```
//
// Finite State Machine Pattern Detector

module FiniteStateMachine(KEY, LEDR, CLOCK_50);

input [1:0] KEY;
input CLOCK_50;
output [17:0] LEDR;

wire out_1hz;
wire in;
wire out;

wire S0;
wire S1;
wire S0_;
wire S1_;

assign in = ~KEY[0];

// setup a one hertz clock
clock_1hz clock(
   .clk(CLOCK_50),
   .out(out_1hz)
);

// the first flip flop (setting S0)
ped_flip_flop fp_1(
   .clk(out_1hz),
   .D(S0_),
   .Q(S0)
);

// the second flip flop (setting S1)
ped_flip_flop fp_2(
   .clk(out_1hz),
   .D(S1_),
   .Q(S1)
);

// the boolean expression for S0, S1, and out
assign S0_ = in & ((~S0 & S1) | (S0 & ~S1) | (S0 & S1));
assign S1_ = in & ((~S0 & ~S1) | (S0 & ~S1) | (S0 & S1));
assign out = (S0 & S1);

// 17th LED is the clock. 0th LED is the output
assign LEDR[0] = out;
assign LEDR[17] = out_1hz;

endmodule

```