Exams/ece241 2013 q8

Implement a **Mealy**-type finite state machine that recognizes the sequence "101" on an input signal named **x**. Your FSM should have an output signal, **z**, that is asserted to logic-1 when the "101" sequence is detected. Your FSM should also have an active-low asynchronous reset. You may only have 3 states in your state machine. Your FSM should recognize overlapping sequences.

```
module top_module (
    input clk,
    input aresetn,    // Asynchronous active-low reset
    input x,
    output z );

    parameter S0 = 3'b000;
    parameter S1 = 3'b010;
    parameter S2 = 3'b100;
    reg [2:0] curr_state;
    reg [2:0] next_state;
    always @ (*)begin
        case(curr_state)
            S0:next_state= x? S1:S0;
            S1:next_state=~x? S2:S1;
            S2:next_state= x? S1:S0; //S2 = received 10
            default: next_state = curr_state;
        endcase
    end
    always @(posedge clk or negedge aresetn)begin
        if(~aresetn)begin
            curr_state <= 3'b0;
        end
        else begin
            curr_state <= next_state;
        end
    end
    assign z = (curr_state==S2) & (x==1'b1);
endmodule
```