

```

1 // Top-level module that defines the I/Os for the DE-1 SoC board
2
3 module DE1_SoC (HEX0, HEX1, HEX2, HEX3, HEX4, HEX5, KEY, LEDR, SW);
4   output logic [6:0] HEX0, HEX1, HEX2, HEX3, HEX4, HEX5;
5   output logic [9:0] LEDR;
6   input logic [3:0] KEY;
7   input logic [9:0] SW;
8
9   // Default values, turns off the HEX displays
10  seg7 zero(.bcd(SW[3:0]), .leds(HEX0));
11  seg7 one (.bcd(SW[7:4]), .leds(HEX1));
12  assign HEX2 = 7'b1111111;
13  assign HEX3 = 7'b1111111;
14  assign HEX4 = 7'b1111111;
15  assign HEX5 = 7'b1111111;
16
17  // Logic to check if SW[3]..SW[0] match your bottom digit,
18  // and SW[7]..SW[4] match the next.
19  // Result should drive LEDR[0].
20  assign LEDR[0] = (SW[0] & SW[1] & SW[2] & ~SW[3] & ~SW[4] & SW[5] & ~SW[6] & ~SW[7]);
21  // 27 = 00100111 //
22 endmodule
23
24 module seg7 (bcd, leds);
25   input logic [3:0] bcd;
26   output logic [6:0] leds;
27   always_comb begin
28     case (bcd)
29       // 3210          6543210
30       4'b0000: leds = 7'b1000000;
31       4'b0001: leds = 7'b1111001;
32       4'b0010: leds = 7'b0100100;
33       4'b0011: leds = 7'b0110000;
34       4'b0100: leds = 7'b0011001;
35       4'b0101: leds = 7'b0010010;
36       4'b0110: leds = 7'b0000010;
37       4'b0111: leds = 7'b1111000;
38       4'b1000: leds = 7'b0000000;
39       4'b1001: leds = 7'b0010000;
40       default: leds = 7'b1111111;
41     endcase
42   end
43 endmodule
44
45 module DE1_SoC_testbench();
46   logic [6:0] HEX0, HEX1, HEX2, HEX3, HEX4, HEX5;
47   logic [9:0] LEDR;
48   logic [3:0] KEY;
49   logic [9:0] SW;
50
51   DE1_SoC dut (.HEX0, .HEX1, .HEX2, .HEX3, .HEX4, .HEX5, .KEY, .LEDR,
52   .SW);
53
54   // Try all combinations of inputs.
55   integer i;
56   initial begin
57     SW[9] = 1'b0;
58     SW[8] = 1'b0;
59     for(i = 0; i < 256; i++) begin
60       SW[7:0] = i; #10;
61     end
62   end
63 endmodule

```