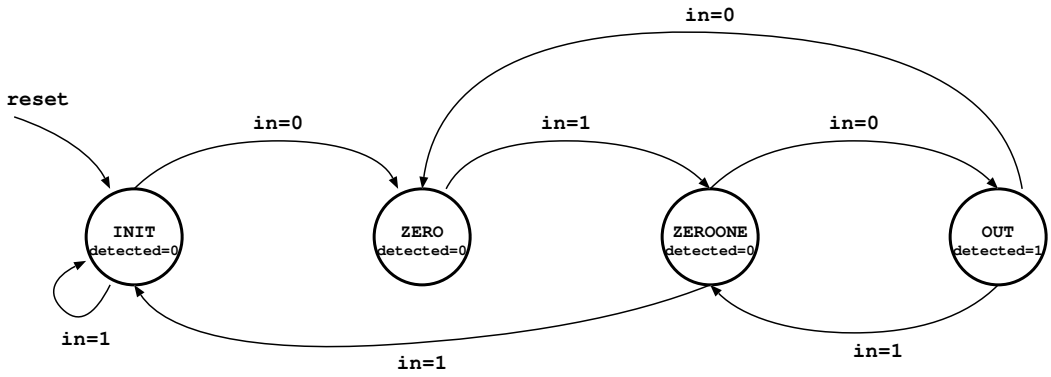
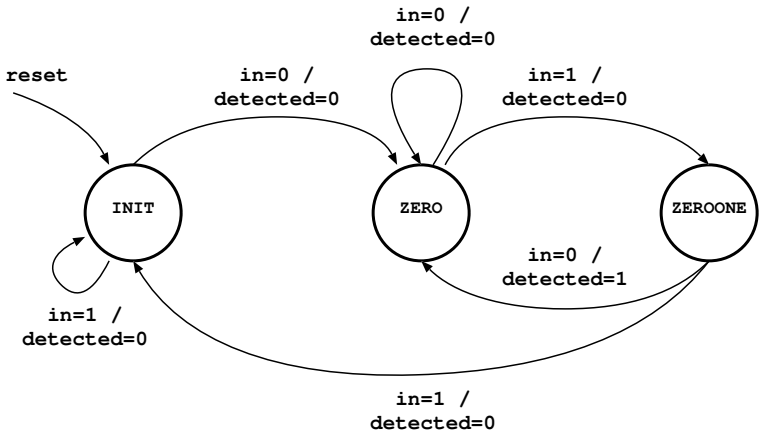
3. We want to design a Finite State Machine (FSM) that has a one bit input (`in`) and will detect the sequence 0-1-0. If this sequence is detected, the one bit output (`detected`) will be set to 1, otherwise this output will remain at 0.

   Two of your colleagues have designed different state transition diagrams given below.

FSM A



FSM B



(a) (1 point) Which one of the state diagrams depicts a Moore and which one a Mealy type of FSM

> **Solution:**
> FSM A is a Moore type FSM, the output depends only on the state. FSM B is a Mealy type FSM, the output depends on both the state and inputs

(b) (4 points) For both state transition diagrams state whether or not they are correct.

> **Solution:**
> FSM A has a small mistake, for state `ZERO` it is not clear what will happen when `in=0`. FSM B is correct.

(c) (7 points) Complete the following Verilog module that would implement the state machine as described in the question. You can implement one state transition diagram of your colleagues if that one is correct.

```
module fsm (input in, input clk, input reset, output reg detected);

reg [1:0] next_state, present_state;

parameter INIT    = 2'b11;
parameter ZERO    = 2'b00;
parameter ZEROONE = 2'b01;

always @ (*)
   begin
     next_state <= present_state;   // default
     detected <= 1'b0;
     case (present_state)
       INIT:     next_state <= in ? INIT   : ZERO;
       ZERO:     next_state <= in ? ZEROONE: ZERO;
       ZEROONE:  if (in)
                   next_state <= INIT;
                 else
                   begin
                     next_state <= ZERO;
                     detected <= 1'b1;
                   end
       default: next_state <= present_state;
     endcase
   end

always @ (posedge clk, posedge reset)
  if (reset)  present_state <= INIT;
  else        present_state <= next_state;

endmodule
```