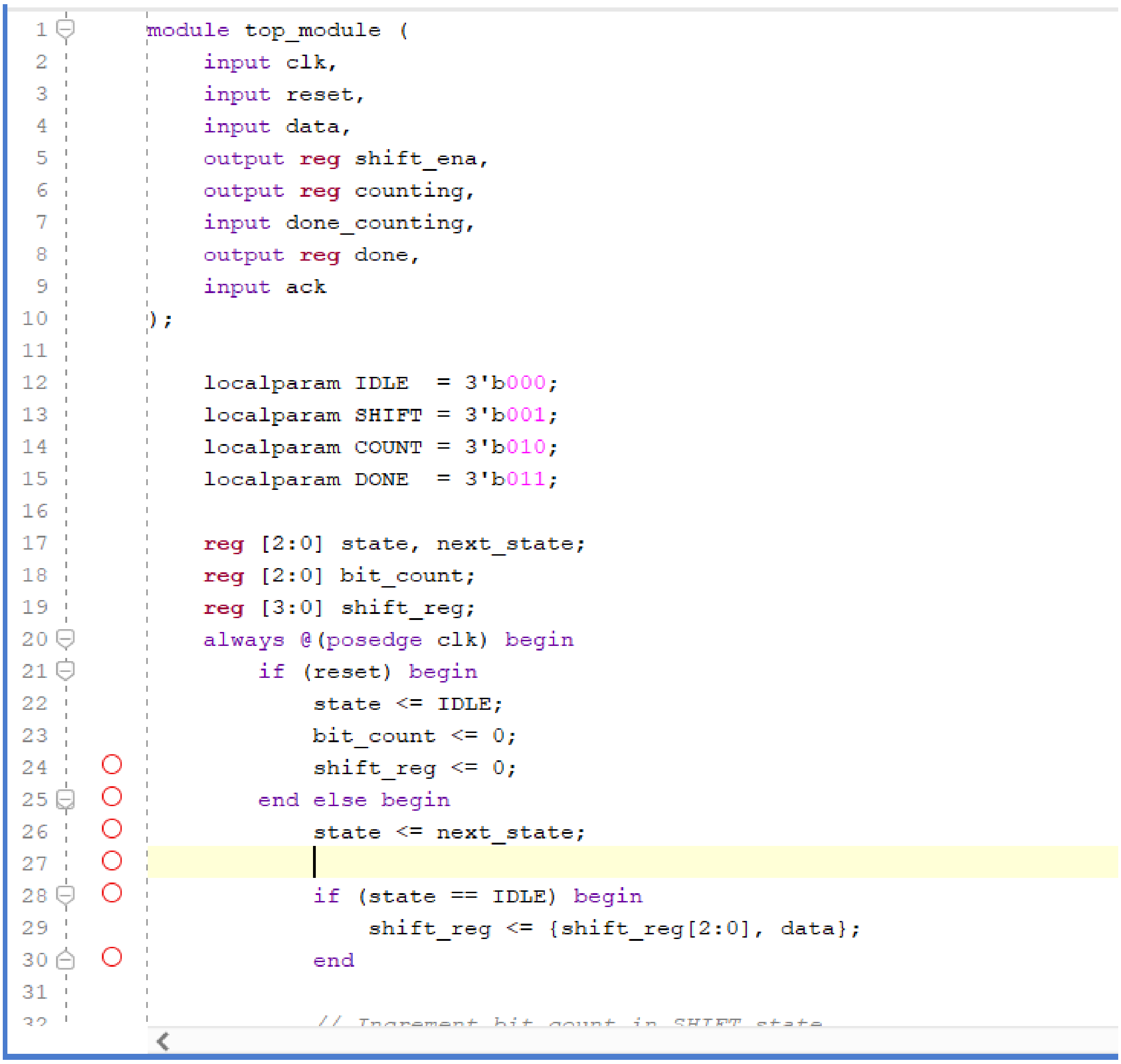
DAY-100
#100DAYSOFRTL

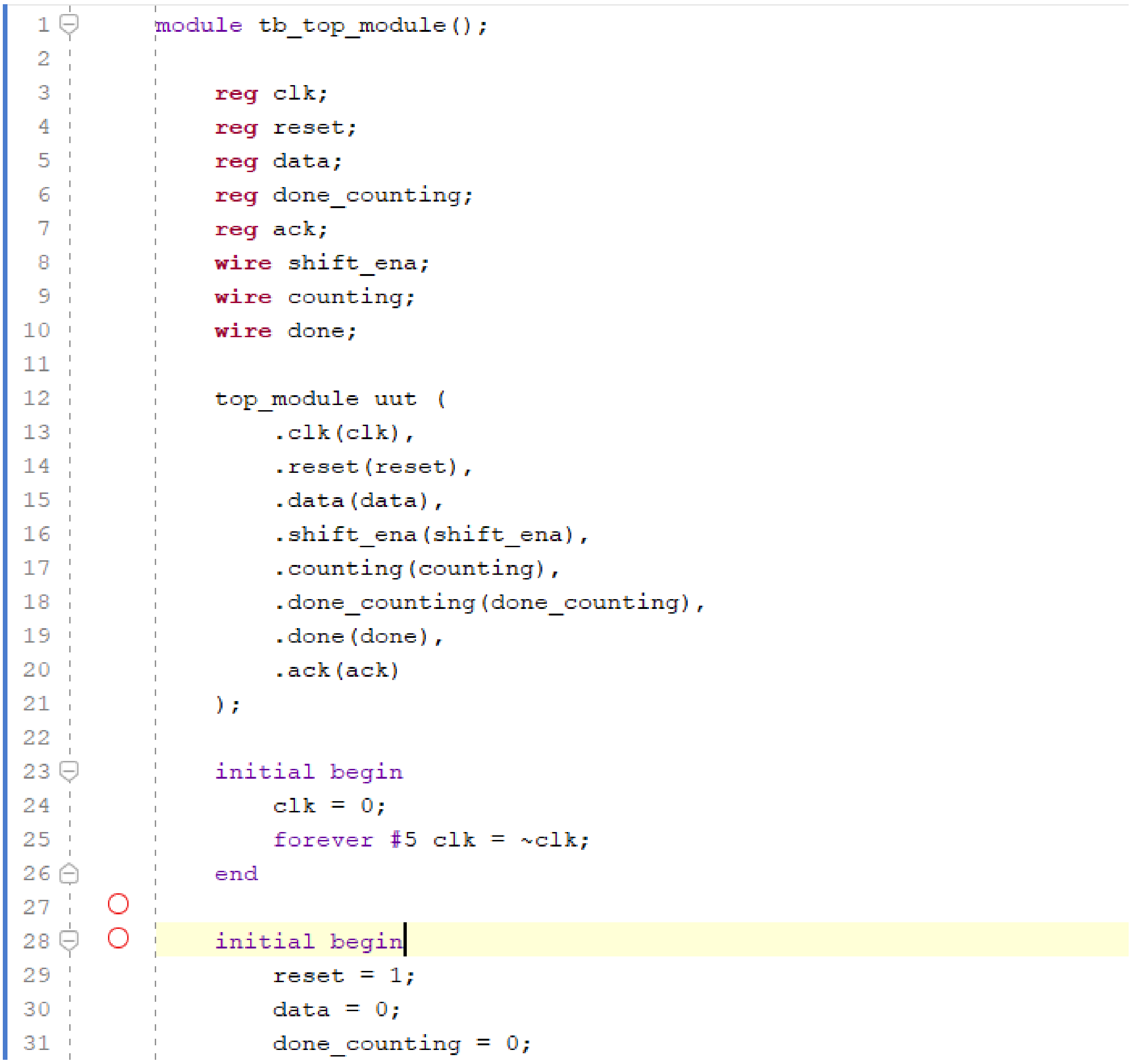
AIM:--- IMPLEMENTATION OF COMPLETE FSM.

VERILOG CODE:--



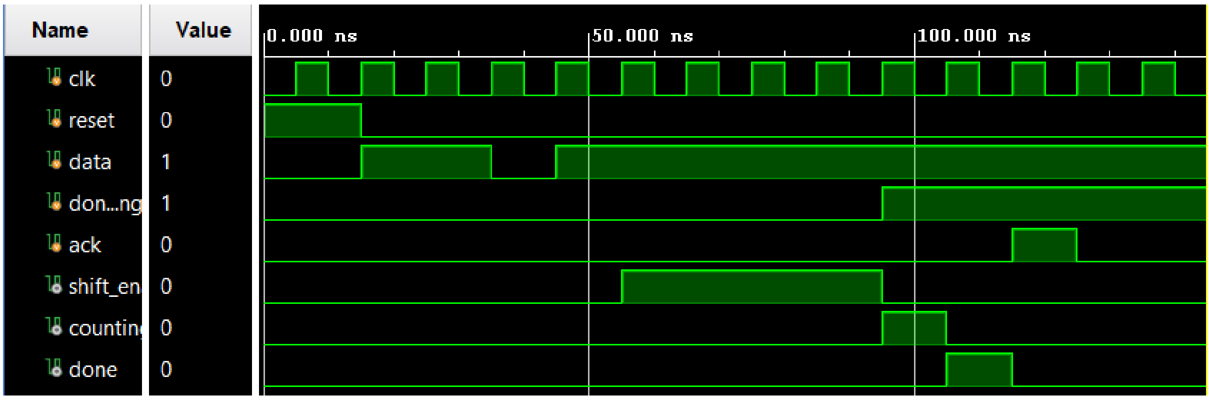
```verilog
module top_module (
    input clk,
    input reset,
    input data,
    output reg shift_ena,
    output reg counting,
    input done_counting,
    output reg done,
    input ack
);

    localparam IDLE  = 3'b000;
    localparam SHIFT = 3'b001;
    localparam COUNT = 3'b010;
    localparam DONE  = 3'b011;

    reg [2:0] state, next_state;
    reg [2:0] bit_count;
    reg [3:0] shift_reg;
    always @(posedge clk) begin
        if (reset) begin
            state <= IDLE;
            bit_count <= 0;
            shift_reg <= 0;
        end else begin
            state <= next_state;

            if (state == IDLE) begin
                shift_reg <= {shift_reg[2:0], data};
            end

            // Increment bit count in SHIFT state
```

