

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： 李远航

学生学号： PB20000137

完成日期： 2021. 11. 03

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
- 区分组合逻辑电路与时序逻辑电路
- 能够理解同步置位、异步置位、同步复位、异步复位之间的区别
- 能够使用 Verilog HDL 设计简单逻辑电路
- 进一步了解 Verilog 相关语法

## 【实验环境】

- PC 一台
- Windows 10 操作系统
- Java 8 (64 bit)
- Logisim 仿真工具

## 【实验过程】

1. 搭建双稳态电路，搭建如下图所示的电路：

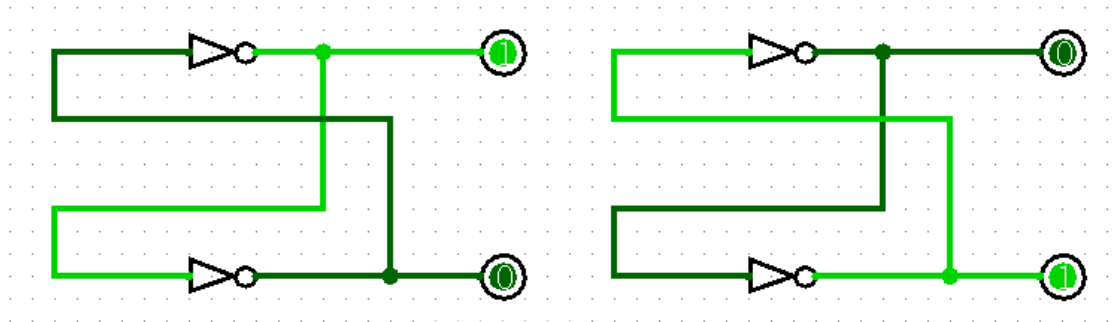


图 1.1

2. 搭建 SR 锁存器，观察电路特性

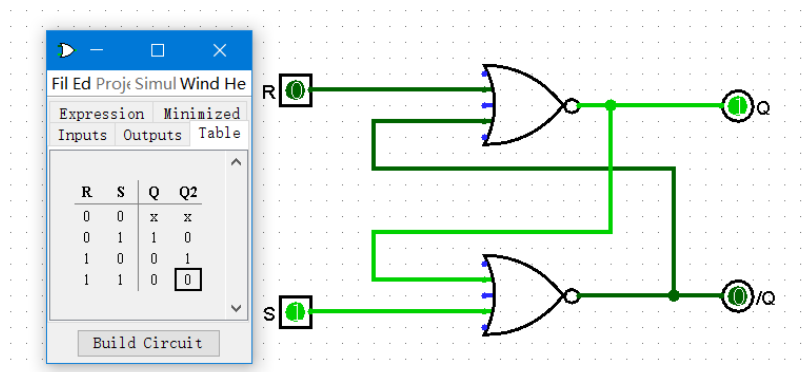


图 2.1

3. 搭建 D 锁存器

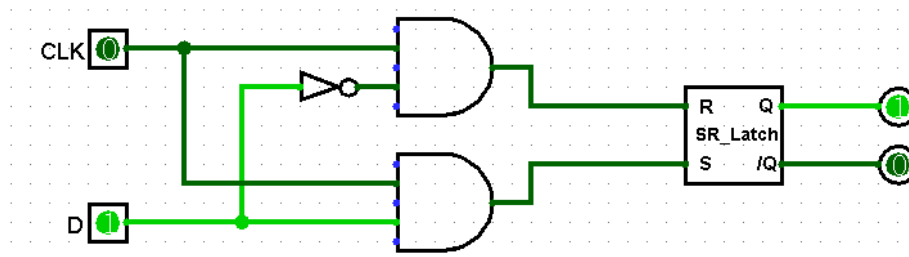


图 3.1

观察发现当 CLK 为高电平时, Q 信号会随 D 的变化而变化, CLK 为低电平时, Q 保持之前的值不变

4. 搭建 D 触发器

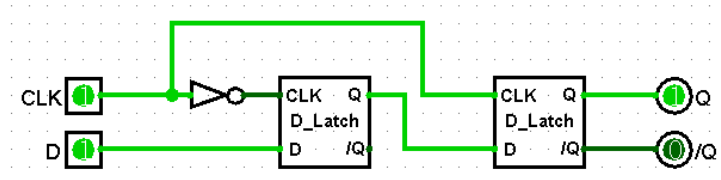


图 4.1

观察电路特性发现，在每一个 CLK 的上升沿，Q 信号会变为此时 D 的值，其他时刻，Q 的值保持不变，对应的 Verilog 代码如下：

```
module d_ff(input clk, d,
            output reg q);
    always@(posedge clk)
        q <= d;
endmodule
```

搭建同步复位的 D 触发器及其 Verilog 代码

