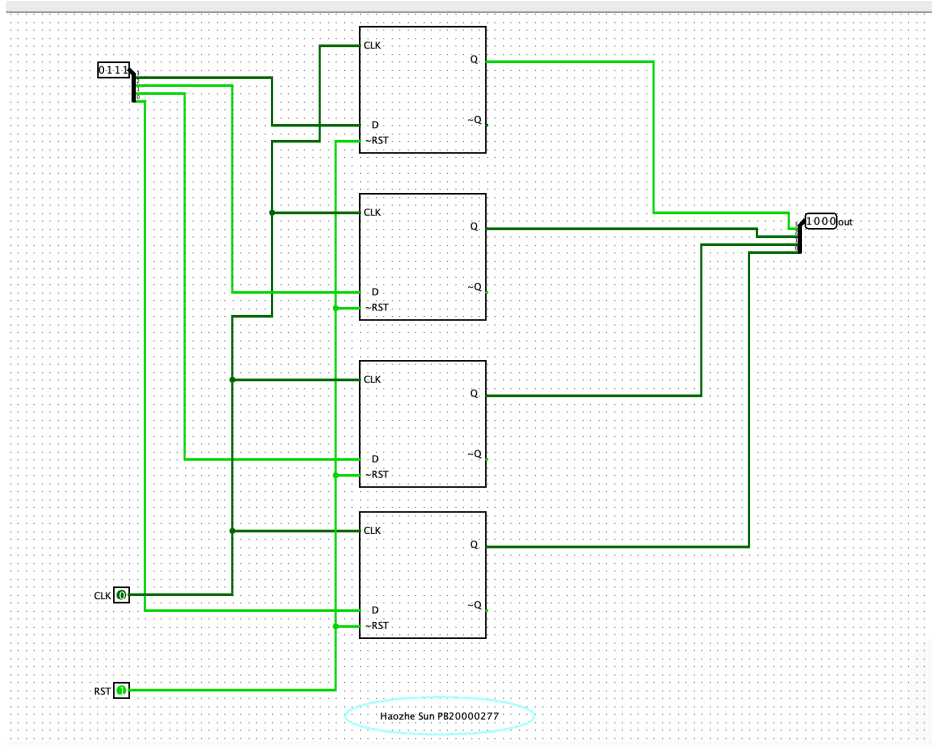
2. 其次我们用SR锁存器搭建一个D锁存器



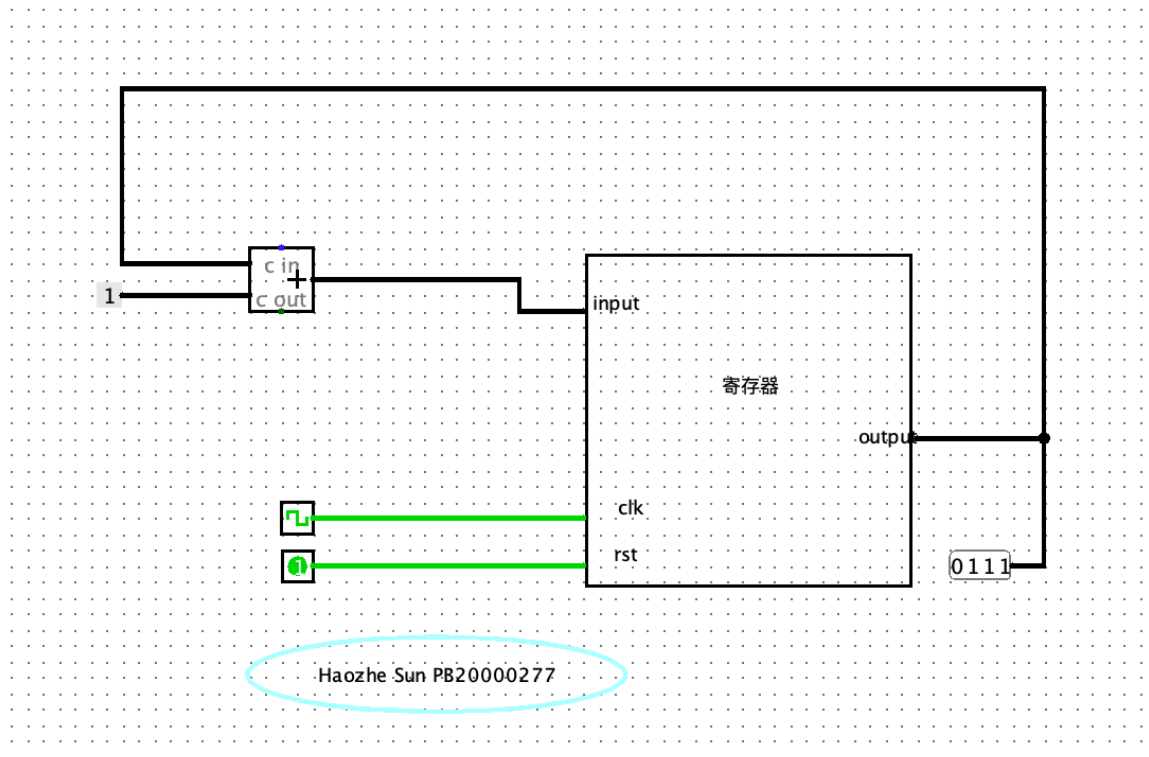
3. 然后我们利用这个D锁存器搭建一个具有异步复位功能的D触发器



4. 我们使用D触发器搭建一个4bit的寄存器



5. 再使用这个寄存器搭建0-15计数器即可



```

module d_ff_r(input clk, rst_n, d, output reg g);
    always@(posedge clk or negedge rst_n)
        begin
            if(rst_n==1'b0)
                q<=1'b0;
            else
                q<=d;
            end
        endmodule

module reg4(input clk, [3:0]d_in, rst_n, output[3:0] d_out);
    d_ff_r ff3(clk,d[3:3],rst_n,d_out[3:3]);
    d_ff_r ff2(clk,d[2:2],rst_n,d_out[2:2]);
    d_ff_r ff1(clk,d[1:1],rst_n,d_out[1:1]);
    d_ff_r ff0(clk,d[0:0],rst_n,d_out[0:0]);
endmodule

module counter(
    input clk,rst_n,
    output[3:0] g);
    wire [3:0]din;
    add add(d_out,4'b1,1'b0,din);
    reg4 reg4(clk,din,rst_n,g);
endmodule

```

T4

该题在检查的时候出现了一些诡异的bug，没有正确完成，后来重启了一下Logism就好了，助教让我在实验报告中调好后可以加一点分

- 1. 该题我们继续利用上一题所搭建的寄存器，不同的是，我们将不再利用寄存器自带的rst接口，而是选择了mux模块，重置的时候自动选择9
- 2. 同时我们需要搭建满足以下真值表的电路，来使得检测到结果为0后复位

S_1	S_2	S_3	S_4	S'_1	S'_2	S'_3	S'_4
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	1
0	1	1	0	0	1	1	0
0	1	1	1	0	1	1	1
1	0	0	0	1	0	0	0
1	1	1	1	1	0	0	1

