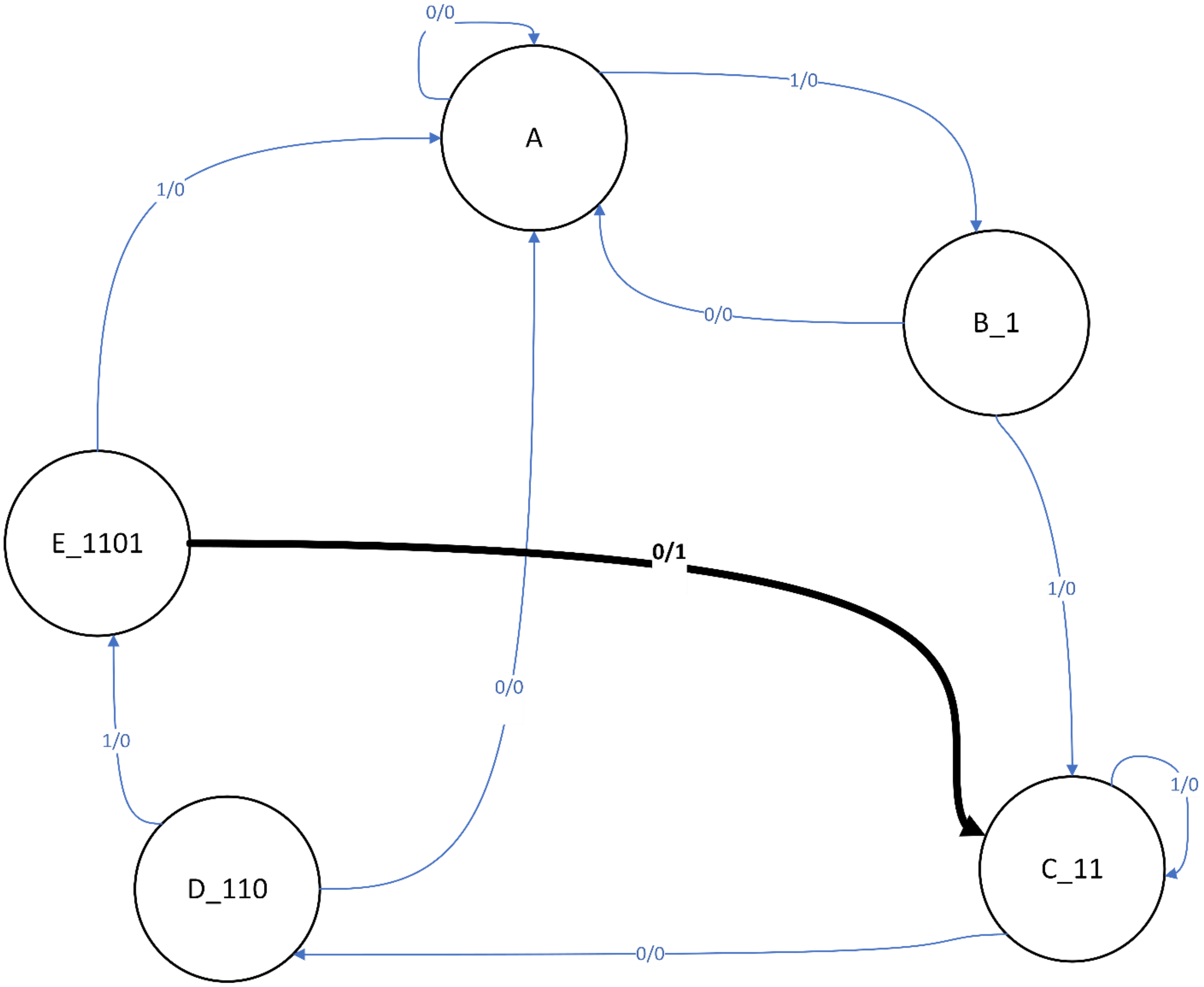
Name: Mohamed Ayman Fathy Group: 1_FPGA

State Diagram



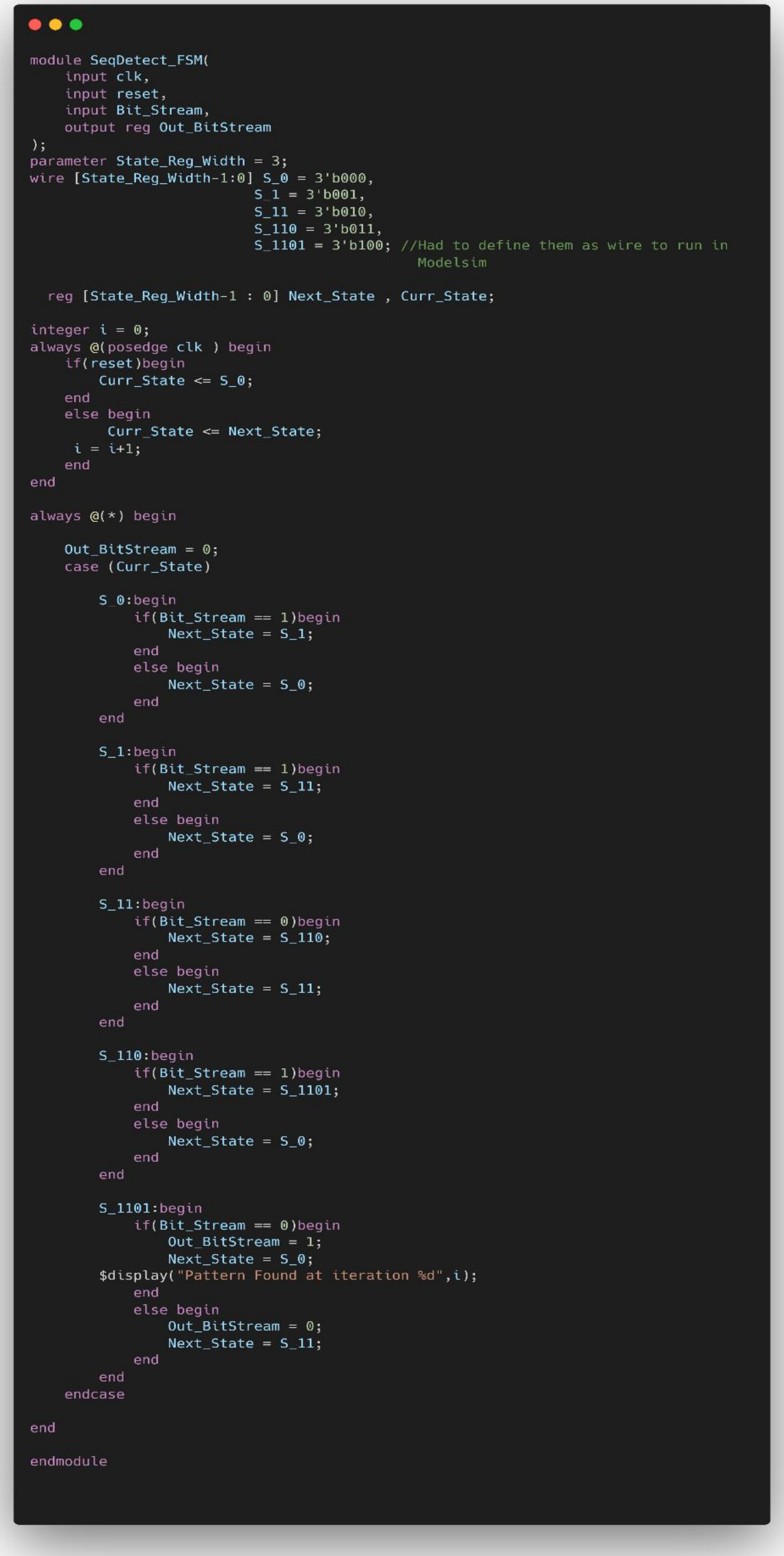
Idea

It’s pretty simple actually where the I go to the next states whenever I find a bit from the sequence as you can see sometimes I don’t find the expected value from the pattern so I keep waiting for it in some cases and some other cases I go back again to the first state and start over.

