# 中国科学技术大学计算机学院
# 《数字电路实验》报告


实验题目：简单时序逻辑电路

学生姓名：林宸昊

学生学号：PB20000034

完成日期：2021.10.28

计算机实验教学中心制

2020 年 09 月

【实验题目】简单时序逻辑电路

【实验目的】

1. 掌握时序逻辑相关器件的原理及底层结构;
2. 能够用基本逻辑门搭建各类时序逻辑器件;
3. 能够使用 verilog HDL 设计简单逻辑电路;
4. 能够使用 vivado 进行仿真。

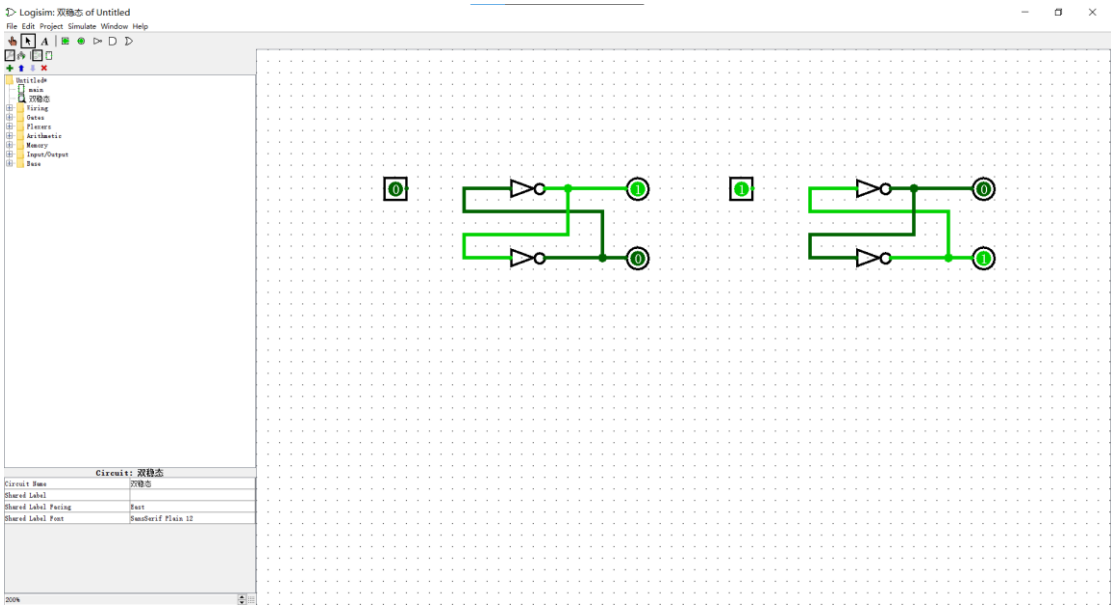
【实验环境】

1. 个人 pc 一台
2. Logisim 仿真工具
3. Vivado 实验平台

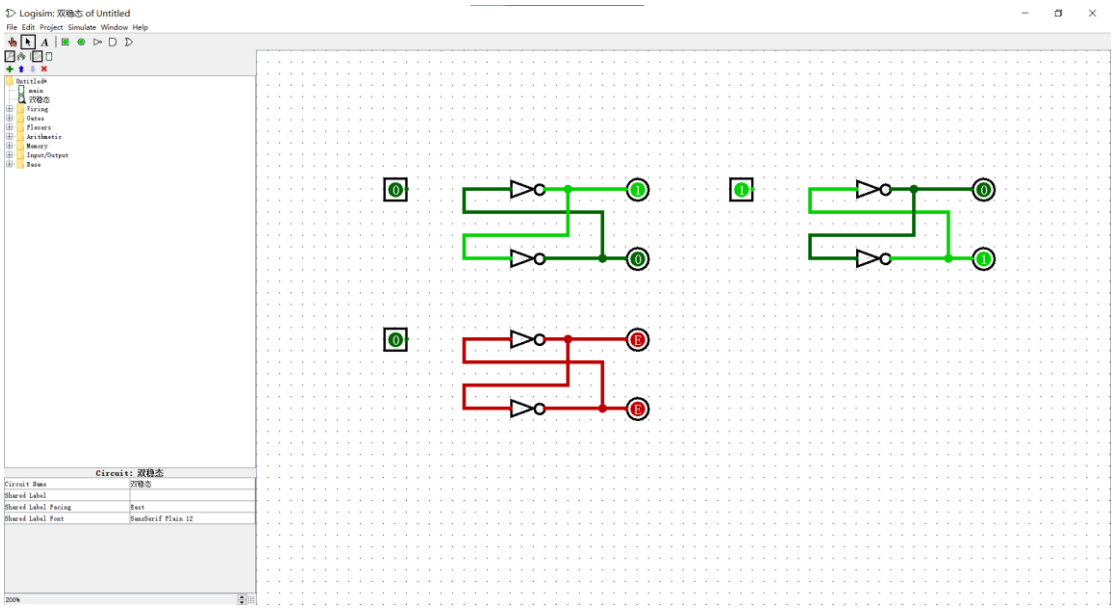
【实验过程】

Step1：搭建双稳态电路

双稳态电路由两个非门交叉耦合构成，虽然有完全一样的电路结构但可以保持两种恰好相反的状态，这是与组合逻辑电路本质区别的地方。

