```verilog
// (c) Ray Allen 2019
// CPEN 311 Assignment 2 Question 2

module FSMQ2( input clk, restart, pause, goto_third,
              output reg terminal, even, odd,
              output reg [2:0] Out1, Out2);

    reg [2:0] state;

    parameter [2:0] FIRST =  3'b000;
    parameter [2:0] SECOND = 3'b001;
    parameter [2:0] THIRD =  3'b010;
    parameter [2:0] FOURTH = 3'b011;
    parameter [2:0] FIFTH =  3'b100;

    //Sequential stage
    always @(posedge clk) begin
        case(state)
            FIRST: begin
                if (restart | pause) state <= FIRST;
                else state <= SECOND;
            end
            SECOND: begin
                if (restart) state <= FIRST;
                else if (pause) state <= SECOND;
                else state <= THIRD;
            end
            THIRD: begin
                if (restart) state <= FIRST;
                else if (pause) state <= THIRD;
                else state <= FOURTH;
            end
            FOURTH: begin
                if (restart) state <= FIRST;
                else if (pause) state <= FOURTH;
                else state <= FIFTH;
            end
            FIFTH: begin
                if (goto_third) state <= THIRD;
                else if (restart) state <= FIRST;
                else if (pause) state <= FIFTH;
                else state <= FIRST;
            end
            default: state <= FIRST;
        endcase
    end

    //Combinational stage
    always @(*) begin
        case(state)
            FIRST: begin
                terminal = 0;
                Out1 = 3'd3;
                Out2 = 3'd2;
```

```verilog
                    even = 0;
                    odd = 1;
                end
                SECOND: begin
                    terminal = 0;
                    Out1 = 3'd5;
                    Out2 = 3'd4;
                    even = 1;
                    odd = 0;
                end
                THIRD: begin
                    terminal = 0;
                    Out1 = 3'd2;
                    Out2 = 3'd7;
                    even = 0;
                    odd = 1;
                end
                FOURTH: begin
                    terminal = 0;
                    Out1 = 3'd6;
                    Out2 = 3'd3;
                    even = 1;
                    odd = 0;
                end
                FIFTH: begin
                    terminal = 1;
                    Out1 = 3'd5;
                    Out2 = 3'd2;
                    even = 0;
                    odd = 1;
                end
                default: begin
                    terminal = 0;
                    Out1 = 3'd0;
                    Out2 = 3'd0;
                    even = 0;
                    odd = 0;
                end
            endcase
    end
endmodule


module FSMQ2_tb();
    reg clk, restart, pause, goto_third;
    wire terminal, even, odd;
    wire [2:0] Out1, Out2;

    FSMQ2 DUT(clk, restart, pause, goto_third, terminal, even, odd, Out1, Out2);

    initial begin
        forever begin
            clk = 0; #3;
            clk = 1; #3;
```

```verilog
            end
        end

        initial begin
            goto_third = 0;
            restart = 1; #5;
            restart = 0; #5;
            pause = 1; #10;
            pause = 0; #20;
            pause = 1; #10;
            pause = 0; #20;
            restart = 1; #5;
            restart = 0; #5;
            goto_third = 1; #50;
            goto_third = 0;

        end
endmodule
```