

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：简单时序电路

学生姓名：谭骏飞

学生学号：PB20061276

完成时间：2021.11.4

实验题目 简单时序电路

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；进一步加深对数字电路中时序电路的理解，巩固触发器、寄存器、计数器等相关知识。

实验环境

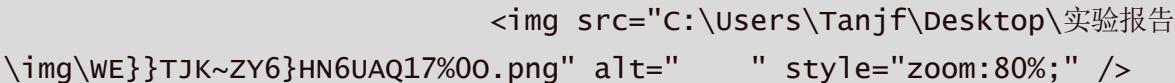
#####	能流畅连接校园网的 PC 一台
vlab.ustc.edu.cn	
#####	Logisim 仿真工具
#####	vscode + 相关插件

实验过程

Step1: 搭建双稳态电路

双稳态电路是由两个非门交叉耦合构成，如下图所示，完全一样的电路结构，却可以具备两种完全不同的状态，这一点与组合逻辑电路存在本质的区别。双稳态电路是一种最简单的时序逻辑电路，没有输入信号，状态一旦确定之后也无法改变，没有实际使用价值，但却是所有时序逻辑电路的基础。

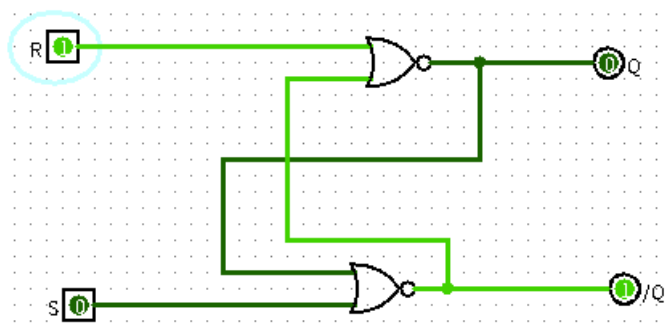
实验过程中搭建的双稳态电路如下。



Step2: 搭建 SR 锁存器

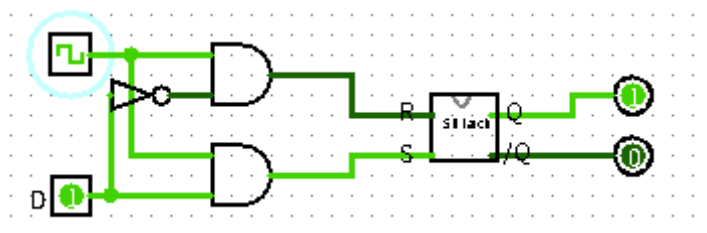
双稳态电路没有输入信号，所以无法进行操作，我们对其进行修改，将两个非门用或非门代替。两个输入信号分别命名为 **S** 和 **R**，输出信号命名为 **Q** 和 \overline{Q} ，其中 \overline{Q} 是 **Q** 取反的意思，**S** 信号负责对 **Q** 置位（**Set**），**R** 信号负责对 **Q** 信号置位（**Reset**）。当 **SR** 信号都无效（为 0）时，电路将保持之前的状态，即处于锁存状态，因此这种电路称为 **SR 锁存器**。**SR** 信号都有效（为 1）时，**Q** 和 \overline{Q} 信号都为零，虽然也是一种确定状态，但不符合 \overline{Q} 为 **Q** 取反的定义，因此我们将其看成是一种未定义态，在实际使用过程中应避免这种状态的出现。

下图为实验过程中搭建的 **SR 锁存器**。



Step3: 搭建 D 锁存器

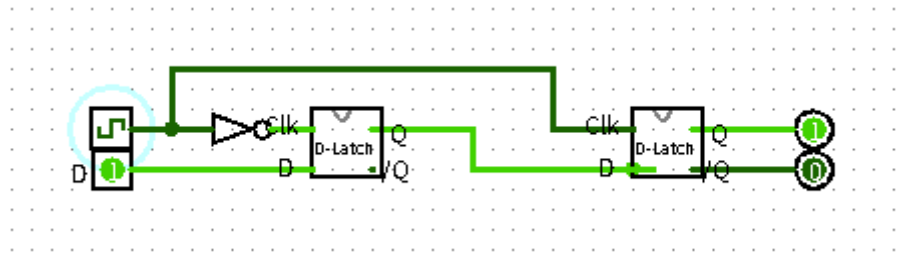
SR 锁存器两个输入都为 1 是一种未定义状态，我们不希望这种状态出现，为此我们在 **SR** 锁存器前面添加两个与门和一个非门便构成了 **D** 锁存器，实验过程中搭建的 **D** 锁存器如下。



