```verilog
module Lab7P2 (SW, KEY, LEDR, LEDG);
  input [17:0] SW;
  input [3:0] KEY;
  output [17:0] LEDR;
  output [8:0] LEDG;

  wire w;
  assign w = SW[1];

  wire Clock;
  assign Clock = KEY[0];
  reg z;

  reg [3:0] y_Q, Y_D; // y_Q represents current state, Y_D represents next state
  parameter A = 4'b0000, B = 4'b0001, C = 4'b0010, D = 4'b0011, E = 4'b0100, F = 4'b0101, G =
  4'b0110, H = 4'b0111, I = 4'b1000;
  always @(w, y_Q)
  begin: state_table
    case (y_Q)
      A: if (!w) Y_D = B;
        else Y_D = F;
      B: if (!w) Y_D = C;
        else Y_D = F;
      C: if (!w) Y_D = D;
        else Y_D = F;
      D: if (!w) Y_D = E;
        else Y_D = F;
      E: if (!w) Y_D = E;
        else Y_D = F;
      F: if (w) Y_D = G;
        else Y_D = B;
      G: if (w) Y_D = H;
        else Y_D = B;
      H: if (w) Y_D = I;
        else Y_D = B;
      I: if (w) Y_D = I;
        else Y_D = B;
      default: Y_D = 4'bxxxx;
    endcase
  end // state_table
  always @(posedge Clock)
  begin: state_FFs
    y_Q = Y_D;
  end // state_FFS

  always
  begin: zset
    case (y_Q)
      E: z = 1;
      I: z = 1;
      default: z = 0;
    endcase
  end
```

```verilog
  assign LEDG[3:0] = y_Q;
  assign LEDR[3:0] = Y_D;

  assign LEDR[17] = z;

  test t1 (SW[17], SW[16], LEDR[17]);
endmodule


module test (A, B, Out);
  input A, B;
  output reg Out;

  always
    case (A)
      0: Out = 0;
      1: Out = B;
    endcase
endmodule
```