```
module fsm(
    input restart,
    input clk,
    input rst,
    input pause,
    input goto_third,
    output [2:0] out1,
    output [2:0] out2,
    output even,
    output odd,
    output terminal
);
//                        state out1, out2, even, odd, terminal
reg [11:0] state;
parameter [11:0] FIRST = 12'b000_011_010_1_0_0;
parameter [11:0] SECOND = 12'b001_101_100_0_1_0;
parameter [11:0] THIRD = 12'b010_010_111_1_0_0;
parameter [11:0] FOURTH = 12'b011_110_011_0_1_0;
parameter [11:0] FIFTH = 12'b100_101_010_1_0_1;

assign out1 = state [8:6];
assign out2 = state [5:3];
assign even = state[2];
assign odd = state[1];
assign terminal = state[0];

always_ff @(posedge clk, posedge rst)
begin
    if (rst) state <= FIRST;
    else
    begin
    case(state)
        FIRST:  if (restart | pause) state <= FIRST;
                else state <= SECOND;
        SECOND: if (restart) state <= FIRST;
                else if (pause) state <= SECOND;
                else state <= THIRD;
        THIRD:  if (restart) state <= FIRST;
                else if (pause) state <= THIRD;
                else state <= FOURTH;
        FOURTH: if (restart) state <= FIRST;
                else if (pause) state <= FOURTH;
                else state <= FIFTH;
        FIFTH:  if (goto_third) state <= THIRD;
                else if (restart) state <= FIRST;
                else if (pause) state <= FIFTH;
        default: state <= FIRST;
    endcase
    end
end
endmodule
```