



中国科学院大学
University of Chinese Academy of Sciences

数 字 电 路

实验报告

班级：教 221

组号：----

姓名：唐嘉良

学号：2020K8009907032

实验名称：状态机实验

2021 年 11 月 18 日

一、实验目的

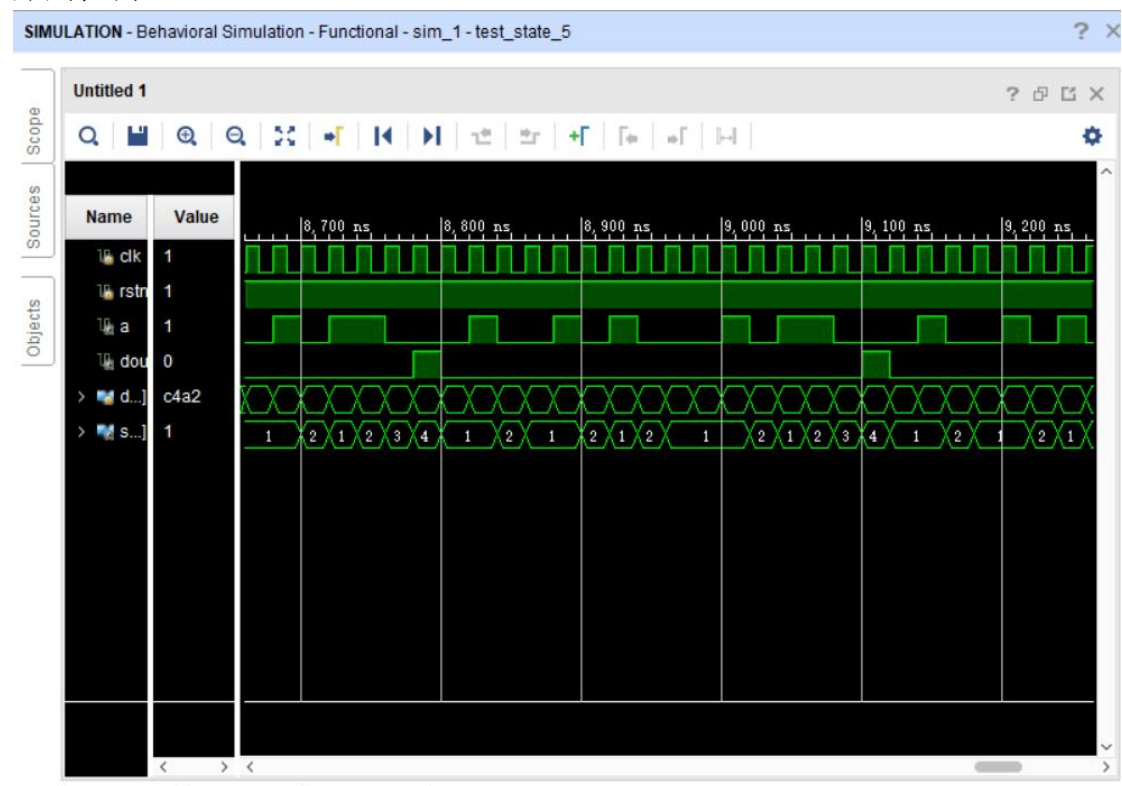
- 1、熟悉 verilog 编程、调试
- 2、熟悉状态机的工作原理，能熟练编写状态机程序