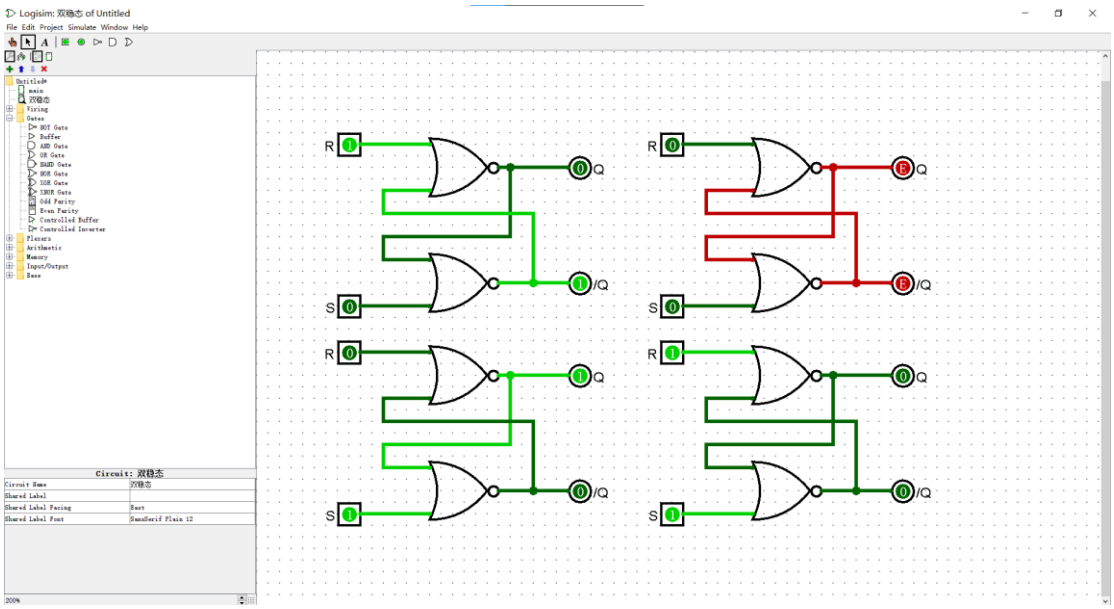


需要注意的是，应先使用输入信号确定其初始态后再将耦合线连上，否则电路将处于不确定状态（如图所示）



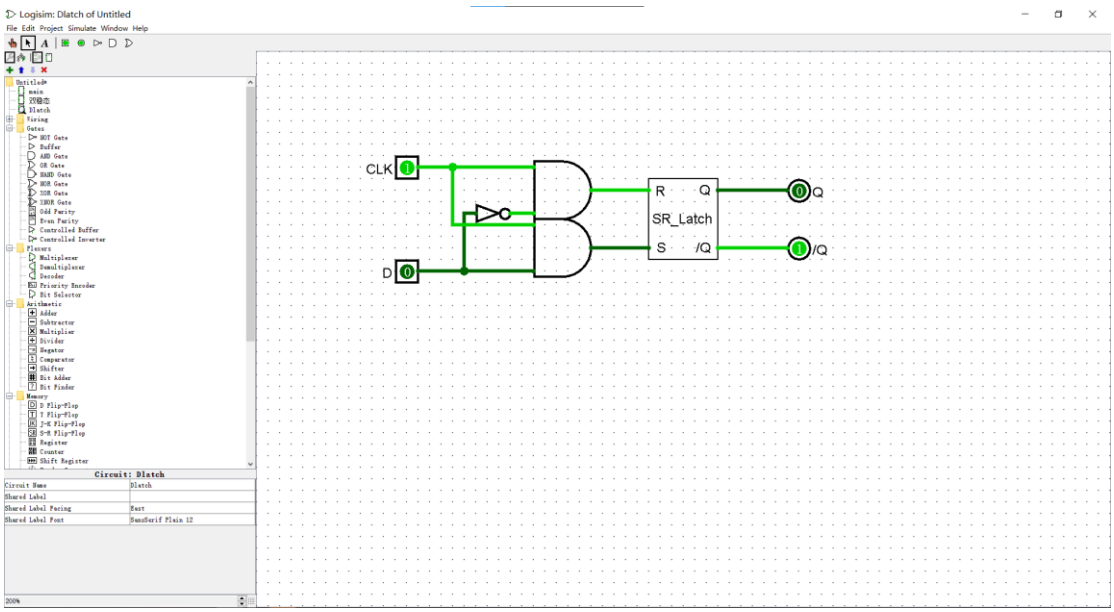
Step2：搭建 SR 锁存器

需要对上述的电路进行修改，使其能接受输入信号。输入信号 S 负责对 Q 置位(set)，输入信号 R 负责对 Q 复位（reset)。二者都有效时 Q 与/Q 均为 0，矛盾，认为这是一种未定义状态，尽量避免发生。

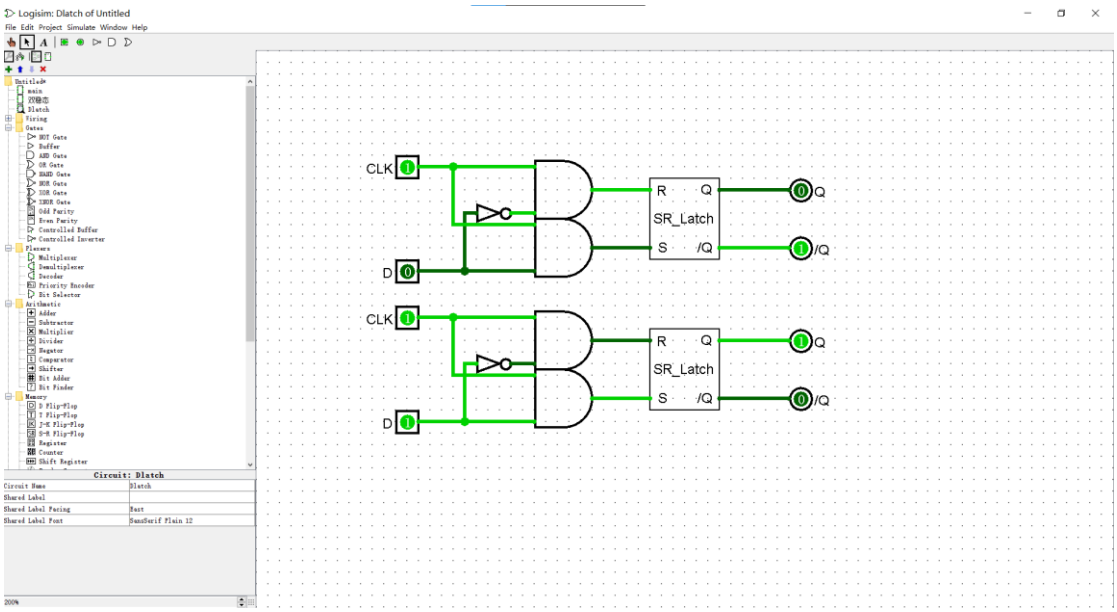


Step3：搭建 D 锁存器

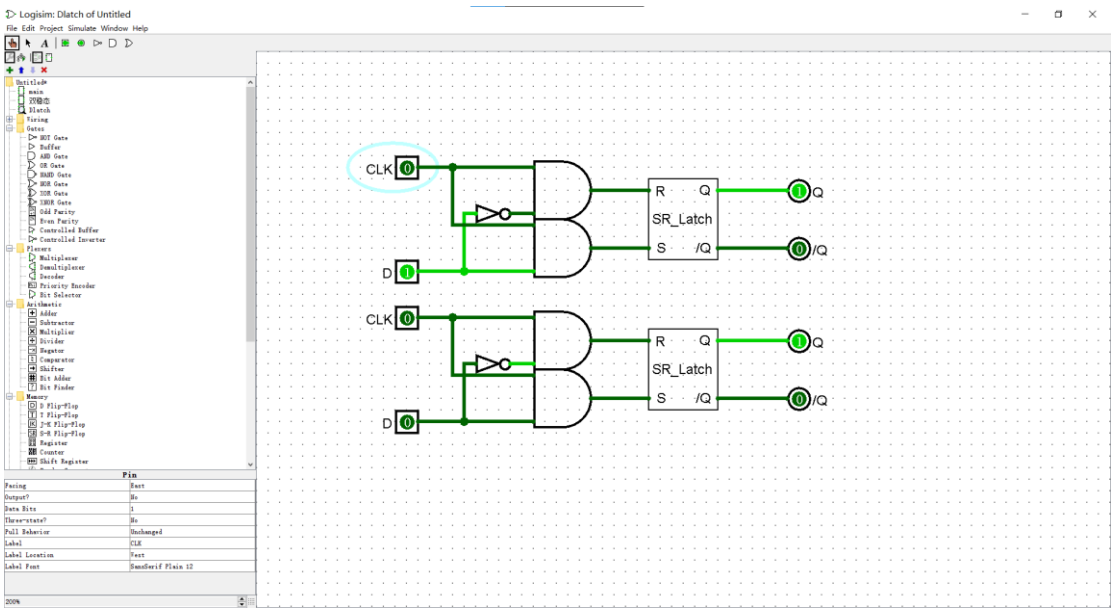
上述的锁存器在 SR 均为 1 时处于未定义状态，为更好的避免这种状态，可以构建 D 锁存器。




