实验报告

实验题目：Lab10　　日期：2018年11月30日

姓名:刘紫檀　　学号:PB17000232　　成绩:____________

实验目的：

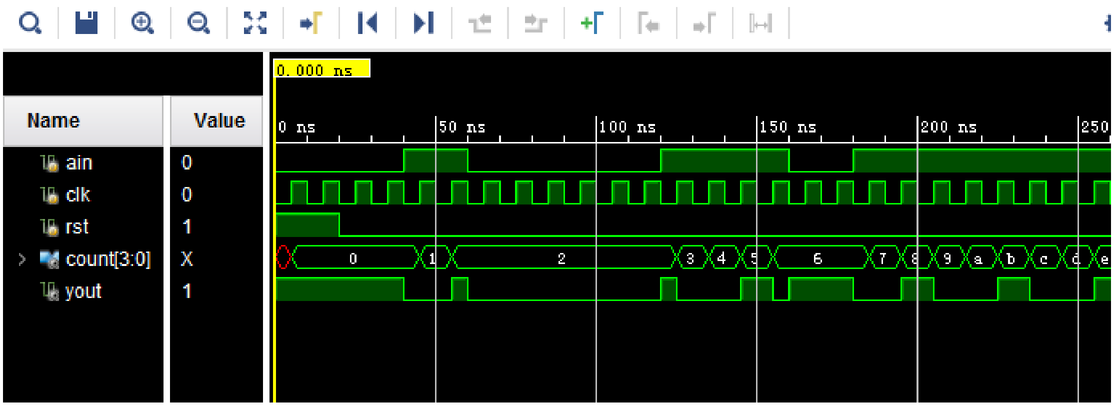
- 1-1 使用三段式Mealy型FSM实现一个序列检测器，当接收到的1的总数为3的整数倍时，输出为1，否则输出0。编写代码2分、仿真2分、下载2分。
- 2-1 使用三段式 Moore 型 FSM 实现一个序列检测器，序列为01/00时输出为0，序列为11/00时输出为1，序列为10/00时输出反转，否则输出保持不变。编码 2 分、仿真 2 分、下载 2 分