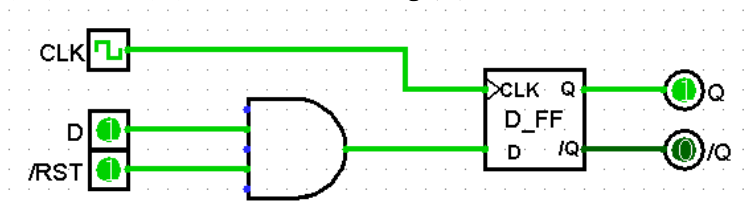


图 4.2

```
module d_ff_r (input clk, rst_n, d,
               output reg q);
    always @(posedge clk)
    begin
        if (rst_n == 0)
            q <= 1'b0;
        else
            q <= d;
        end
    end
endmodule
```

异步复位的 D 触发器 Verilog 代码：

```
module d_ff_r(input clk, rst_n, d,
               output reg q );
    always@(posedge clk or negedge rst_n)
    begin
        if (rst_n == 0)
            q <= 1'b0;
        else
            q <= d;
        end
    end
endmodule
```

### 5. 搭建寄存器

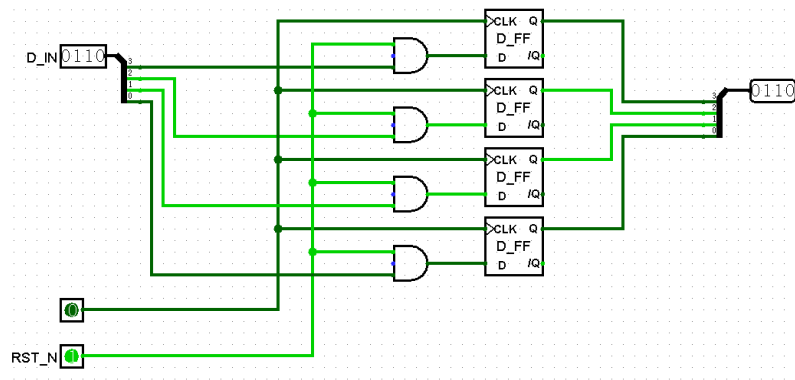


图 5.1

该寄存器能够存储 4bit 的数据，在时钟上升沿，复位信号为高电平时，将输入值存储到输出端，其 Verilog 代码如下：

```
module REG4 (input CLK, RST_N,
             input [3:0] D_IN,
             output reg [3:0] D_OUT);
    always @(posedge CLK)
    begin
        if (RST_N == 0)
            D_OUT <= 4'b0;
        else
            D_OUT <= D_IN;
    end
endmodule
```