分析 **D** 锁存器电路可以发现，当 **CLK** 信号为高电平时，**Q** 信号将随着 **D** 端输入信号的变化而变化，称之为“跟随”状态。当 **CLK** 信号为低电平时，**Q** 信号将保持之前的值，不会收到 **D** 信号变化的影响，称之为“锁存”状态。**D** 锁存器是一种电平敏感的时序逻辑器件。

Step4: 搭建 D 触发器

将两个 D 锁存器串起来，并将其控制信号有效值始终相反，就构成了 D 触发器，CLK 信号为低电平时，D 信号通过了 D1，当 CLK 信号由低电平变为高电平时，D1 关闭，D2 打开，信号到达 Q 端。下图为实验过程中搭建的 D 触发器以及对应的 Verilog 代码。



```
module d_ff (
    input clk,d,
    output reg q
);
always @(posedge clk)
    q<=d;
endmodule //数电实验3-0
```

我们还可以为触发器添加复位信号，当复位信号有效（低电平有效）时，输出信号 Q 始终为零。实验过程中编写的相关代码如下。

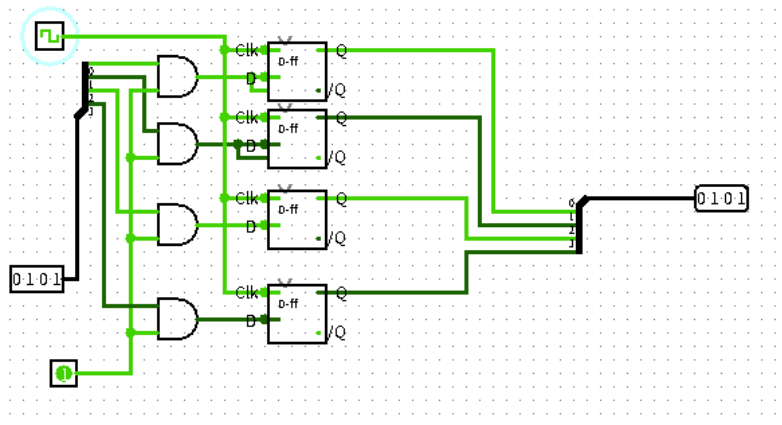
```
module d_ff_r( //同步复位触发器
    input clk,d,rst,
    output reg q
);
always @(posedge clk) begin
    if(rst == 0)
        q<=1'b0; //一种数据表示方式，一般表示为“数据位宽'进制数值”
    else
        q<=d;
end
endmodule //数电实验3-0
```

与此同步复位相对应的，还有一种异步复位方式，即不论时钟和 **D** 信号如何，一旦复位信号有效，输出端 **Q** 立即变为确定的复位值(一般为低电平)。其对应的**Verilog**代码如下。

```
module d_ff_r2(//异步复位触发器
    input clk,d,rst,
    output reg q
);
always @(posedge clk or negedge rst) begin
    if(rst==0)
        q<=1'b0;
    else
        q<=d;
end
endmodule //数电实验3-0
```

Step5: 搭建寄存器

寄存器本质上来说就是 **D** 触发器，如用**4** 个 **D** 触发器构成了一个能够存储 **4bit** 数据的寄存器，带有低电平有效的同步复位信号。其逻辑图**Verilog**代码如下。



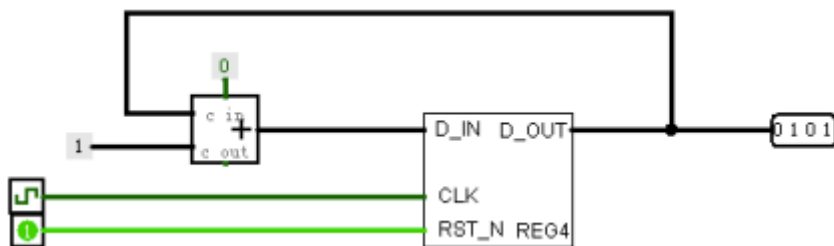
```

module REG4(
    input CLK,RST_N,
    input [3:0] D_IN,
    output reg [3:0] q
);
always @(posedge CLK) begin//同步复位
    if(RST_N==0)
        q<=4'b0;
    else
        q<=D_IN;
end
endmodule //数电实验3-0