如图，当 CLK 信号为高电平时，Q 信号随着 D 变化而变化，称为跟随状态。CLK 低电平时，Q 保持之前的值，不会受到影响，称为锁存状态。



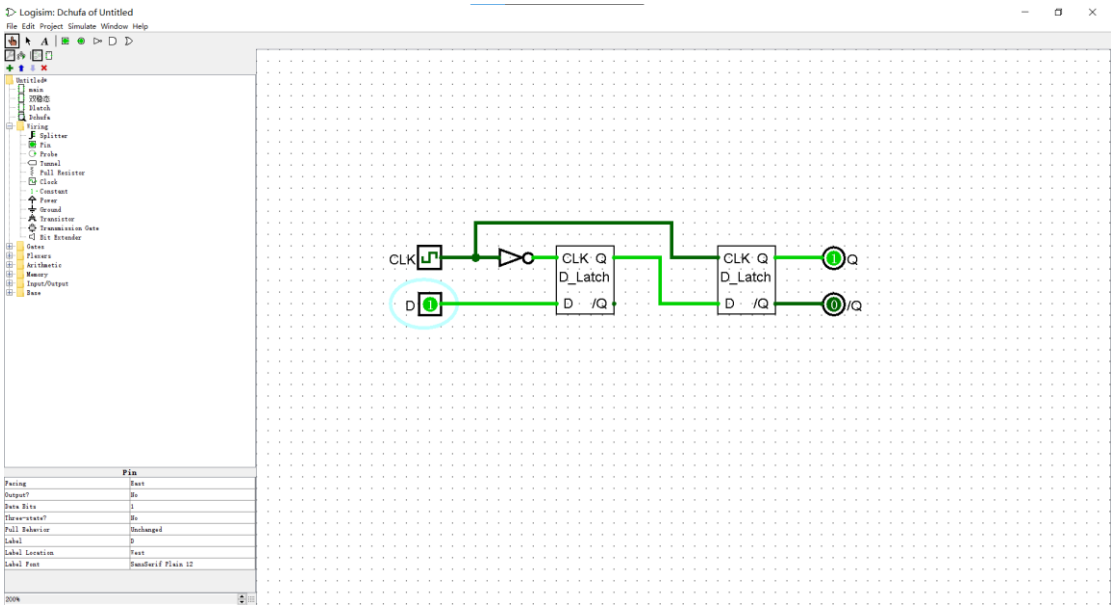

跟随




锁存

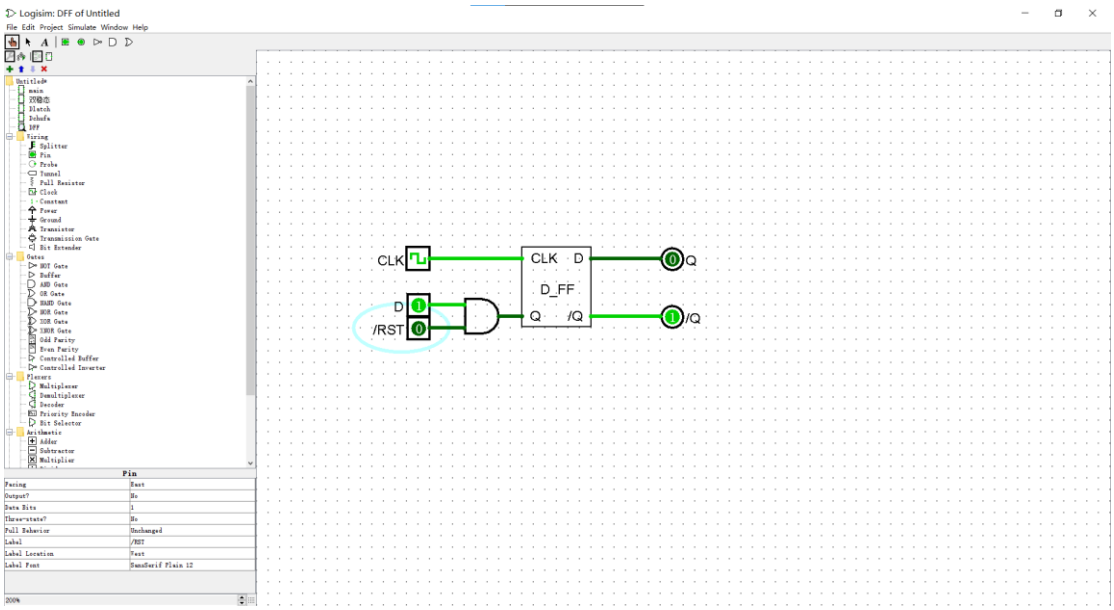
Step4：搭建 D 触发器

将两个 D 触发器串联起来，使其控制信号有效值相反，则构成 D 触发器。CLK 为低电平时，D 到达 D1，CLK 由低变高时，D2 打开，信号到达 Q。

