**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：\_\_\_\_ 黄瑞轩\_\_\_\_\_

学生学号：\_\_ PB20111686\_\_\_

完成日期：\_\_ 2021.10.24 \_\_

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**

* 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

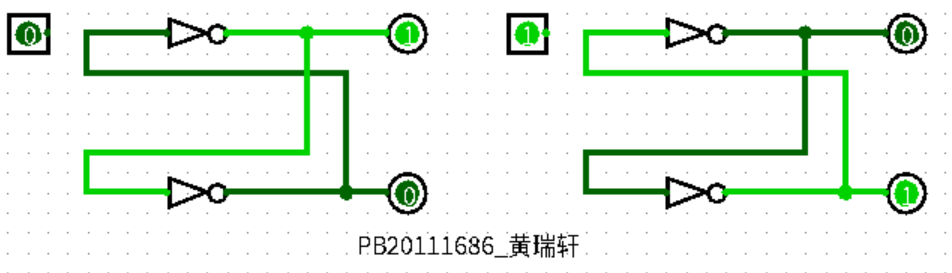
**实验环境**

* PC一台：安装了Linux操作系统的一台虚拟PC
* VLAB实验中心平台（vlab.ustc.edu.cn）
* Logisim Version 2.7.1

**实验过程**

1. 搭建双稳态电路

* 双稳态电路由两个非门交叉耦合构成，是一种最简单的时序逻辑电路，是所有时序逻辑电路的基础。
* 在Logisim中搭建电路时应先将两条交叉耦合线断开一条，等输入信号将其状态初始到确定状态后再将耦合线连上。否则电路将处不确定状态。



1. 搭建SR锁存器

* SR锁存器的功能已在理论课上说明，这里不多记录。
* 用Logisim搭建电路，并改变输入输出，观察电路仿真情况是否与理论一致。
* 理论分析得出R和S同时置1时电路会处于不确定状态，因此要避免这种情况。

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 初始状态  *Q*=1, /*Q*=0 | Reset状态  *Q*=0, /*Q*=1 | Set状态  *Q*=1, /*Q*=0 | 不确定状态  *Q*=0, /*Q*=0 |

* 将此电路封装为“SR\_Latch”以便复用。

1. 搭建D锁存器

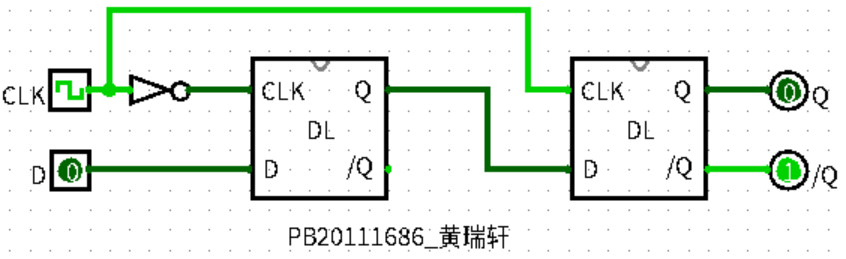
* SR锁存器两个输入都为1是一种未定义状态，为了避免这种情况出现设计了下面这种D锁存器。D锁存器的理论分析也已在理论课上说明，这里不多记录。
* 用Logisim搭建电路，并改变输入输出，观察电路仿真情况是否与理论一致。

|  |  |
| --- | --- |
|  |  |
| *D*=1, *CLK*=0 保持 | *D*=0, *CLK*=0 保持 |
|  |  |
| *D*=1, *CLK*=1 跟随 | *D*=0, *CLK*=1 跟随 |

* 将此电路封装为“DL”以便复用。

1. 搭建D触发器

* 用D锁存器搭建D触发器的电路如下图所示。



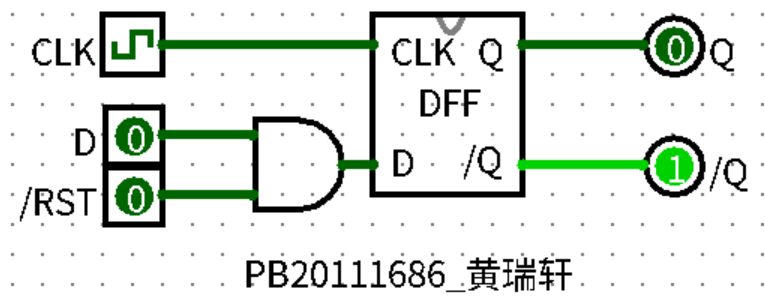
* Logisim菜单栏—“simulation”—“Tick Frequency”设置为“1Hz”，然后Enable仿真和触发功能，在“CLK”信号以 1Hz 频率跳变过程中，改变 D 信号的输入值，观察Q信号的输出。
* 其Verilog代码如下所示，“posedge CLK”表示“CLK信号的上升沿”这一事件。