```

Step6: 搭建简单时序逻辑电路

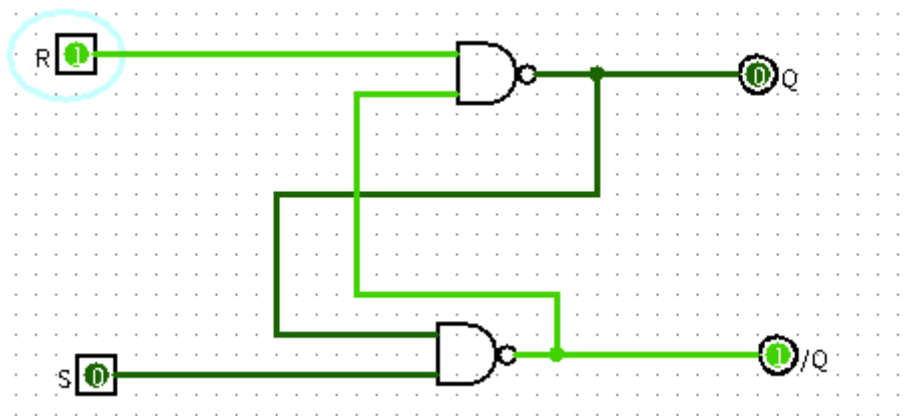
我们利用 **4bit** 寄存器，搭建一个 **4bit** 的计数器，该计数器在 **0~15** 之间循环计数，复位时输出值为 **0**，电路图如下所示。