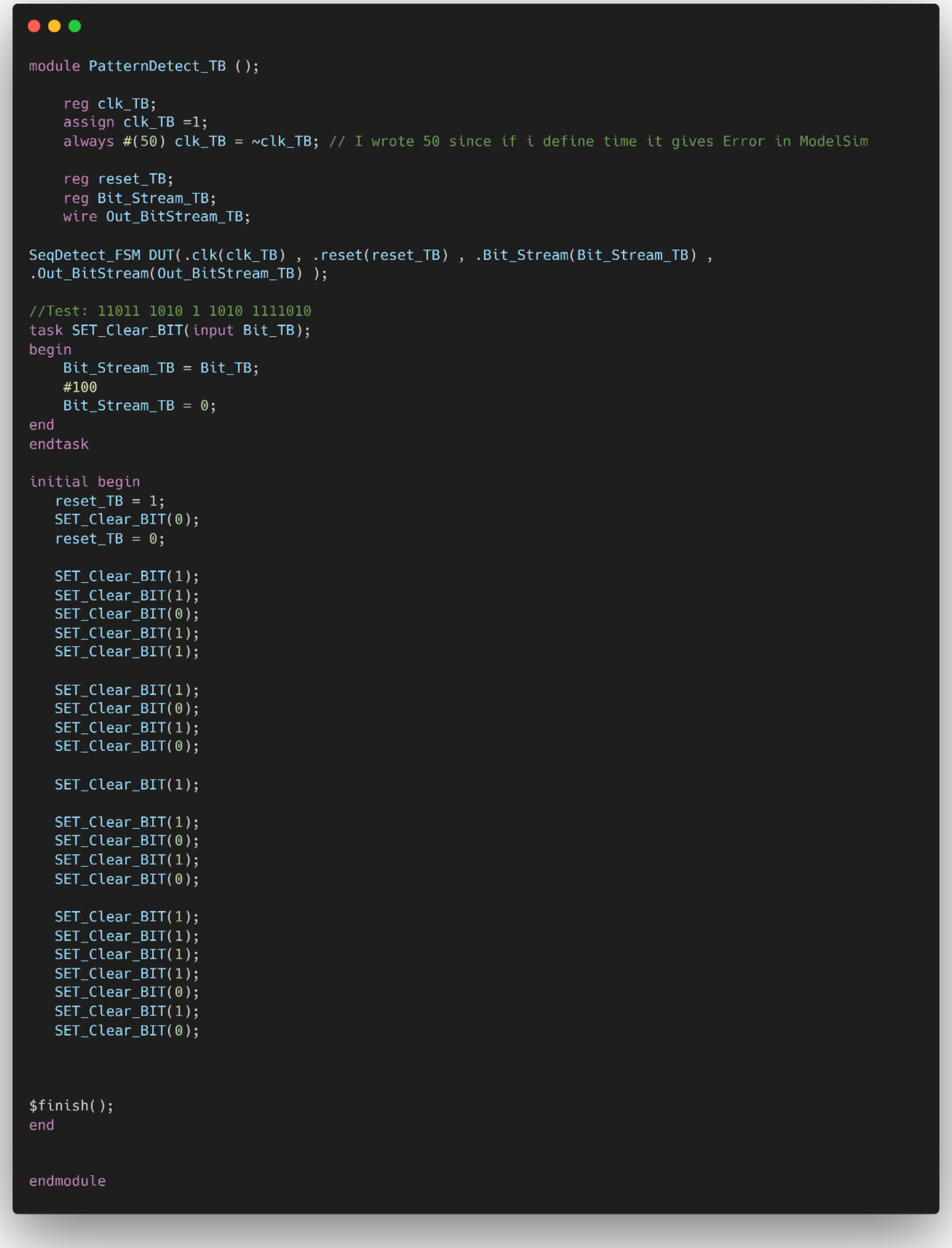
Why do I go to C_11 if I got the last one wrong?