module DFF(input clk, d, output reg q);

always@(posedge clk) q <= d;

endmodule

* [同步复位] 添加复位信号/RST，当/RST有效时，Q始终为0。这种复位信号只有在时钟信号的上升沿才起作用，在非上升沿时刻不起作用。其电路和Verilog代码如下所示。



module DFF\_R(input clk, d, rst, output reg q);

always@(posedge clk)

begin

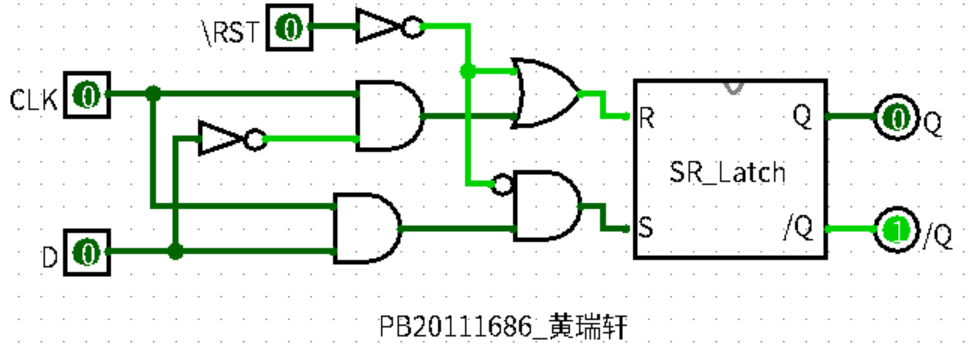
if(rst == 0) q <= 1’b0;

else q <= d;

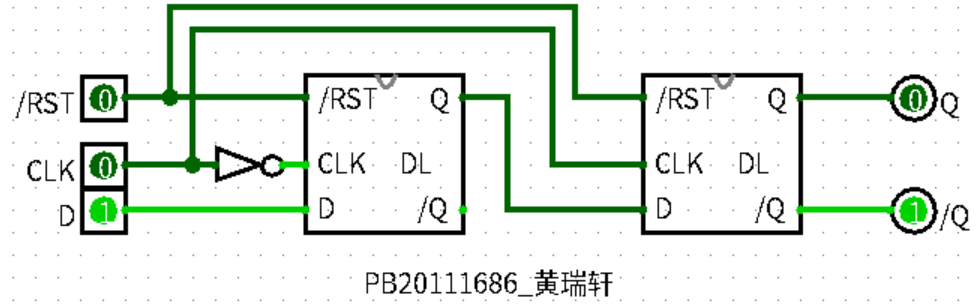
end

endmodule

* [异步复位] 添加复位信号/RST，当/RST变为有效时(负电平)，Q立即变为0，无论CLK和D如何。
* 首先，制作一个/RST负有效的D锁存器，电路图如下。



* 再用相似的结构做D触发器，电路图和Verilog代码如下。



module DFF\_R(input clk, d, r, output reg q);

always@(posedge clk or negedge r)

begin

if(r == 0) q <= 1’b0;