截图：

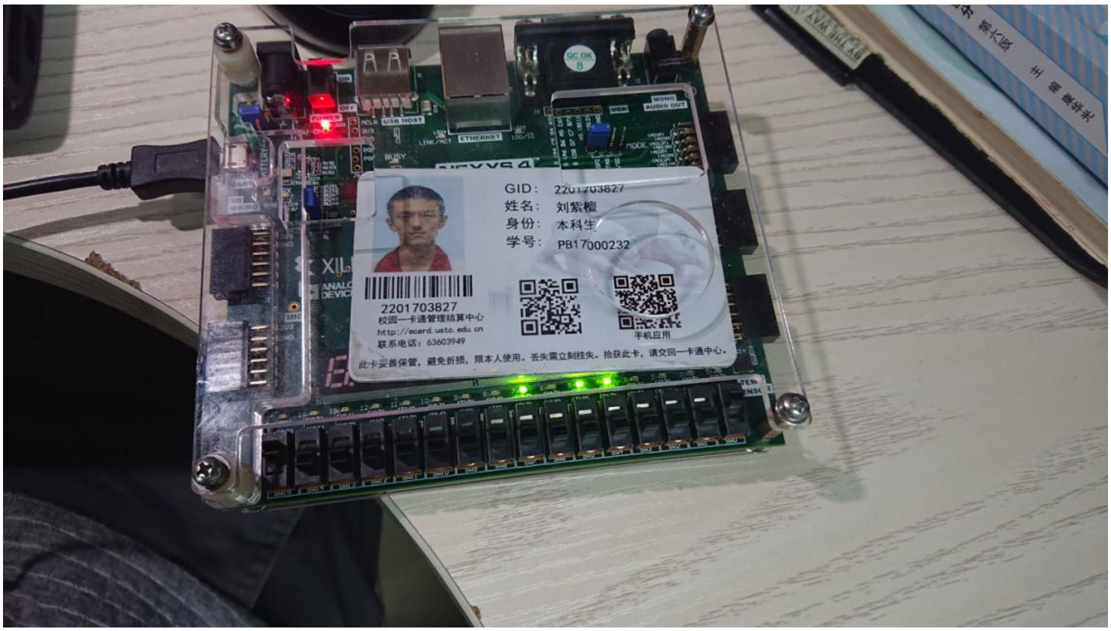
```
//实验 1-1 代码
`timescale 1ns / 1ps

// 0 -> 1 -> 2
//(1)  (0)  (0)
//a=1 represents the [next] state prediction..
//receiving 0 will cause the circ hang at their current state

module seq_detect(
    input ain,
    input clk,
    input rst,
    output reg yout,
    output reg [3:0] count
    );
    reg [1:0] state;
    reg [1:0] next_state;
    reg count_carry;
    always @ (posedge clk or negedge rst) begin
        if (rst) begin
```

```
            state <= 2’b00;
            count <= 4’b0000;
        end else begin
            state <= next_state;
            count <= count + count_carry;
        end
    end

    always @ (state or ain) begin  //next_state
        count_carry = 0;
        case(state)
            2’b00: if (ain == 1) begin next_state = 2’b01; count_carry = 1; end else
next_state = 2’b00;
            2’b01: if (ain == 1) begin next_state = 2’b10; count_carry = 1; end else
next_state = 2’b01;
            2’b10: if (ain == 1) begin next_state = 2’b00; count_carry = 1; end else
next_state = 2’b10;
            //illegel
            2’b11: next_state = 2’b00;
        endcase
    end

    always @ (state or ain) begin  //output
        case(state)
            2’b00: if (ain == 1) yout = 0; else yout = 1;
            2’b01: yout = 0;
            2’b10: if (ain == 1) yout = 1; else yout = 0;
            default: yout = 1; //SHOULD NEVER REACH THIS
        endcase
    end

endmodule

//实验 1-1 Testbench
`timescale 1ns / 1ps

module seq_detect_tb();

    reg ain;
    reg clk;
    reg rst;
    wire [3:0] count;
    wire yout;
    seq_detect DUT(ain, clk, rst, yout, count);
    initial begin clk = 0; end
    always #5 clk=~clk;
    initial begin
        ain = 0;
        rst = 1;
        #20 rst = 0;
        #20 ain = 1;
        #20 ain = 0;
        #60 ain = 1;
        #40 ain = 0;
        #20 ain = 1;

    end

endmodule
```