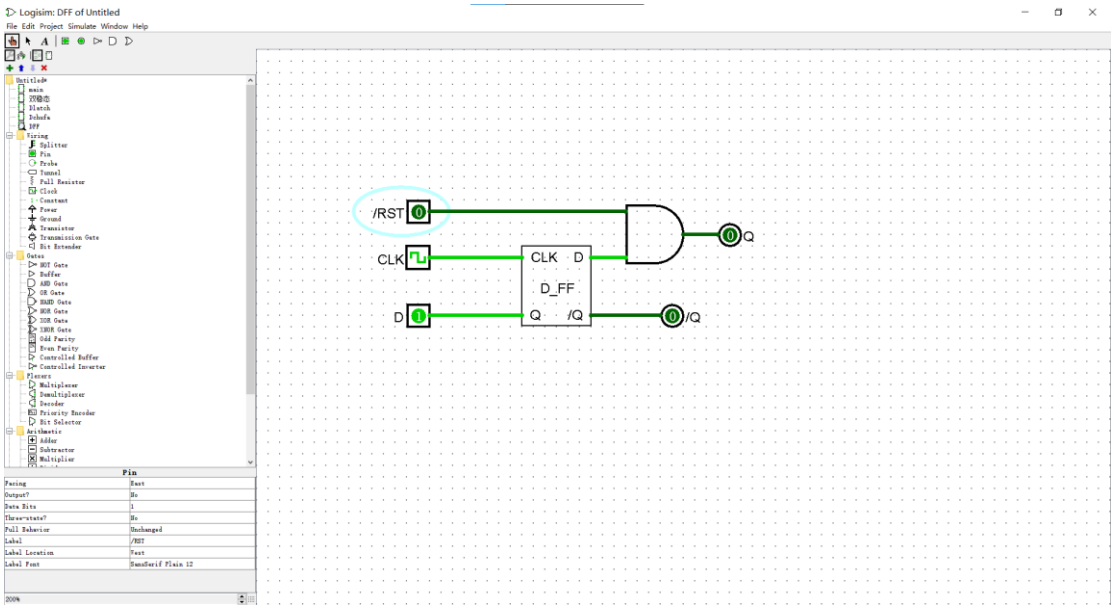



只有 CLK 信号从低电平变为高电平时，D 信号才会传播到 Q 端。

在此基础上，还可以添加复位信号，当复位信号为低电平时，输出信号 Q 始终为 0



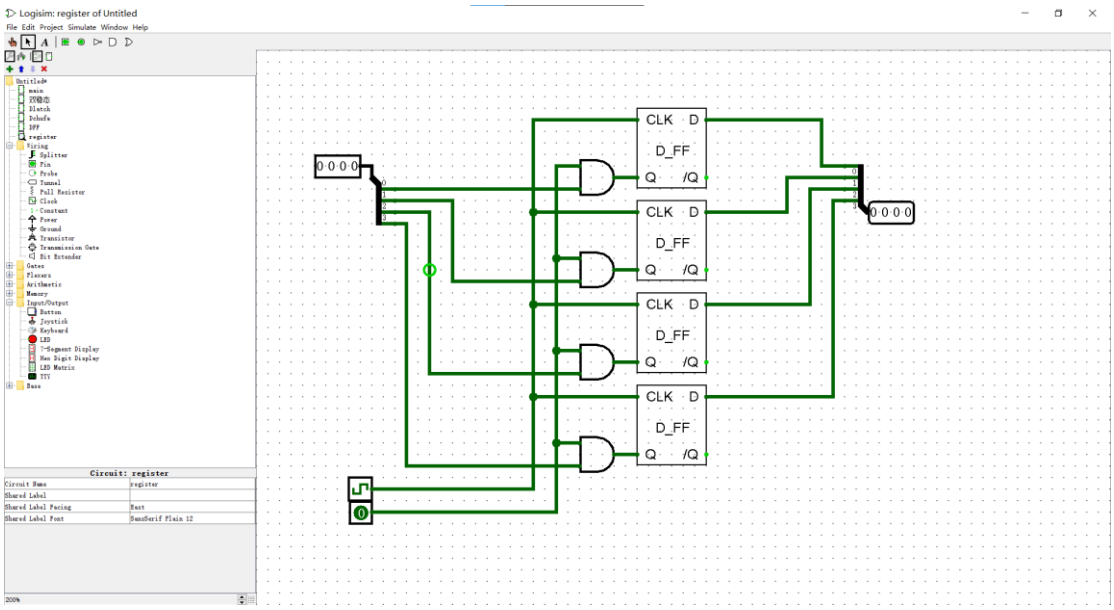

这种触发器的复位信号只在时钟信号的上升沿才起作用，称为同步复位。相对的，还有一种异步复位方式，不论时钟信号与 D 信号如何，只要其有效，输出端 Q 理科变为确定的复位值。