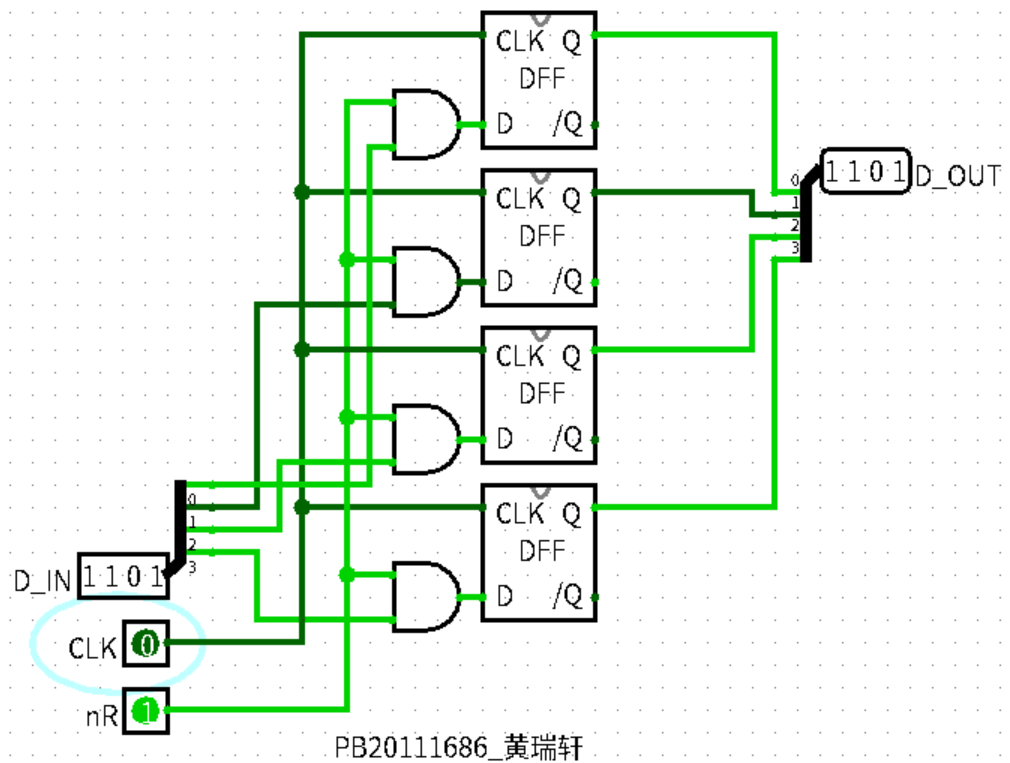
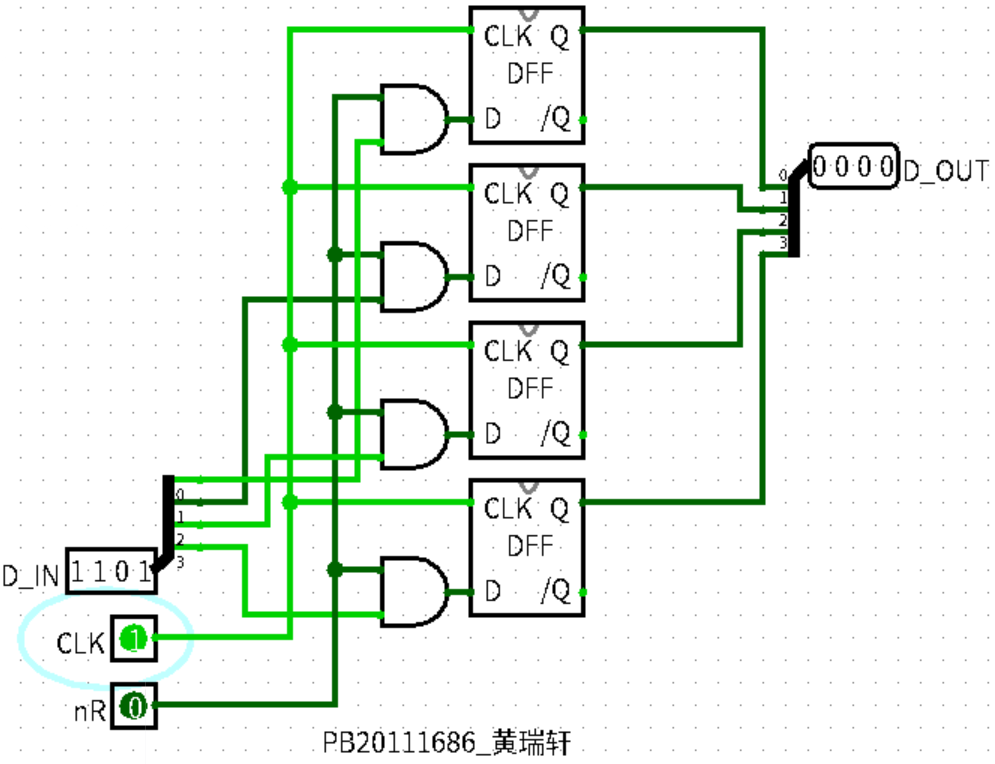
else q <= d;

