```
module FSM_0(clk_n, reset, a, b, c, d, e, f, y);
  input clk_n, reset;
  input a,b,c,d,e,f;
  output  y;
  reg [1:0] p_state, n_state;
parameter S0 = 2'd0, S1 = 2'd1, S2 = 2'd2, S3 = 2'd3;
always @(negedge clk_n or posedge reset)
begin
        if(reset)
                        p_state <= S0;
        else
                        p_state <= n_state;
end
always @(p_state or a or b or c or d or e or f)
begin
case (p_state)
S0: begin
                if (a << b)
                n_state = S1;
                else
                n_state = S0;
end
...
S2: begin
                if (e ^ f)
                n_state = S1;
                else
                n_state = S2;
end
default: n_state = S0;
endcase
end
assign y = (p_state == S2);
```