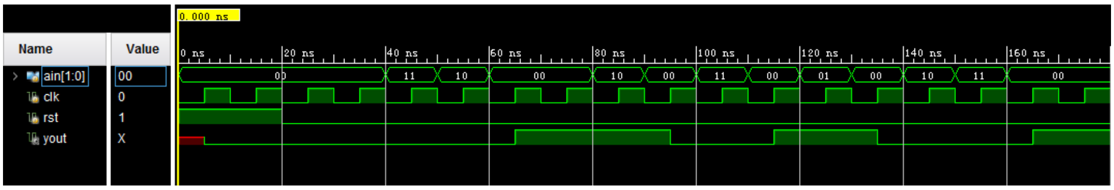
实验 1-1 仿真截图



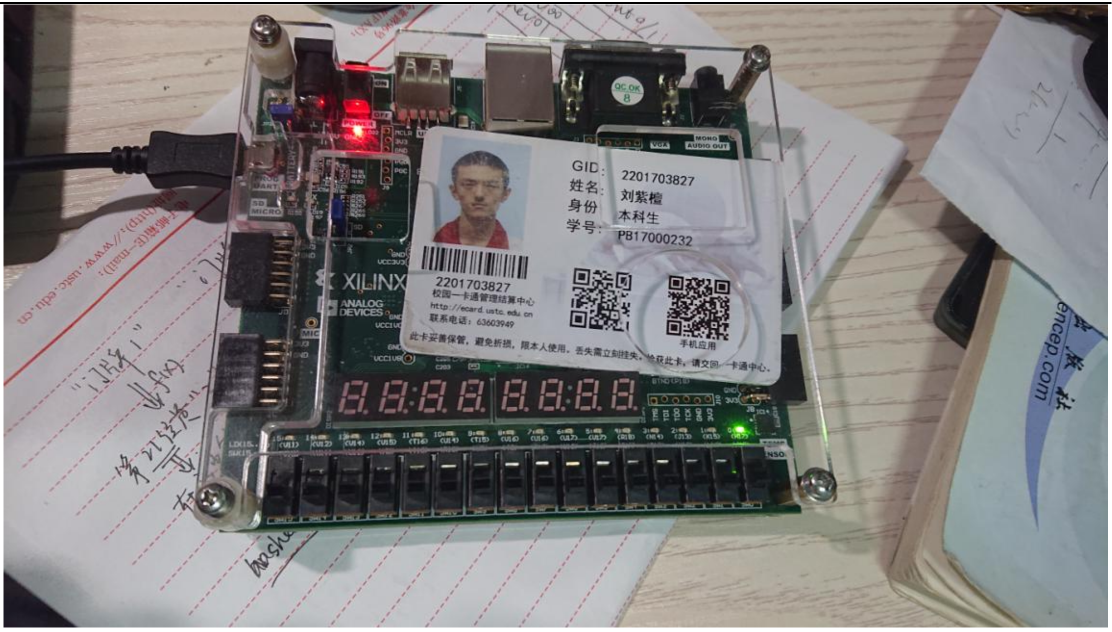
实验 1-1 下载截图



```
//实验2-1代码
`timescale 1ns / 1ps
/* States variable is designed as {out, prev[1], prev[0]} */

module seq_detect_moore(
    input [1:0] ain,
    input clk,
    input rst,
    output reg yout);

    reg out;
    reg [1:0] prev;
    reg next_out;
    reg [1:0] next_prev;
    always @ (posedge clk or negedge rst) begin
        if (rst) begin
           out <= 0;
           prev <= 2’b00;
        end else begin
           out <= next_out;
           prev <= next_prev;
        end
    end

   always @ (out) begin
       yout = out;
   end

   always @ (out, prev, ain) begin
        case ({out, prev[1], prev[0]})
            3’b000:
               next_out = out;
            3’b100:
               next_out = out;
            3’b001:
               case (ain)
                   2’b00: next_out = 0;
                   default: next_out = out;
               endcase
            3’b101:
               case (ain)
                   2’b00: next_out = 0;
                   default: next_out = out;
               endcase
            3’b010:
                case (ain)
                    2’b00: next_out = 1;
                    default: next_out = out;
                endcase
            3’b110:
                case (ain)
                    2’b00: next_out = 0;
                    default: next_out = out;
                endcase
            3’b011:
                case (ain)
                    2’b00: next_out = 1;
                    default: next_out = out;
                endcase
            3’b111:
                case (ain)
                    2’b00: next_out = 1;
                    default: next_out = out;
                endcase
        endcase
        next_prev = ain;
    end

endmodule

//实验2-1 Testbench
`timescale 1ns / 1ps

module seq_detect_tb();

    reg [1:0] ain;
    reg clk;
    reg rst;
    wire yout;
    seq_detect_moore DUT(ain, clk, rst, yout);
    initial begin clk = 0; end
    always #5 clk=~clk;
    initial begin
        ain = 2’b00;
        rst = 1;
        #20 rst = 0;
        #20 ain = 2’b11;
        #10 ain = 2’b10;
        #10 ain = 2’b00;
        #20 ain = 2’b10;
        #10 ain = 2’b00;
        #10 ain = 2’b11;
        #10 ain = 2’b00;
        #10 ain = 2’b01;
        #10 ain = 2’b00;
        #10 ain = 2’b10;
        #10 ain = 2’b11;
        #10 ain = 2’b00;
        #30 ain = 2’b10;

    end

endmodule
```



实验 2-1 仿真截图



实验 2-1 下载截图

实验总结：这次实验我了解了状态机在 Verilog 中的实现方法。