end

endmodule

1. 搭建寄存器

* 用4个D触发器组成一个能够存储4bit数据的寄存器，带有低电平有效的同步复位信号，其电路结构和Verilog代码如下所示，可以看出寄存器本质上就是D触发器。

module REG4(input clk, [3:0] d\_in, rst, output reg [3:0] q);

always@(posedge clk)

begin

if(rst == 0) q <= 4’b0;

else q <= d;

end

endmodule

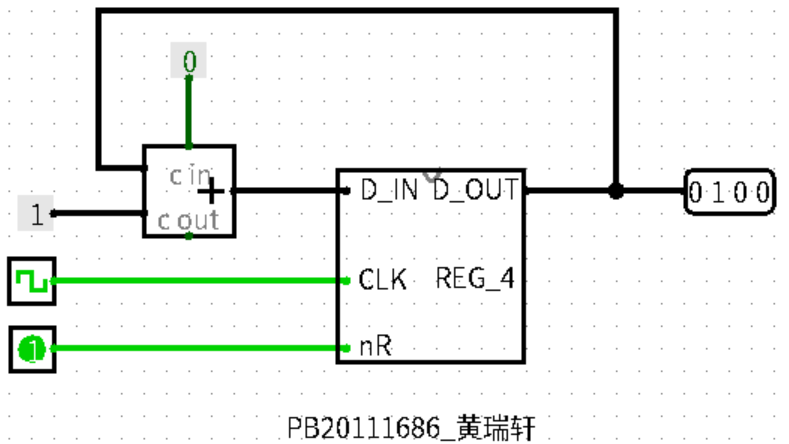
* 有些情况下信号的复位值需要为1，如将上述代码复位部分改成

if(rst == 0) q <= 4’b0011;

后电路应将低两位的D触发器的  部分改成  ，此时如果nR有效，D将成为1，电路稳定后相应输出位将为1。如果nR无效，D仍是D\_IN中的相应位，这是一种同步复位方法。

1. 搭建简单时序逻辑电路

* 利用4 bit寄存器，搭建一个4 bit的计数器，该计数器在0~15之间循环计数，复位时输出值为0，其电路结构和Verilog代码如下所示。



module CTR(input clk, rst, output reg [3:0] cnt);

always@(posedge clk)

begin

if(rst == 0) cnt <= 4’b0;

else cnt <= cnt + 4’b1;