TESTBENCH CODE:---



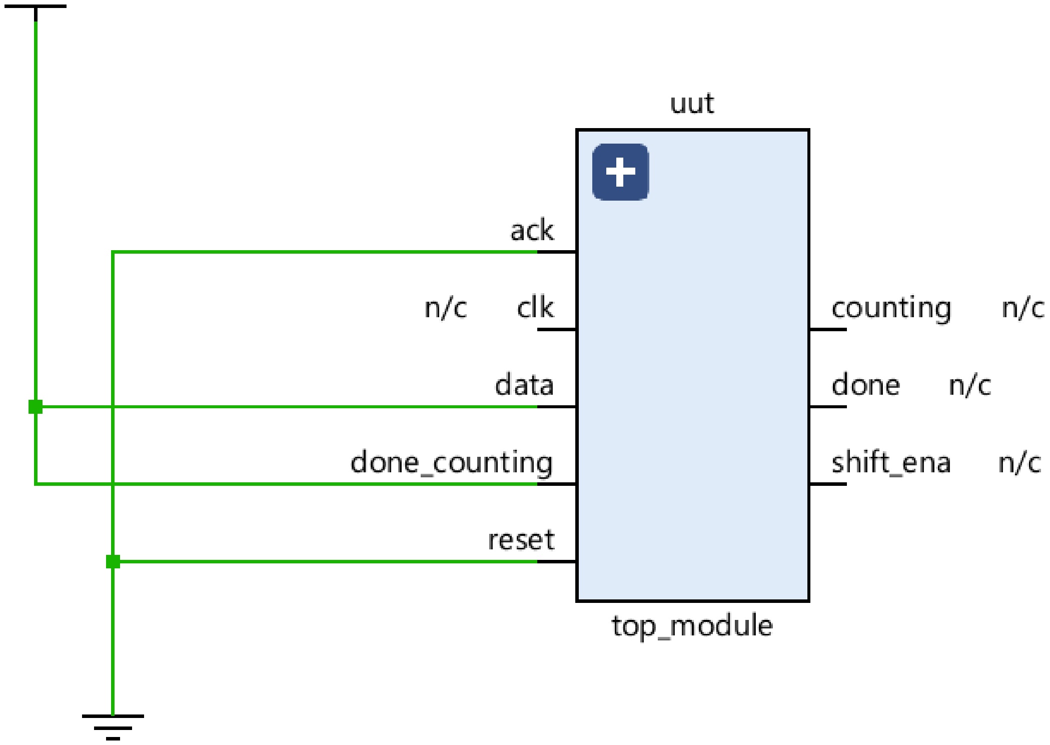
```
module tb_top_module();

    reg clk;
    reg reset;
    reg data;
    reg done_counting;
    reg ack;
    wire shift_ena;
    wire counting;
    wire done;

    top_module uut (
        .clk(clk),
        .reset(reset),
        .data(data),
        .shift_ena(shift_ena),
        .counting(counting),
        .done_counting(done_counting),
        .done(done),
        .ack(ack)
    );

    initial begin
        clk = 0;
        forever #5 clk = ~clk;
    end

    initial begin
        reset = 1;
        data = 0;
        done_counting = 0;
```

WAVEFORM:-----



Name	Value
clk	0
reset	0
data	1
don...ng	1
ack	0
shift_en	0
countin	0
done	0

SCHEMATIC BLOCK :-------


uut
ack
n/c
clk
data
done_counting
reset
counting
n/c
done
n/c
shift_ena
n/c
top_module