### 6. 搭建简单时序逻辑电路

在 Logism 中利用寄存器搭建如下所示的循环计数器：

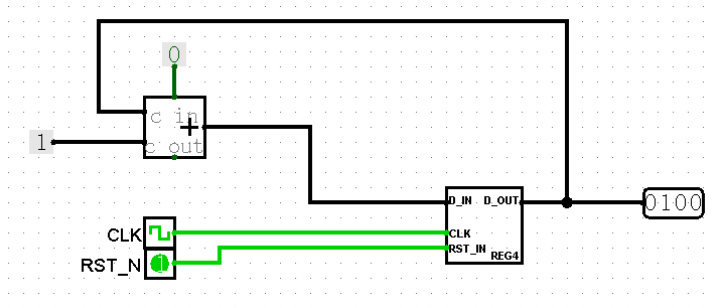


图 6.1

当复位信号高电平，随着 CLK 的变化，输出端从 0~15 再变回 0  
Verilog 代码如下：

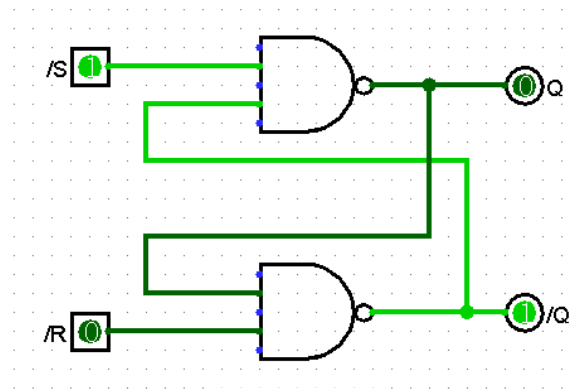
```

module REG4 (input CLK, RST_N,
              output reg [3:0] CNT);
always @(posedge CLK) begin
    if (RST_N == 0)
        CNT <= 4'b0;
    else
        CNT <= CNT+4'b1;
    end
endmodule