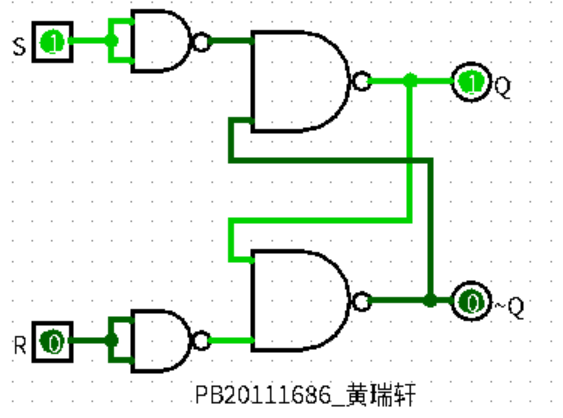
end

endmodule

**实验练习**

**【题目1】**

* 在Logisim中用与非门搭建SR锁存器，其电路图如下所示。



* 写出逻辑表达式：。
* 当S = 0, R = 0时，电路将保持原来的状态；

当S = 1, R = 0时，Q将先变成1，随后~ Q随其变成0（置1）；

当S = 0, R = 1时，~ Q将先变成1，随后Q随其变成0（置0）；

当S = 1, R = 1时，电路下一个状态不确定。

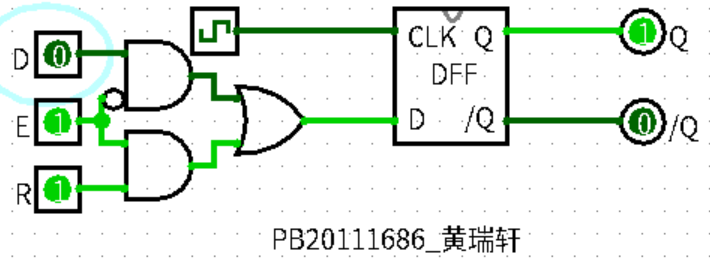
以表列出如下，这与普通SR锁存器的功能表相同。

|  |  |  |  |
| --- | --- | --- | --- |
| S = 0, R = 0 | S = 1, R = 0 | S = 0, R = 1 | S = 1, R = 1 |
| 保持 | 置1 | 置0 | 不确定 |

**【题目2】**

* 设计同步置位功能的D触发器只需要修改D的逻辑。新增输入端E和R，当E为0时置位功能无效，当E为1时启动置位功能，D输入端将输入R。写成逻辑式就是：

设计成电路图如下：



其Verilog代码如下：

module DFF\_S(input clk, d, e, r, output reg q);

always@(posedge clk)

begin

if(e == 1’b1) q <= r;

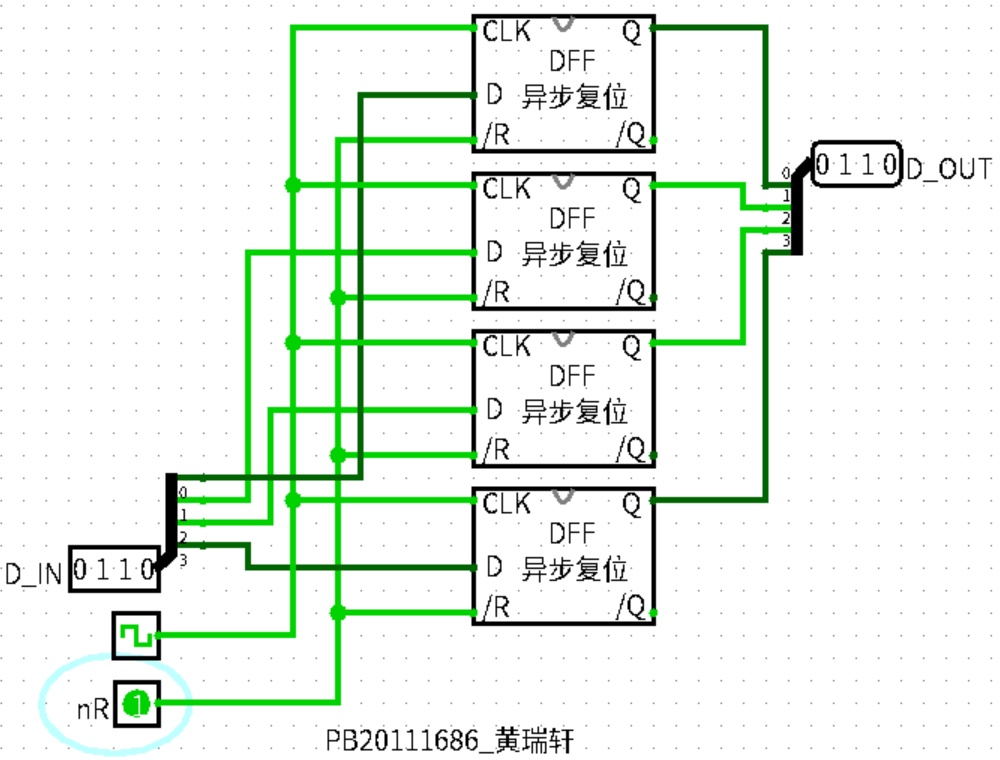
else q <= d;

