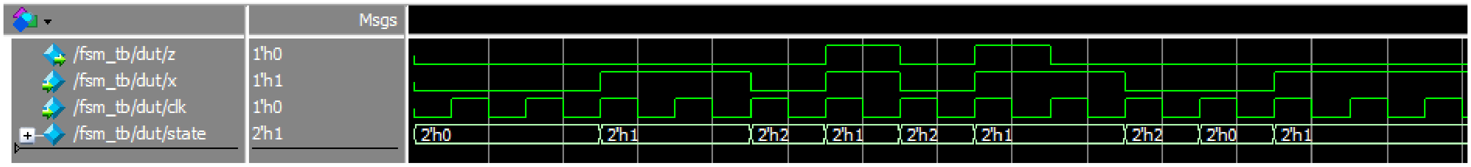
Kevin Hoser
EE 435
Exercise 3
1 April 2020

1. Early in the semester, we designed a Moore FSM that would detect the pattern “101” on an input X, and once the pattern was detected, it would set the single output “Z” to “1”. At all other times output “Z” was set to “0”. We allowed overlap in detecting this pattern.

   Use behavioral Verilog and model this FSM using a case statement. No System Verilog constructs are allowed in this exercise. Explain which case constructs (case, casex or casez) you selected in your implementation and why.

   Write a testbench, and make sure your implementation is correct. Submit your design module, and the waveform of its simulation.

```
module fsm(z, x, clk);

output reg z;
input x, clk;

reg [1:0] state;

initial
begin
    state = 2'b00;
    z = 1'b0;
end

always @(posedge clk) begin
    case (state)
        2'b00: begin
            state = x ? 2'b01 : 2'b00;
            z = 1'b0;
        end

        2'b01: begin
            state = x ? 2'b01 : 2'b10;
            z = 1'b0;
        end

        2'b10: begin
            state = x ? 2'b01 : 2'b00;
            z = x ? 1'b1 : 1'b0;
        end

        default: begin
            state = 2'b00;
            z = 1'b0;
        end

    endcase
end

endmodule
```



I used a case construct since I used 3 out of 4 states in a 2-bit register, where states cannot be reduced with don’t cares.

2. Use a for loop to rewrite the example 7-22 implemented using a while loop. Which implementation is more compact?

```
//Illustration 1: Increment count from 0 to 127. Exit at count 128.
integer count;

initial begin
       for (count = 0; count < 128; count = count +1)
       begin
              $display("Count = %d", count);
              count = count + 1;
       end
end


//Illustration 2: Find the first bit with a value 1 in flag (vector variable)
'define TRUE 1'b1';
'define FALSE 1'b0;
reg [15:0] flag;
integer i; //integer to keep count reg continue;

initial begin
       flag = 16'b 0010_0000_0000_0000;

       continue = 'TRUE;
       for (i = 0; i < 16 && continue; i++)
       begin
              if (flag[i])
              begin
                     $display("Encountered a TRUE bit at element number %d", i);
                     continue = 'FALSE;
              end
       end
end
```

The for loop implementations are more compacted.

3. Can you write a forever loop using a while statement?

Yes, a forever loop can be used inside a while loop.