```

## 【实验练习】

### 1. 使用与非门搭建 SR 锁存器

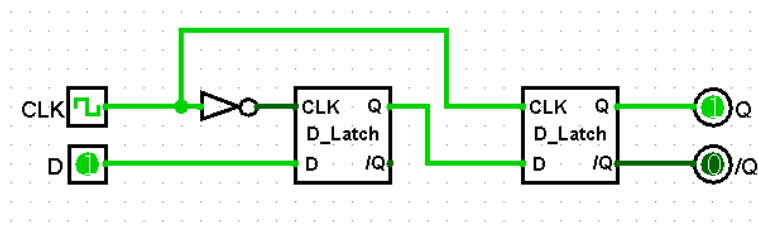


题图 1.1

S	R	Q	$\sim Q$
0	0	1	1
0	1	1	0
1	0	0	1
1	1	X	X

表 1: 电路特性

### 2. 使用 D 锁存器搭建出如图所示的 D 触发器，以及对应的 Verilog 代码



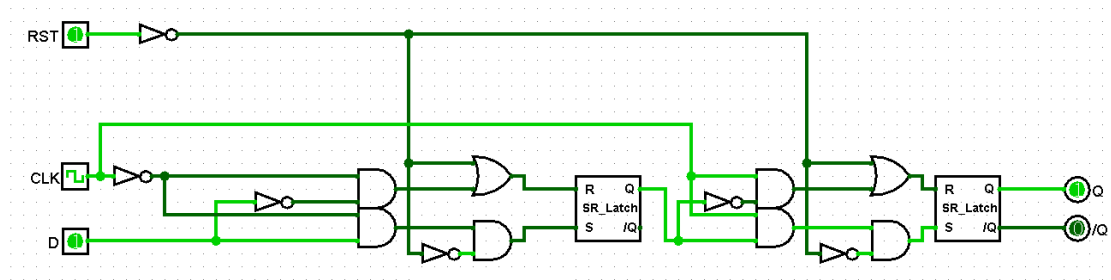
题图 2.1

```

module D_FF (input CLK, D,
              output reg Q);
    always @(posedge CLK)
    begin
        Q <= D;
    end
endmodule

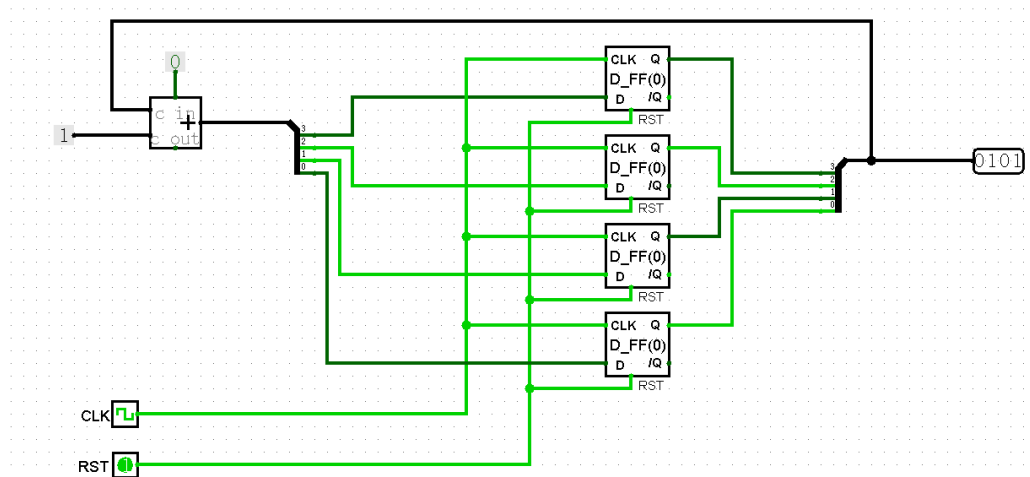
```

3. 异步复位的 D 触发器如图所示，使用 SR 锁存器，RST 从 1 变成 0 时，Q 置 0



题图 3.1

0~15 循环计数的 4bit 计数器如图所示，RST 从 1 变 0 时，输出端置 0:



题图 3.2

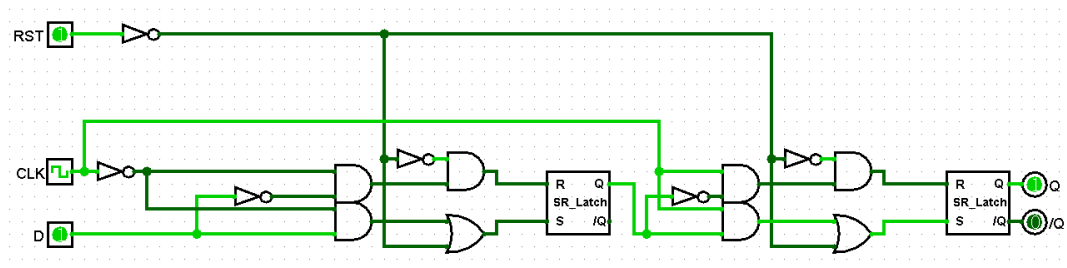
对应循环计数器的 Verilog 代码:

```

module fun (input CLK, RST,
            output reg [3:0] OUT);
    always @(posedge CLK or negedge RST)
    begin
        if (RST == 0)
            OUT <= 4'b0;
        else
            OUT <= OUT+4'b1;
        end
    end
endmodule

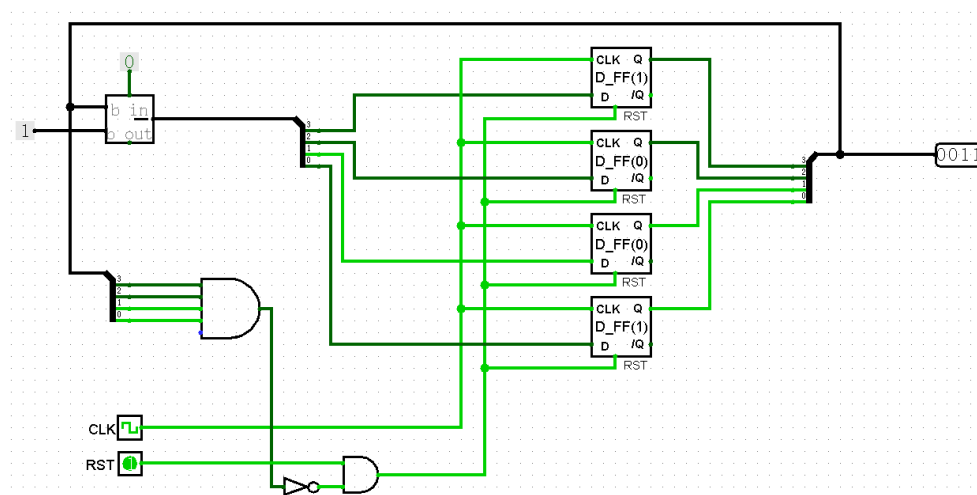
```