end

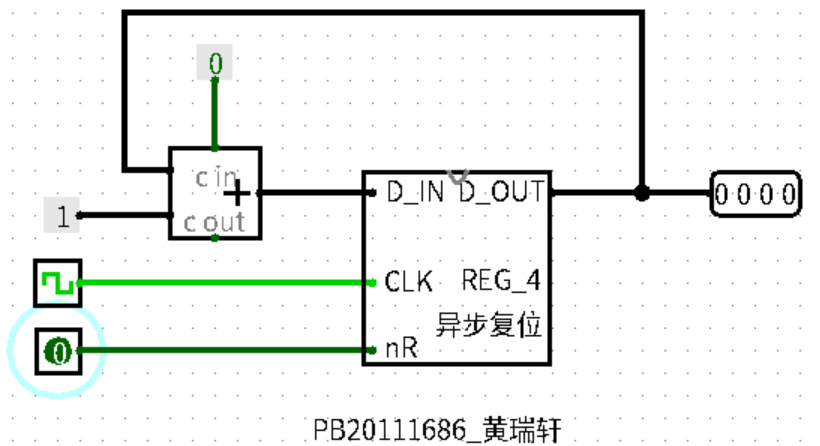
endmodule

**【题目3】**

* 电路设计和Verilog代码已在实验过程(4.)[异步复位]中体现，这里不再重复，仅完成后半部分。先将寄存器中所有的触发器换成异步复位的，如下图所示。



* 计数器的电路与Verilog代码如下：



module CTR\_R(input clk, rst, output reg [3:0] cnt);

always@(posedge clk or negedge r)

begin

if(rst == 0) cnt <= 4’b0;

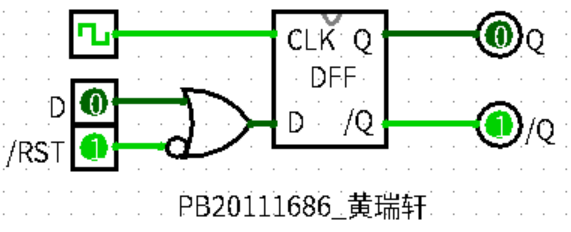
else cnt <= cnt + 4’b1;

end

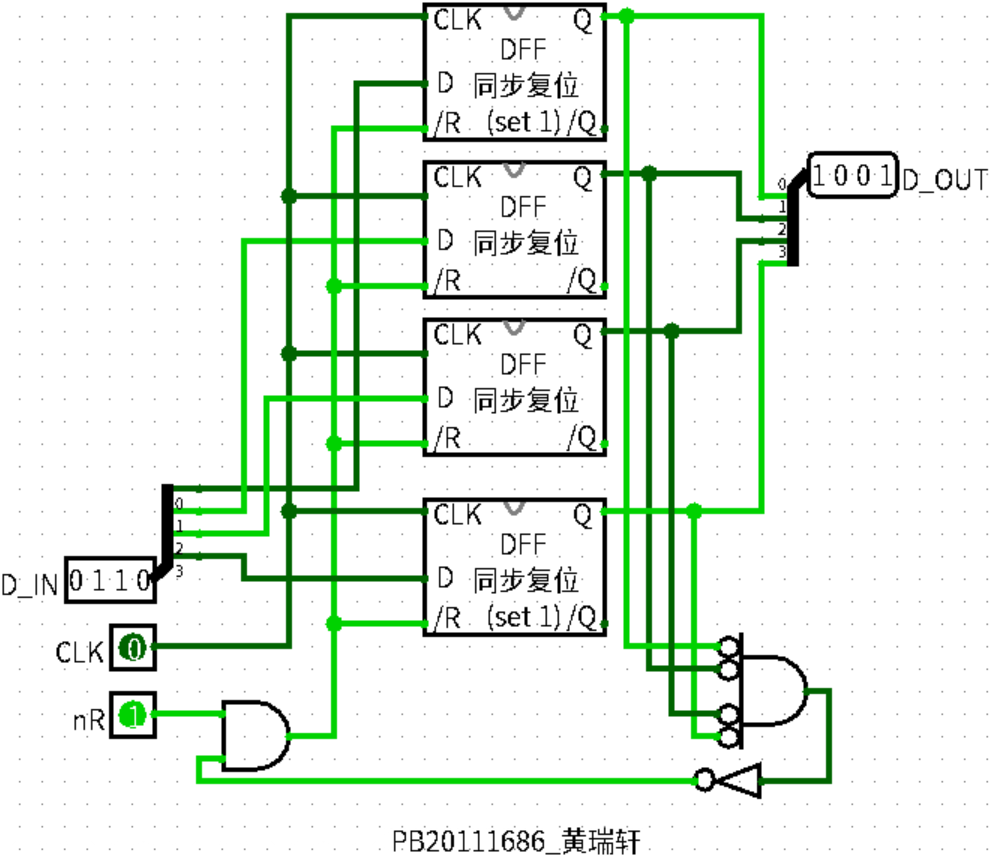
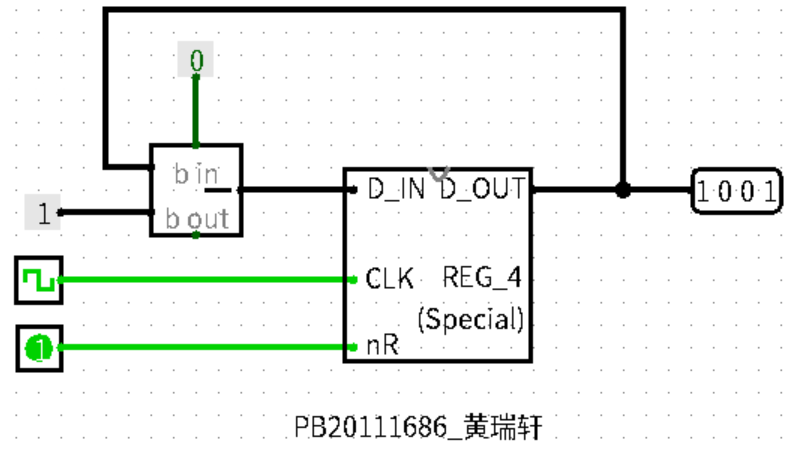
endmodule