To follow the overlapping technique where if I find 11011 the 11 at the end can be useful if 11011 01000 you could see here it’s better to search for “010” I already have the 11 from the previous pattern that is purely overlapping if I was non -overlapping no pattern will be found because they wouldn’t consider the pattern from the previous iteration.

Code



```
module SeqDetect_FSM(
    input clk,
    input reset,
    input Bit_Stream,
    output reg Out_BitStream
);
parameter State_Reg_Width = 3;
wire [State_Reg_Width-1:0] S_0 = 3'b000,
                          S_1 = 3'b001,
                          S_11 = 3'b010,
                          S_110 = 3'b011,
                          S_1101 = 3'b100; //Had to define them as wire to run in
                                             Modelsim

  reg [State_Reg_Width-1 : 0] Next_State , Curr_State;

integer i = 0;
always @(posedge clk ) begin
    if(reset)begin
        Curr_State <= S_0;
    end
    else begin
         Curr_State <= Next_State;
     i = i+1;
    end
end

always @(*) begin

    Out_BitStream = 0;
    case (Curr_State)

        S_0:begin
            if(Bit_Stream == 1)begin
                Next_State = S_1;
            end
            else begin
                Next_State = S_0;
            end
        end

        S_1:begin
            if(Bit_Stream == 1)begin
                Next_State = S_11;
            end
            else begin
                Next_State = S_0;
            end
        end

        S_11:begin
            if(Bit_Stream == 0)begin
                Next_State = S_110;
            end
            else begin
                Next_State = S_11;
            end
        end

        S_110:begin
            if(Bit_Stream == 1)begin
                Next_State = S_1101;
            end
            else begin
                Next_State = S_0;
            end
        end

        S_1101:begin
            if(Bit_Stream == 0)begin
                Out_BitStream = 1;
                Next_State = S_0;
        $display("Pattern Found at iteration %d",i);
            end
            else begin
                Out_BitStream = 0;
                Next_State = S_11;
            end
        end
    endcase

end

endmodule
```



```
module PatternDetect_TB ();

    reg clk_TB;
    assign clk_TB =1;
    always #(50) clk_TB = ~clk_TB; // I wrote 50 since if i define time it gives Error in ModelSim

    reg reset_TB;
    reg Bit_Stream_TB;
    wire Out_BitStream_TB;

SeqDetect_FSM DUT(.clk(clk_TB) , .reset(reset_TB) , .Bit_Stream(Bit_Stream_TB) ,
.Out_BitStream(Out_BitStream_TB) );

//Test: 11011 1010 1 1010 1111010
task SET_Clear_BIT(input Bit_TB);
begin
    Bit_Stream_TB = Bit_TB;
    #100
    Bit_Stream_TB = 0;
end
endtask

initial begin
   reset_TB = 1;
   SET_Clear_BIT(0);
   reset_TB = 0;

   SET_Clear_BIT(1);
   SET_Clear_BIT(1);
   SET_Clear_BIT(0);
   SET_Clear_BIT(1);
   SET_Clear_BIT(1);

   SET_Clear_BIT(1);
   SET_Clear_BIT(0);
   SET_Clear_BIT(1);
   SET_Clear_BIT(0);

   SET_Clear_BIT(1);

   SET_Clear_BIT(1);
   SET_Clear_BIT(0);
   SET_Clear_BIT(1);
   SET_Clear_BIT(0);

   SET_Clear_BIT(1);
   SET_Clear_BIT(1);
   SET_Clear_BIT(1);
   SET_Clear_BIT(1);
   SET_Clear_BIT(0);
   SET_Clear_BIT(1);
   SET_Clear_BIT(0);



$finish();
end


endmodule
```