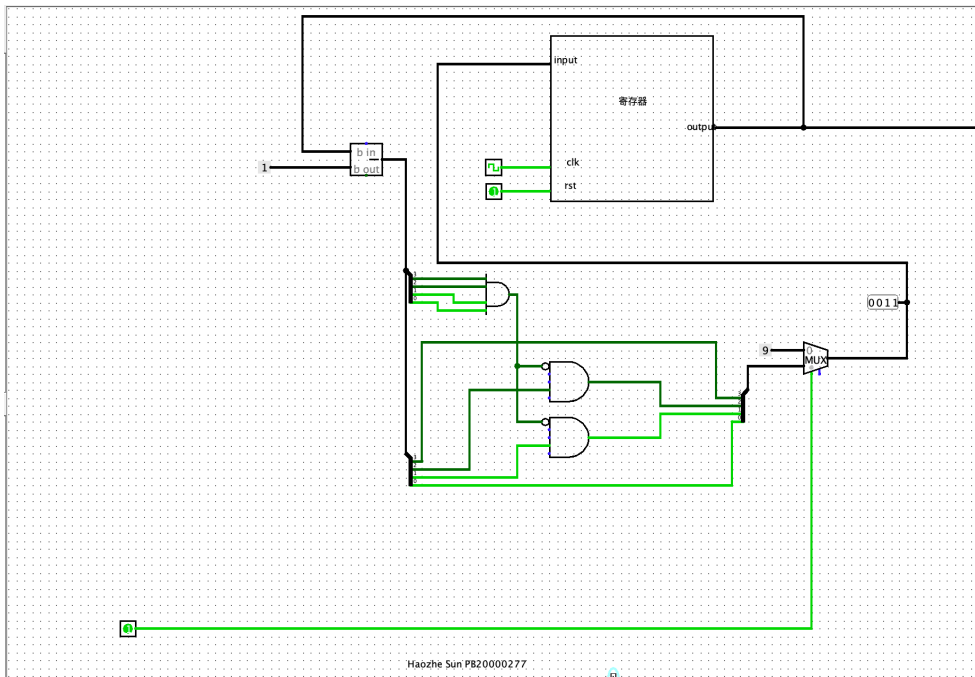
即，我们想让 1111 变成 1001 即可

- 3. 我们根据真值表，可以得出如下表达式：

$$\begin{aligned} S'_1 &= S_1 \\ S'_2 &= S_2 \cdot \overline{S_1 \cdot S_2 \cdot S_3 \cdot S_4} \\ S'_3 &= S_3 \cdot \overline{S_1 \cdot S_2 \cdot S_3 \cdot S_4} \\ S'_4 &= S_4 \end{aligned}$$

(1)

4. 我们根据这个表达式画出电路图

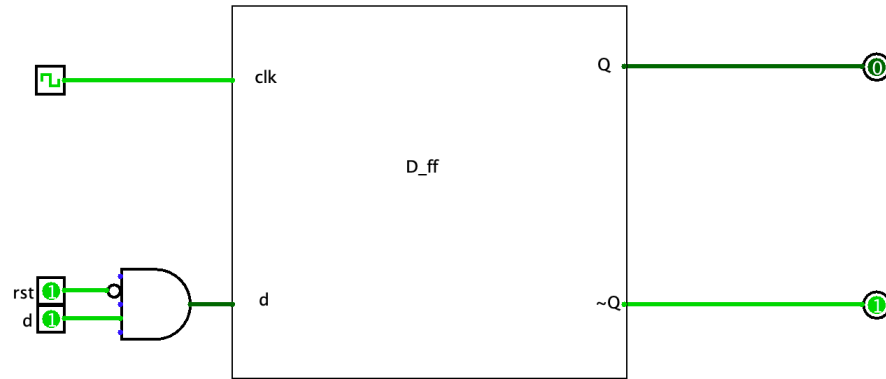


```
module counter(input clk, input rst_n, output reg[3:0] g);
    always@(posedge clk)
    begin
        if(rst_n==1'b0)
            g<=4'b1001;

        else
            begin
                if(g==4'b0000)
                    g<=4'b1001;
                else
                    g<=g-1;
            end
        end
    end
endmodule
```

T5

直接取反即可



Haozhe Sun PB20000277

```
module d_ff_s(input clk, rst, d, output reg q);  
    always@(posedge clk)  
    begin  
        if(rst==1)  
            q<=1'b1;  
        else  
            q<=d;  
        end  
    endmodule
```

总结与思考

- 本次实验掌握了使用logism和verilog搭建非逻辑电路的方法，同时学习到了很多关于异步和同步相关的知识
- 本次试验难度较大
- 本次试验任务量较重
- 无改进建议