4. 循环递减的计数器需要用到异步复位为 1 的 D 触发器，如图所示：



题图 4.1

如图所示对于不同位选用复位值不同的触发器，以此达到复位 9 的目的，同时在检测到输出值变成 4'b1111 时同时触发复位信号，可以得到符合题意的电路，虽然输出端经历短暂的 4'b1111 的状态，但是属于复位为 9 的时钟脉冲：



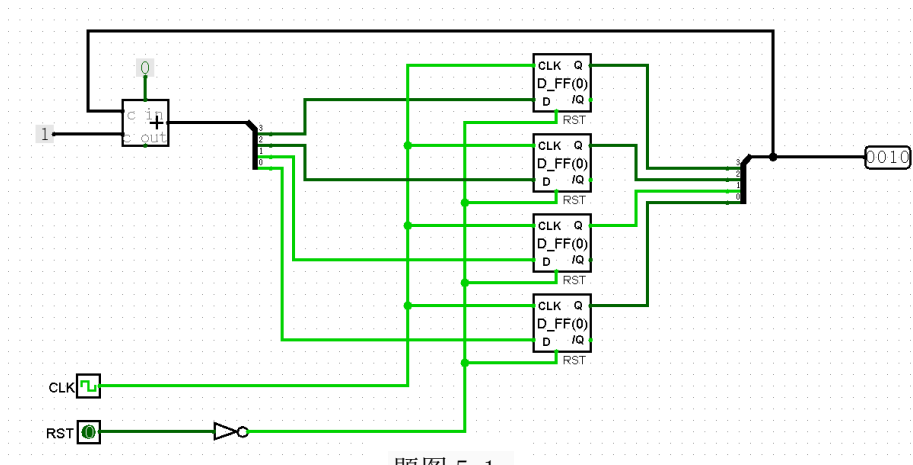
题图 4.2

Verilog 代码如下:

```
module fun (input CLK, RST,
            output reg [3:0] CNT);
    always @(posedge CLK or negedge RST)
    begin
        if (RST == 0 || CNT == 4'b0)
            CNT <= 4'b1001;
        else
            CNT <= CNT-4'b1;
        end
    endmodule
```



5. 在复位端加上一个非门就可以使得复位信息高电平有效，选用题目 3 的电路和 Verilog 代码进行修改：



题图 5.1

```
module fun (input CLK, RST,
            output reg [3:0] OUT);
    always @(posedge CLK or posedge RST)
    begin
        if (RST == 1)
            OUT <= 4'b0;
        else
            OUT <= OUT+4'b1;
        end
    end
endmodule
```

### 【总结与思考】

- 学习了时序逻辑相关器件的原理及底层结构
- 学会使用基本逻辑门搭建各类时序逻辑器件
- 学会使用 Verilog HDL 设计简单逻辑电路
- 更加深刻认识到组合逻辑电路与时序逻辑电路的区别
- 本次实验较之前几次实验任务量稍大，难度有提高