实验 3　状态机实验

班级：01　　学号：2021k8009925006　　姓名：冯浩瀚

实验目的:

1、熟悉 Verilog 编程、调试

2、熟悉状态机的工作原理，能熟练编写状态机程序

实验环境:

1、操作系统：Windows 11

2、软件及版本：Vivado 2017.4

原理说明

1. 状态机：

有限状态机（Finite-State Machine，FSM），又成为有限状态自动机，简称状态机，是表示有限个状态以及在这些状态之间的转移和动作等行为的数学模型。

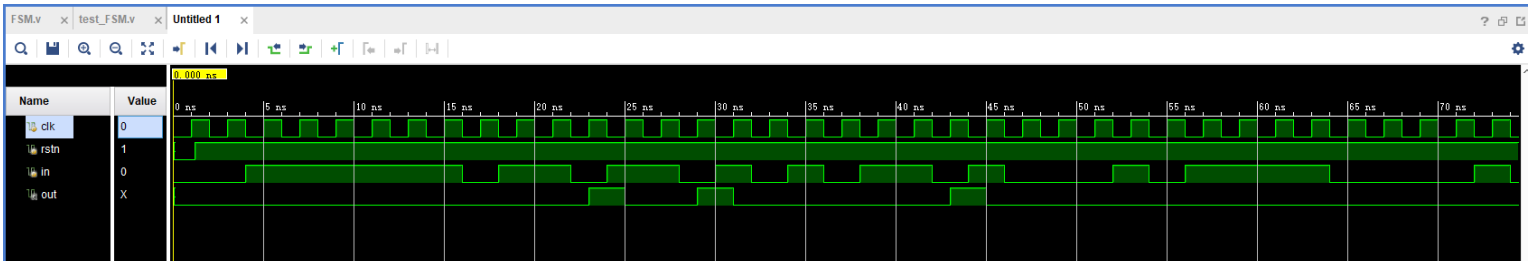
- 有限状态机的组成
  a) 状态寄存器：记忆状态机当前所处的状态
  b) 产生下一状态的组合逻辑：根据输入信号或当前状态，确定下一状态
  c) 输出逻辑：由当前状态和输入信号，决定当前状态的输出
- 两类状态机
  a) Moore 状态机：状态机的输出仅仅依赖于当前状态，而与输入条件无关
  b) Mealy 状态机：状态机的输出不仅依赖于当前状态，而且取决于该状态的输入条件

接口定义:

Input 信号			Output 信号		
名称	说明	位宽	名称	说明	位宽
clk	时钟信号	1	out	检测结果	1
rstn	同步置位信号	1			
in	输入信号	1			

调试过程及结果波形:

结果波形：



实验总结:

本实验中练习编写了有限状态机的编写，选择了推荐的三段式写法。自学了课前资料后编写该项目并未遇到困难。

源代码:

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2022/11/10 10:30:09
// Design Name:
// Module Name: FSM
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
```

```
module FSM(
    input clk,
    input rstn,
    input in,
    output reg out
    );

    localparam S0 = 3'b000;
    localparam S1 = 3'b001;
    localparam S2 = 3'b010;
    localparam S3 = 3'b011;
    localparam S4 = 3'b100;

    reg [2:0] state;
    reg [2:0] next_state;

    always@(negedge rstn or posedge clk)begin
        if(!rstn) begin
            state <= S0;
        end
        else begin
            state <= next_state;
        end
    end

    always@(*) begin
        case(state)
        S0:begin
            next_state = in ? S0 : S1;
        end
        S1:begin
            next_state = in ? S2 : S1;
        end
        S2:begin
            next_state = in ? S3 : S1;
        end
        S3:begin
            next_state = in ? S0 : S4;
        end
        S4:begin
            next_state = in ? S2 : S1;
        end
        default: next_state = S0;
        endcase
```

```
    end

    always@(*) begin
        case(state)
        S4:out = 1'b1;
        S0:out = 1'b0;
        S1:out = 1'b0;
        S2:out = 1'b0;
        S3:out = 1'b0;
        default: out = 1'bx;
        endcase
    end
endmodule
```

testbench:

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2022/11/10 10:58:36
// Design Name:
// Module Name: test_FSM
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module test_FSM(

    );
    reg clk, rstn, in;
    wire out;
    FSM instance_FSM(
        .clk(clk),
        .rstn(rstn),
```

```
        .in(in),
        .out(out)
    );

    initial begin
        clk = 0;
        rstn = 1;
        #0.1 rstn = 0;
        #1.1 rstn = 1;
    end

    initial begin
        in = 0;
        #4 in = 1;
        #4 in = 0;
    end

    always begin
        #1 clk = ~clk;
    end

    always begin
        #2 in = $random() % 2;
    end
endmodule
```