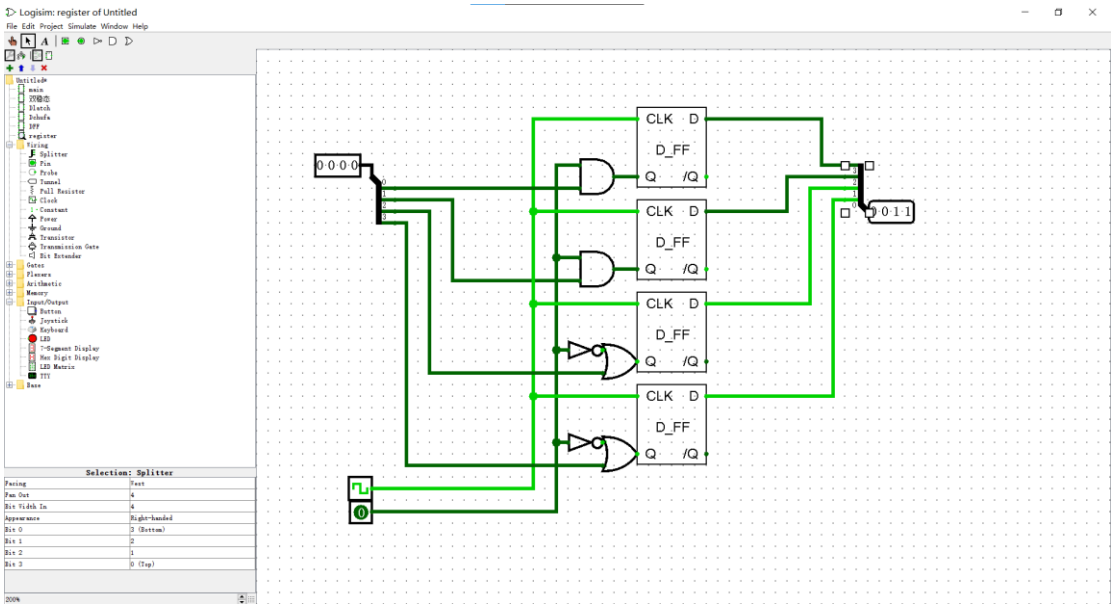

复位信号不再完全与时钟信号上升沿同步，故称为异步信号。

Step5：搭建寄存器

本质来说就是 D 触发器。用 4 个 D 触发器可以构成一个能够储存 4bit 数据的寄存器，同时带有低电平复位信号。

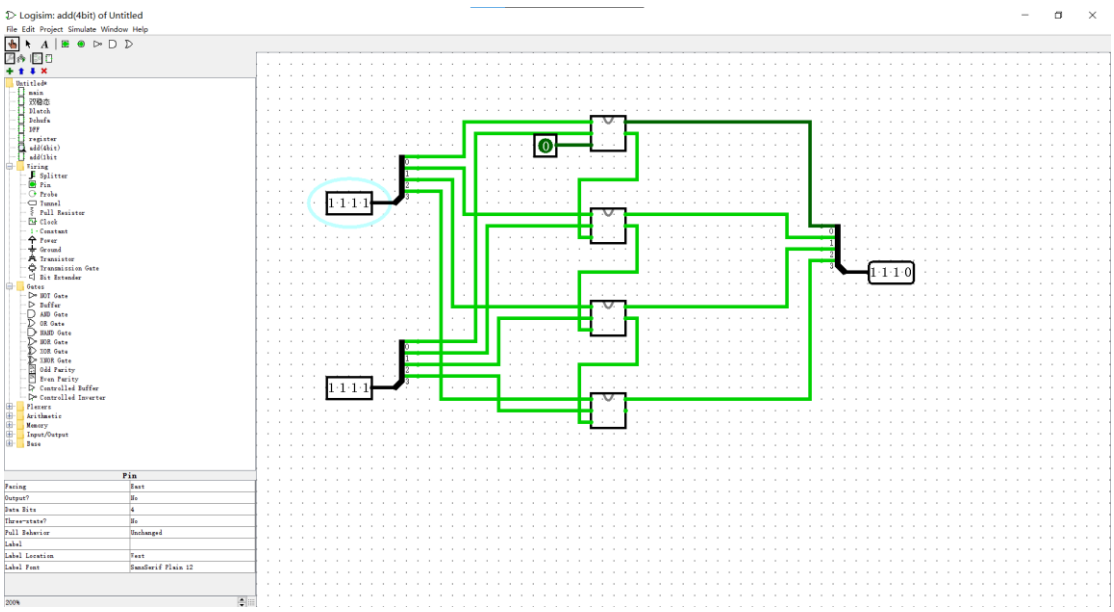



若想要改变复位时信号的输出，如改成 4’b0011，则可以做如下修改

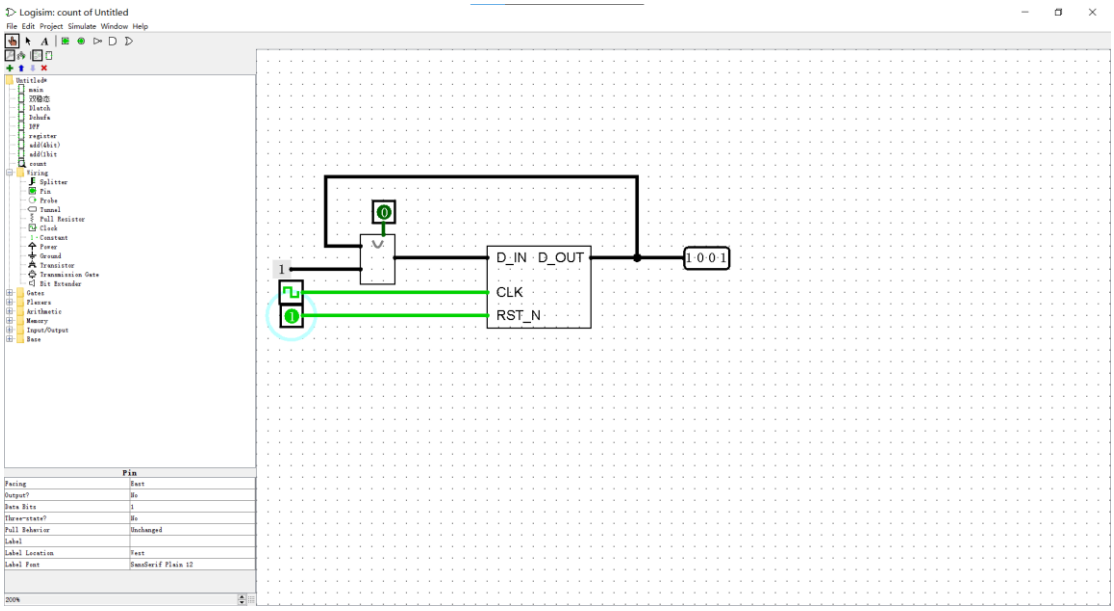



Step6：简单时序逻辑电路的搭建

利用 4bit 寄存器可以搭建一个 4bit 计数器。先设计一个 4bit 加法器

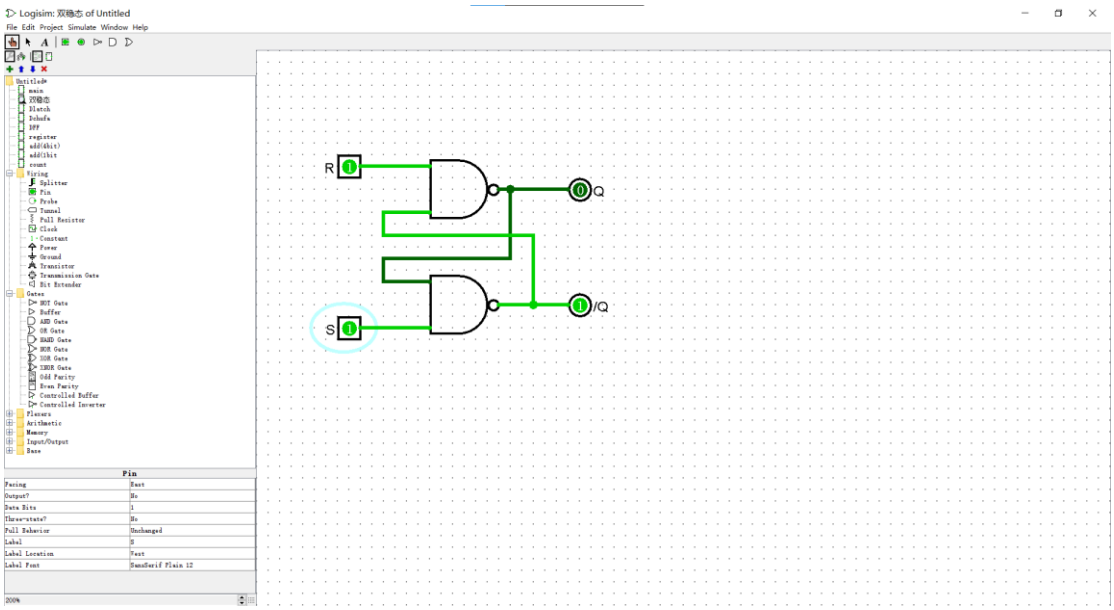



再封装得到 4 位计数器

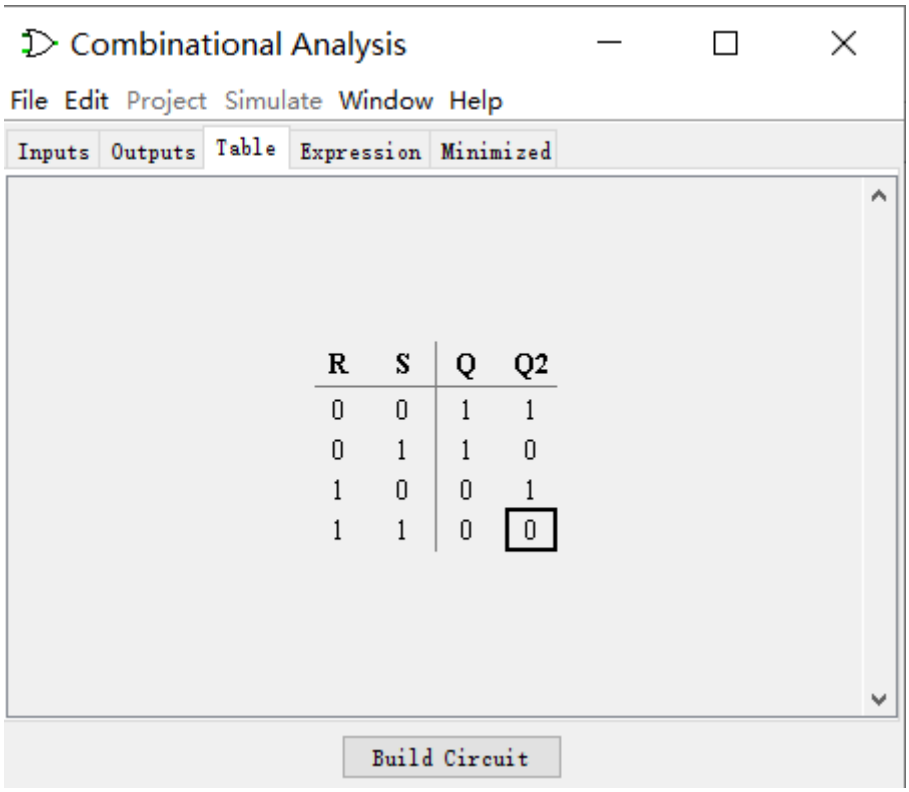



【实验练习】

题目 1：用与非门搭建 SR 锁存器




真值表如下：

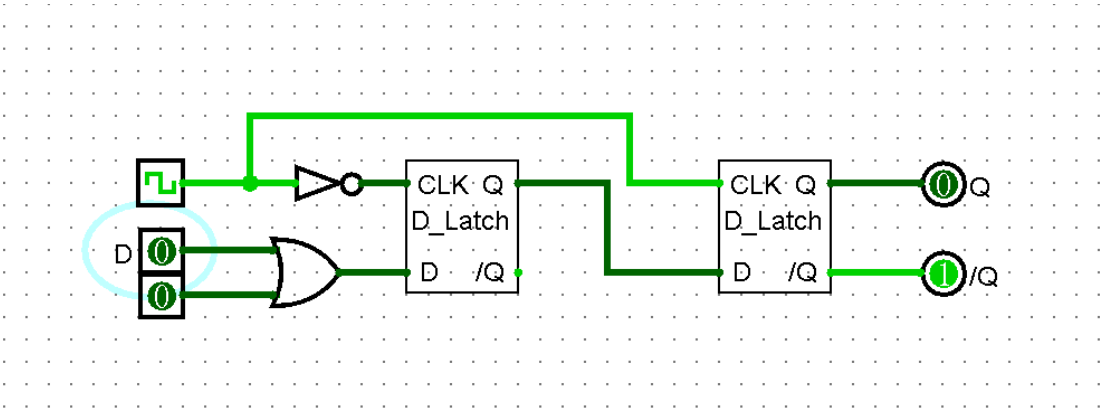