二、实验环境

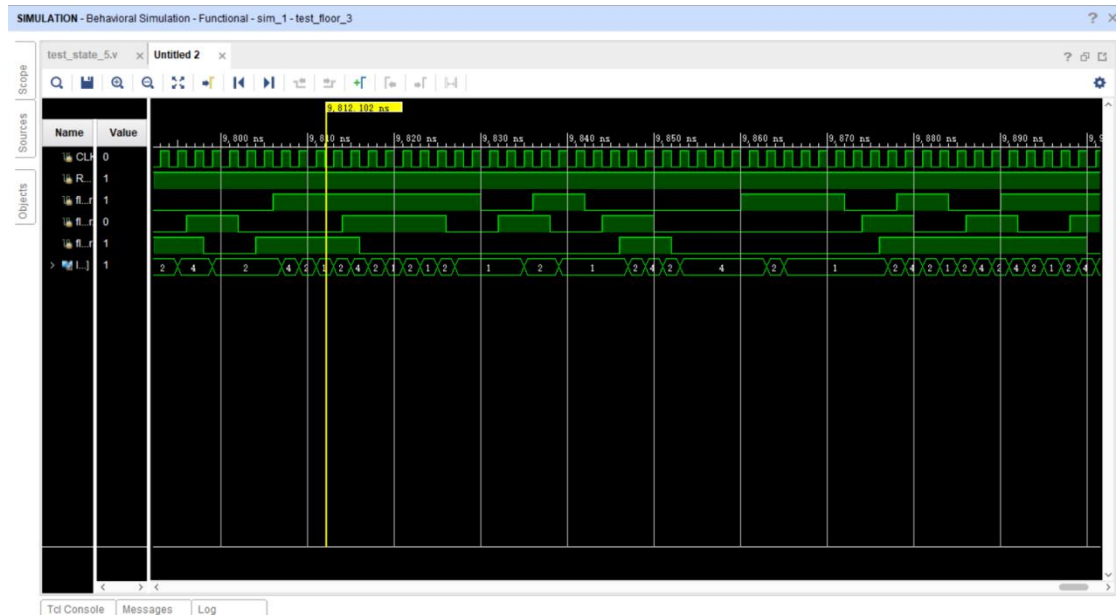
本次实验我采用的是 vivado 2017.4 版本。

三、调试过程

序列检测



电梯



四、实验总结

在此次实验中，我更加熟悉 vivado 平台的操作流程，现在能够创建激励文件并进行调试。同时，通过构建序列检测器和电梯控制器，我对 Verilog 语言的掌握程度大大提升，能够更加熟练自如地构建并调试某些常见组合电路模块。最重要的是，这两类题目均偏向应用，我的应用能力和全面思考能力得到了提升。

五、源代码

1.序列检测

测试代码

```
module state_5(  
    input clk,  
    input rstn,  
    input a,  
    output reg dout,  
    output reg [2:0] state  
);  
  
reg[2:0] next_state;  
parameter S0='d0, S1='d1, S2='d2, S3='d3, S4='d4;  
  
always @(negedge rstn or posedge clk)  
begin  
    if(!rstn)  
        begin  
            state <= S0;
```

```
end
else
begin
state <= next_state;
end
end
```

```
always @(state or a)
begin
case(state)
S0:begin
if(a==1)
begin
next_state=S0;
end
else
begin
next_state=S1;
end
end
S1:begin
if(a==0)
begin
next_state=S1;
end
else
begin
next_state=S2;
end
end
S2:begin
if(a==1)
begin
next_state=S3;
end
else
begin
next_state=S1;
end
end
S3:begin
if(a==1)
begin
next_state=S0;
```

```

    end
    else
    begin
    next_state=S4;
    end
    end
    S4:begin
    if(a==1)
    begin
    next_state=S2;
    end
    else
    begin
    next_state=S1;
    end
    end
    default:next_state=S0;
    endcase
end

```

```

    always @(state)
    begin
    case(state)
    S4:dout<=1;
    default:dout<=0;
    endcase
    end

```

```

endmodule

```

激励文件

```

module test_state_5(

);
    reg clk;
    reg rstn;
    wire a;
    wire dout;
    reg [15:0] data;
    wire [2:0] state;
    state_5 obj_state_machine(
        .clk(clk),
        .rstn(rstn),
        .a(a),

```

```

        .dout(dout),
        .state(state)
    );
    assign a=data[15];
    always #10clk=~clk;
    always @(posedge clk)
    begin
        data={data[14:0],data[15]};
    end
    initial begin
        clk=0;
        rstn=1;
        #2 clk=1;
        #30 clk=0;
        data='b0110_0101_1000_1001_0100;
    end
end
endmodule

```

2. 电梯

测试代码

```

module test_state_5(

    );
    reg clk;
    reg rstn;
    wire a;
    wire dout;
    reg [15:0] data;
    wire [2:0] state;
    state_5 obj_state_machine(
        .clk(clk),
        .rstn(rstn),
        .a(a),
        .dout(dout),
        .state(state)
    );
    assign a=data[15];
    always #10clk=~clk;
    always @(posedge clk)
    begin
        data={data[14:0],data[15]};
    end
    initial begin
        clk=0;

```

```

    rstn=1;
    #2 clk=1;
    #30 clk=0;
    data='b0110_0101_1000_1001_0100;
end
endmodule

```

激励文件

```

module test_floor_3(

);
    reg CLK;//时钟信号，每过一个时钟周期转化一次楼层
    reg RESET;//清零信号
    reg floor1;//按下 1 楼输入信号
    reg floor2;//按下 2 楼输入信号
    reg floor3;//按下 3 楼输入信号
    wire [2:0] led;//表示当前所对应楼层,[2:0]

    floor_3 u0(CLK, RESET ,floor1, floor2, floor3, led);

    initial begin
        CLK = 0;
        RESET = 1;
        #0.1 RESET = 0;
        #1.1 RESET = 1;
    end
    initial begin
        floor1 = 0;
        floor2 = 0;
        floor3 = 0;

    end
    always begin
        #1 CLK = ~CLK;
    end
    always begin
        #2 floor1 = $random() % 2;
        #2 floor2 = $random() % 2;
        #2 floor3 = $random() % 2;
    end

endmodule

```