**实验4**

课程名称:数字逻辑

实验教学学时：

学时 ；

年级/班级： 6

分组学生人数： 2

专业：软件工程

姓名：王业超

1. **实验目的：**After this is complete, you will be able to: • models • Moore FSMs Mealy FSMs
2. （1）Design a sequence detector implementing a Mealy state machine using three always blocks. The Mealy state machine has one input (ain) and one output (yout). The output yout is 1 if and only if the total number of 1s received is divisible by 3 (hint: 0 is inclusive, however, reset cycle(s) do not count as 0- see in simulation waveform time=200). Develop a testbench and verify the model through a behavioral simulation. Use SW15 as the clock input, SW0 as the ain input, the BTNU button as reset input to the circuit, number of 1s count on LED7:LED4, and LED0 as the yout output. Go through the design flow, generate the bitstream, and download it into the Basys3 board. Verify the functionality.

**二、实验原理或预习内容**

（1）Operating the inputs logcially according to the digital circuit’s rules.The switch is one(true)when it is on and the LED is light when the output is one(right).We can observe the LEDs to test our circuit.

**三、实验环境**

（1）Basys3 board

（2）Vivado

**四、实验内容**

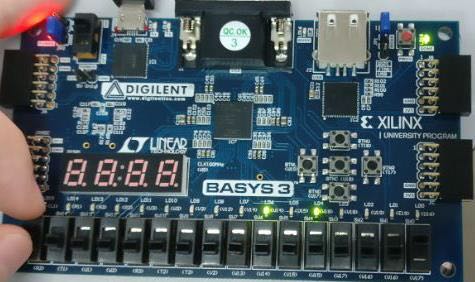
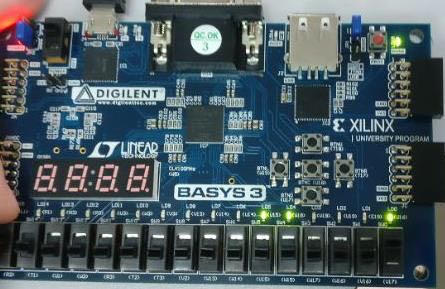
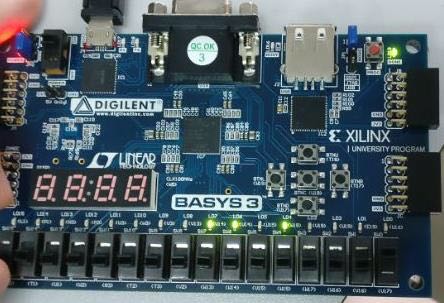
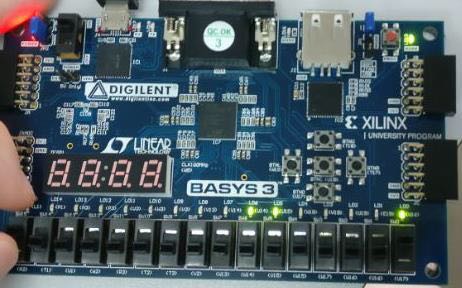
**The Verilog code is listed as follows。**

module mealy\_sequence\_detector\_3processes(

input clk,  
    input reset,  
    input ain,  
    output reg [3:0] count,  
    output reg yout  
    );  
    reg [1:0] state, nextstate;  
    parameter [1:0] S0=0, S1=1, S2=2;  
//state register and counter  
    always @(posedge clk )  
    begin  
if(reset)  
begin  
state<=S0;  
count<=4'b0000;  
end  
else  
  begin  
    state<=nextstate;  
    if(ain)  
      begin  
         if(count==4'b1111)count<=4'b0000;  
         else count<=count+1'b1;  
         end  
  end  
end  
//output logic  
    always @(state or reset)  
    begin  
    yout = ~reset&~state[1]&~state[0];  
end  
//next state logic  
    always @(state or ain)  
    begin  
    nextstate[1]=~state[1]&state[0]&ain | state[1]&~state[0]&~ain;  
  
    nextstate[0]=~state[1]&~state[0]&ain | ~state[1]&state[0]&~ain;  
    end  
endmodule  
module moore\_sequence\_detector\_3processes(  
  
    input clk,  
    input reset,  
    input [1:0] ain,  
    output reg yout  
    );  
    reg [2:0] state, nextstate;  
    parameter S0 = 3'b000;  
    parameter S1 = 3'b001;  
    parameter S2 = 3'b010;  
    parameter S3 = 3'b011;  
    parameter S4 = 3'b100;  
    parameter S5 = 3'b101;  
    parameter S6 = 3'b110;  
    parameter S7 = 3'b111;  
//state register logic  
    always @(posedge clk or posedge reset)  
    begin  
if(reset)state<=S0;  
else state<=nextstate;  
end  
//next state logic  
    always @(state or ain)  
    begin  
    nextstate[2]=~state[2]&state[1]&~ain[1]&~ain[0] | state[2]&~state[1]&~state[0] | state[2]&state[1]&state[0]   
                 | state[2]&~state[0]&ain[0] | state[2]&state[0]&ain[0] | state[2]&ain[1]&~ain[0];  
    nextstate[1]=ain[1]&~ain[0] | ain[1]&ain[0];  
    nextstate[0]=ain[1]&ain[0] | ~ain[1]&ain[0];  
    end  
//output logic

  always @(state)  
    begin  
    yout=state[2];   
    end  
endmodule

The results are showed in the following pictures.



**五、实验结论及思考题**

（1）According to the outputs of Basys3 board,we can draw a conclusion that we have successed in designing a sequence detector implementing a Mealy state machine .

（2）The perfect outputs show that the digital circuit we design can work well and we have successed in getting the right outputs.