S = R = 0 时，电路状态保持不变；R = 0,S = 1 时，Q 置 1；R = 1,S = 0 时，Q 置 0；S = R = 1 时，电路视为是不确定态，即未定义状态。

| S | R | Q | /Q | 状态 |
|---|---|---|---|---|
| 0 | 0 | x | x | 保持 |
| 1 | 0 | 1 | 0 | 置 1 |
| 0 | 1 | 0 | 1 | 置 0 |
| 1 | 1 | 0 | 0 | 未定义 |

题目 2：支持同步置位 D 触发器

画出电路图：

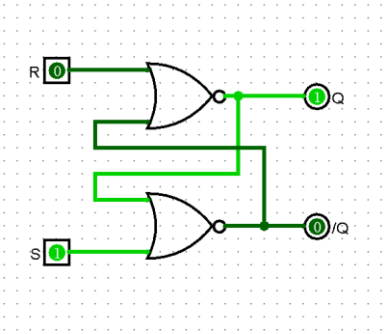



编写 verilog 代码

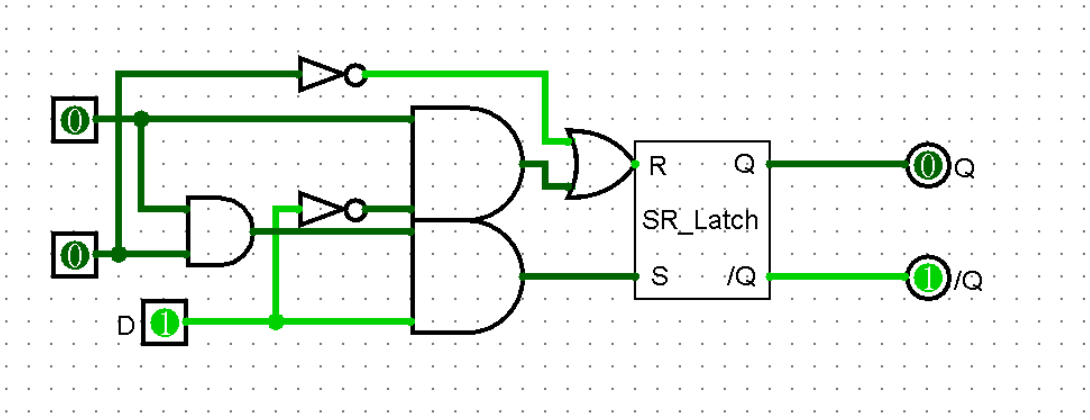
```verilog
module d_ff(
  input clk,
  input rst,
  input d,
  output q);
  always @ (posedge clk)
    begin
      if(rst == 1)
        q <= 1;
      else
        q <= d;
    end
endmodule
```