**【题目4】**

* 首先做一个低电平有效且复位置1的触发器，电路图如下：



然后用此触发器实现4位寄存器。由于需要的数据范围是0000 ~ 1001，所以只要复位信号nR为0或者当前状态是0000时，下一个状态就要复位，寄存器设计如下左图所示，最终计数器电路如下右图所示，其Verilog代码附后。

module MNR(input clk, rst, input [3:0] d, output reg [3:0] q);

always@(posedge clk)

begin

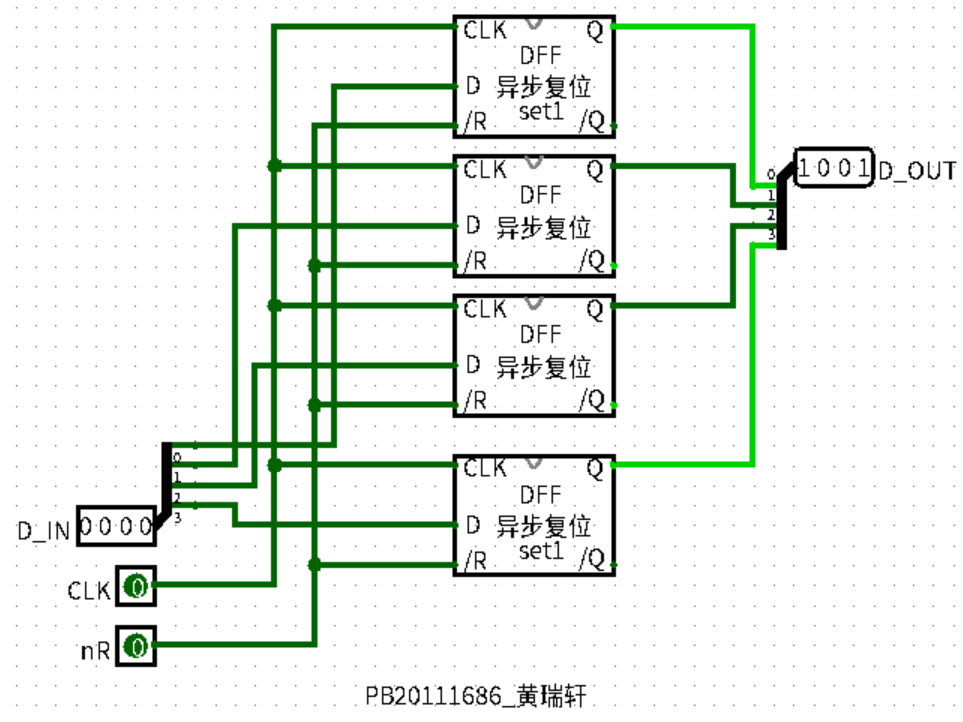
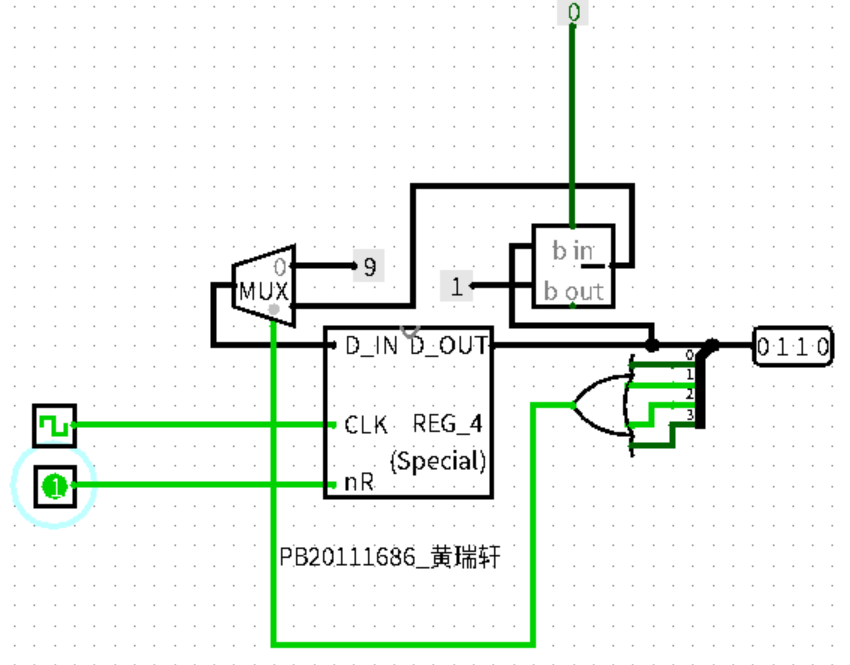
if(rst == 0 || d == 4’b0000) q <= 4’b1001;