实验练习

题目 1.在 **Logisim** 中用与非门搭建 **SR** 锁存器，画出电路图，并分析 其行为特性，列出电路在不同输入时的状态。

实验过程中搭建的与非门**SR**锁存器电路图如下。



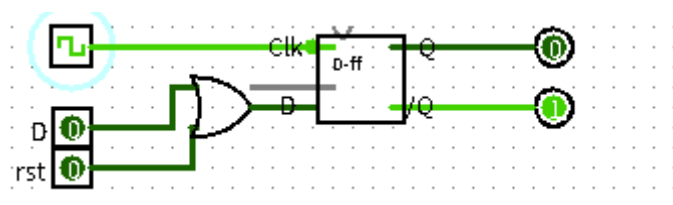
其状态表如下。

R	S	Q	/Q
1	1	保持	保持
0	1	0	1
1	0	1	0
0	0	无效	无效

由此可见，该SR锁存器中的S（Set）和R（Reset）信号都是高电平有效，且在S R同时为1时保持之前状态，不允许出现RS同时为0的情况。

题目 2. 在 Logisim 中搭建一个支持同步置位功能的 D 触发器，画出其电路图，并编写对应的 Verilog 代码。

实验过程中搭建的同步置位D触发器及其相应代码如下。



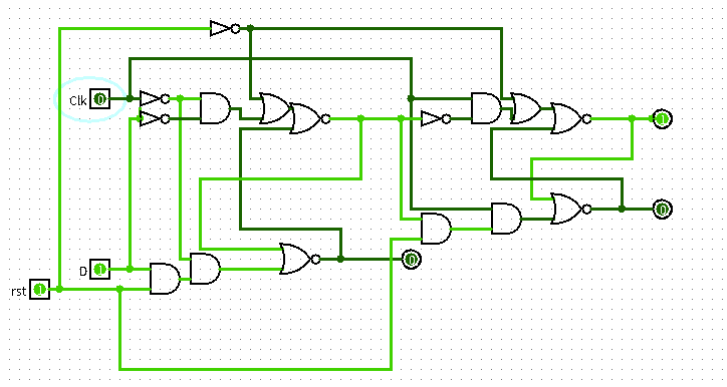
```

module d_ff_r(
    input clk,rst,d,
    output reg q
);
always @(posedge clk) begin
    if(rst==1'b1)//置位信号高电平有效
        q<=1'b1;
    else
        q<=d;
end
endmodule //数电实验3-2

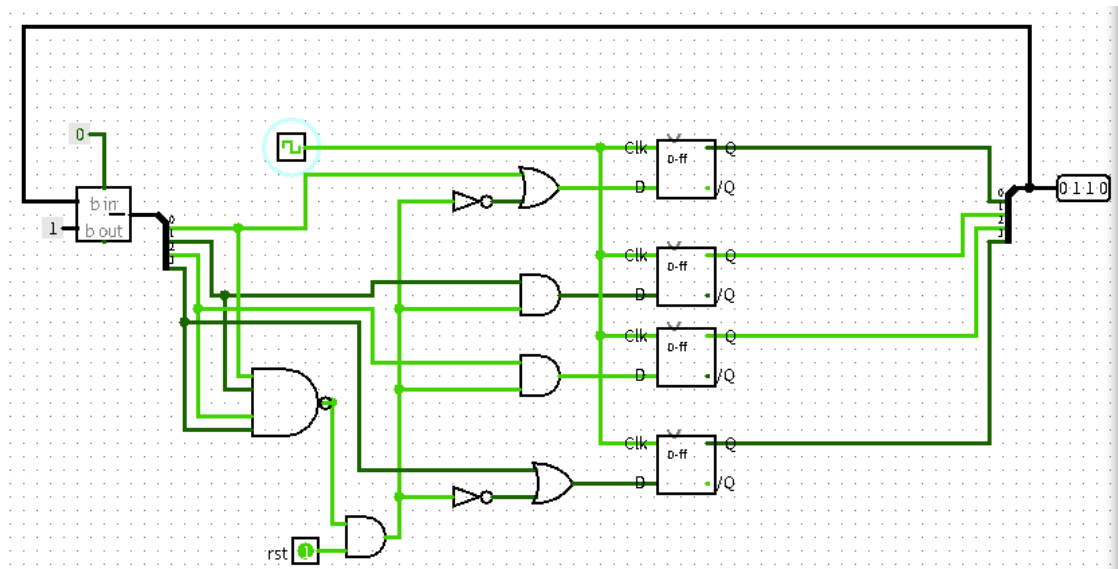
```

题目 3. 在 **Logisim** 中搭建一个带有异步复位功能的 **D** 触发器，画出 其完整电路图，并进一步调用该触发器设计一个从 **0~15** 循环计数的 **4bit** 计数器（可使用 **Logisim** 中的加法器模块，也可自行设计计数器），写出计数器的 **Verilog** 代码。

通过将**rst**信号和**D**信号做与运算的结果直接连到**D**触发器中的两个锁存器，搭出如下电路。



通过异步复位的触发器、加法器及其他相关器件，画出如下的循环计数器。



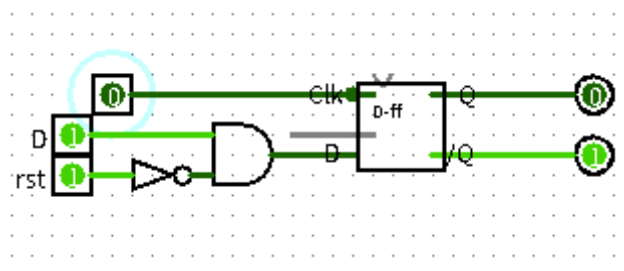
下面附上**Verilog**代码。

```
module down_counter(//递减计数器
    input clk,rst,
    output reg [3:0]d_out
);

always @(posedge clk or negedge rst) begin//异步复位
    if(rst==0)
        d_out<=4'd9;
    else if(d_out==4'b0000)//减到0的时候就恢复到9
        d_out<=4'd9;
    else
        d_out<=d_in;
end
endmodule //数电实验3-4
```

题目 5.前面所有电路的复位信号都是低电平有效，如要使复位信号 高电平有效，应如何实现？试用 **Logisim** 画出一个示例电路，并编写 **Verilog** 代码。

利用与门达到类似效果。示例电路和**Verilog**代码如下。



```

module d_ff_r(//高电平有效的异步复位
    input d,clk,rst,
    output reg q
);
    always @(posedge clk) begin
        if(rst==1)
            q<=1'b0;
        else
            q<=d;
        end
    endmodule //数电实验3-5

```

总结与思考

此次实验工作量和难度较大，但同时收获也很大。实验三不但加深了我对锁存器、触发器、递增递减计数器等时序逻辑电路常见模块的理解，提升了我对时序逻辑电路中同/异步置/复位的认识，也加强了我对verilog语言的掌握，让我得以在应用中结合电路图更直观地理解Verilog代码的运行逻辑，令人受益匪浅。

#####