题目 3：异步复位 D 触发器

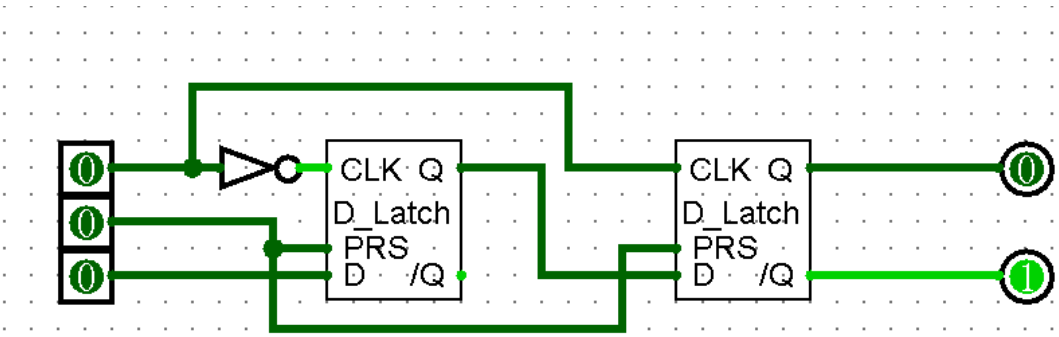
Step1：先画出 SR 锁存器



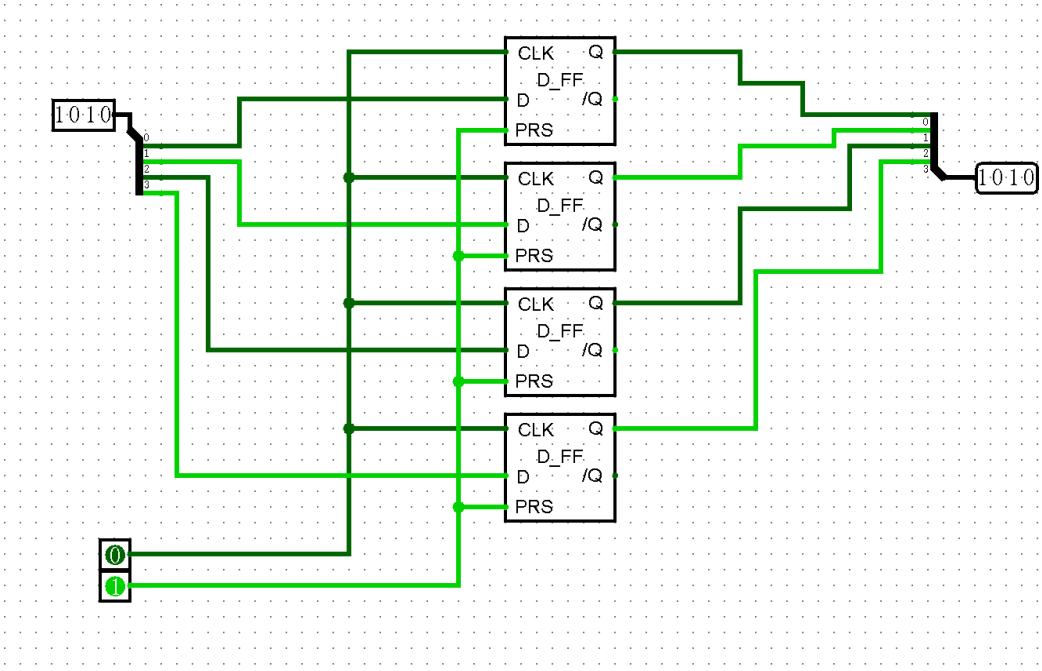

Step2：再由此画出异步复位 D 锁存器



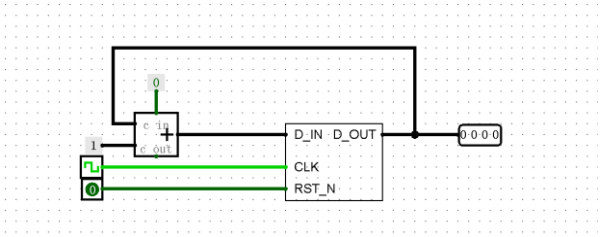

Step3：再由此画出异步复位 D 触发器




Step4：再由此画出异步复位 4bit 寄存器




Step5：再由此画出 4bit 循环计数器（利用加法器模块）

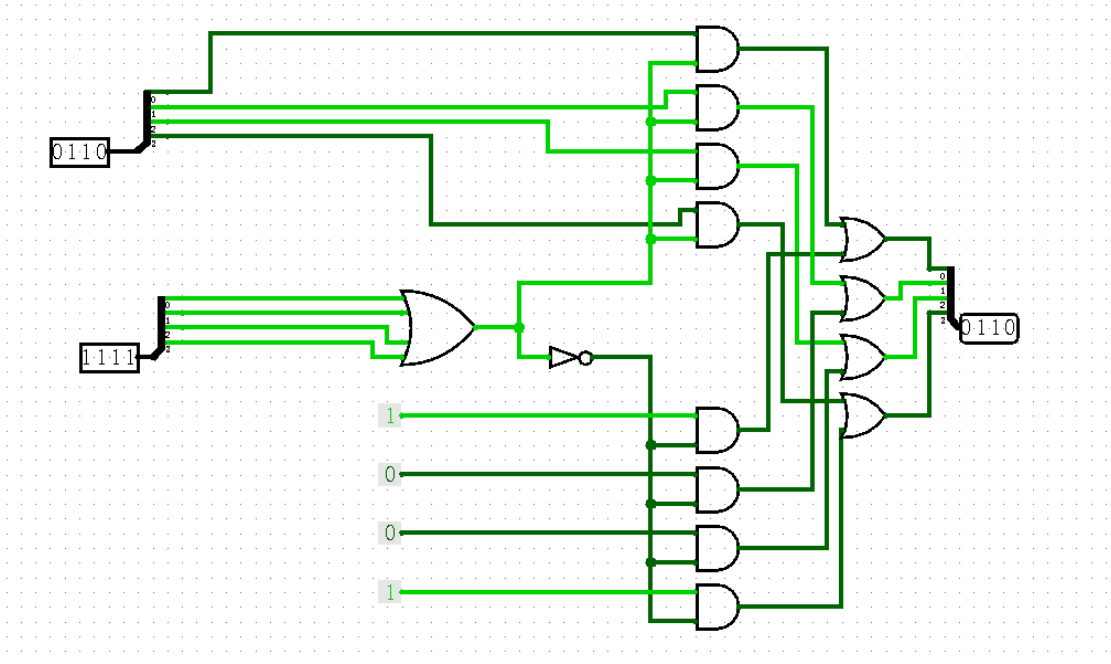



Step6：编写 verilog 代码

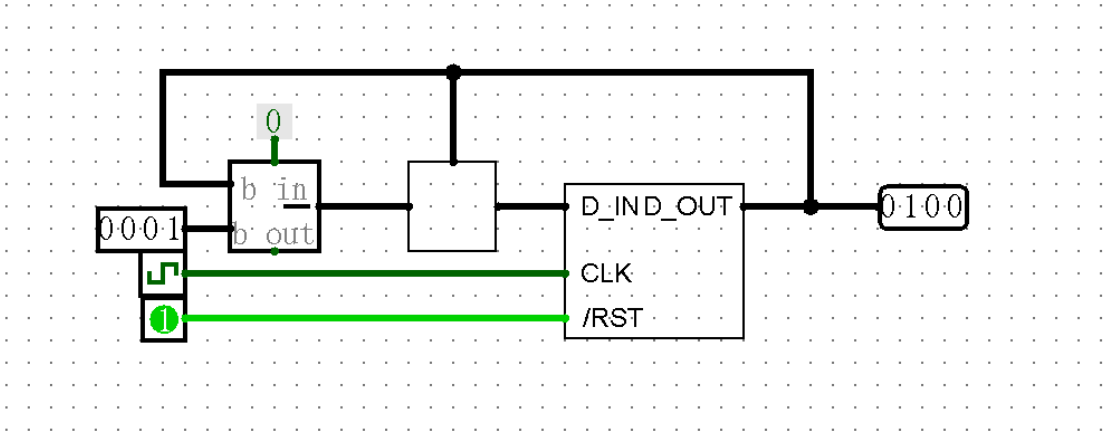
```verilog
module d_ff(
  input clk,
  input rst,
  input d,
  input prs,
  output [3:0] cnt);
  always @ (posedge clk or negedge prs)
    begin
      if(!prs)
        cnt <= 4'b0000;
      else
        cnt <= cnt + 1;
    end
endmodule
```

题目 4：9~0 循环递减计数器

由于需要在计数器为 0 时完成重置，可以设计一个特殊选择器，使得选择信号全为 0 时使输出 1001，否则输出跟随输入变化。




而要使其递减，可以使用减法器，最终可以得到电路图如下：

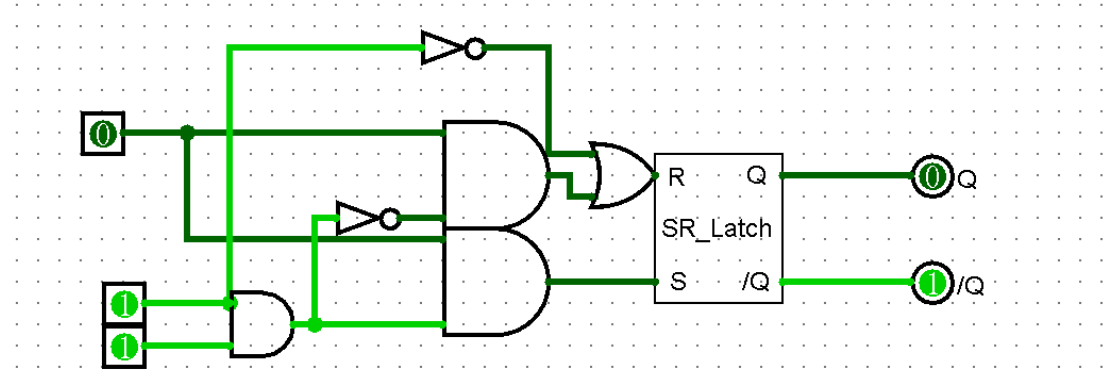



编写对应 verilog 代码：

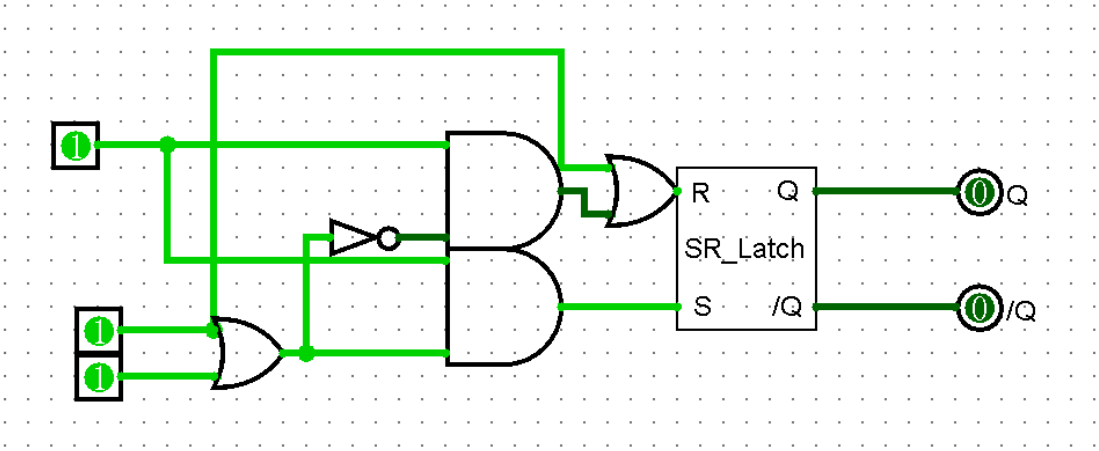
```
module d_ff(
  input clk,
  input d,
  input prs,
  output [3:0] cnt);
  always @ (posedge clk or negedge prs)
    begin