else q <= q - 4’b1;

end

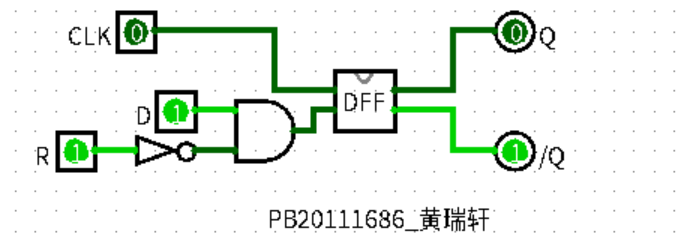
endmodule

如果要求异步复位，就不能通过当前状态来锁住nR，这样会导致计数器卡在1001的状态。可以通过在实现计数器时加一个数据选择器来完成，当前状态为0000时给D\_IN传入0x9，其他时刻传入q - 4’b1。这种情况的寄存器电路如下左图所示，计数器电路如下右图所示。（因为Logisim的数据选择器不能改变方向，导致电路图些许不美观）

**【题目5】**

* 以之前的同步复位D触发器为例，将复位信号后加一个非门即可，电路图如下：



Verilog代码为：

module DFF\_R(input clk, d, r, output reg q);

always@(posedge clk)

begin

if(r == 1) q <= 1’b0;

else q <= d;

end

endmodule

**总结与思考**

* 本次实验中我了解到了在仿真情况下如何搭建时序逻辑电路的方法，并且搭建了锁存器、触发器和寄存器等各种常用的元件。我对时序逻辑电路的理解有了更深刻的认识。
* 本次实验的难度比之前稍有增加，主要在于时序逻辑电路本身比组合逻辑电路要更复杂，实现起来的思维容量比较大，但是经过思考我仍完成了所有练习题。
* 本次实验的任务量较大，主要在实验过程部分，但是这些都是时序逻辑电路的基础。关于任务量的改进建议写在后面。
* 改进建议：

1. 理论课的进度还未讲到寄存器，建议将实验部分的进度和理论部分进行统一。
2. 题目5适合作为思考题，让学生简要回答一下即可，不需要给出示例和代码。