      if(!prs)
        cnt <= 4'b1001;
      else
        begin
          if(cnt == 0)
            cnt <= 4'b1001;
          else
            cnt <= cnt - 1;
        end
    end
endmodule
```

题目 5：

首先看低电平有效的异步复位电路：




这是一个异步复位的 D 锁存器，低电平有效。若需要使其高电平有效，可对门级电路做相应更改：




此时复位信号即为高电平有效。

编写 verilog 代码:

```
module d_ff(
  input clk,
  input d,
  input prs,
  output q);
  always @ (posedge clk or posedge prs)
    begin
      if(prs)
        q <= 0;
      else
        q <= d;
    end
endmodule
```

【总结与思考】

1. 实验收获

   学会了简单时序逻辑电路的设计，理解了一些简单时序电路的原理。后者最为重要，也对我帮助很大。我的的确确理解了很多课堂上未能搞懂的知识——原本艰涩的原理通过实验变得极其明了。同时还有同步置位，同步复位，异步复位的电路搭建方式；

2. 实验难易及任务量

   较难，任务量较大。相较前两次实验难度跨度较大，尤其是异步复位电路的搭建上。但是总体上来说犯了很多理解上的错误以及基础知识掌握不牢的原因，使得花费了很多时间在研究怎么异步复位，怎么改变复位值，怎么使其变为高电平复位上——其实如果熟练掌握 SR 锁存器特性的话这些都不是难题。不过好在做完了并且获益匪浅。

3. 实验建议

   第五题从 0000 跳转至 1001 我使用了数据选择器——但我并不理解这是否是考察范围，或者说在考察范围之中，因为这事实上是较之原有内容额外多出来的任务，真正的考察点是否应该在复位至特殊值上（如 1001 而非 0000）而不是将 0000 跳转至 1001。除此之外暂无。

*其实实验要求可以写得更具体一点，真的*