```
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     assign HEX0 = 7'b1111111;
11     assign HEX1 = 7'b1111111;
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
21     // Logic for the last digit, SW[3] = A, SW[0] = D
22     // LEVEL 1
23     logic OUT1, OUT4, OUTAB, OUTCD;
24     not L1A (OUT1, SW[3]);
25     nand L1AB (OUTAB, OUT1, SW[2]);
26     not L1D (OUT4, SW[0]);
27     nand L1CD (OUTCD, SW[1], OUT4);
28     // LEVEL 2
29     logic OUTABCD, OUTABCDinv;
30     nor L2 (OUTABCD, OUTAB, OUTCD);
31     not L2inv (OUTABCDinv, OUTABCD);
32
33
34     // Logic for the second-to-last digit, SW[7] = A, SW[4] = D
35     // LEVEL 1
36     logic OUT1_, OUT3_, OUTAB_, OUTCD_;
37     not L1A_ (OUT1_, SW[7]);
38     nand L1AB_ (OUTAB_, OUT1_, SW[6]);
39     not L1C_ (OUT3_, SW[5]);
40     nand L1CD_ (OUTCD_, OUT3_, SW[4]);
41     // LEVEL 2
42     logic OUTABCD_, OUTABCDinv_;
43     nor L2_ (OUTABCD_, OUTAB_, OUTCD_);
44     not L2inv_ (OUTABCDinv_, OUTABCD_);
45
46
47     // LEVEL 3 (combine both digits)
48     nor OUT (LEDR[0], OUTABCDinv, OUTABCDinv_);
49
50
51  endmodule
52
53  module DE1_SoC_testbench();
54     logic [6:0] HEX0, HEX1, HEX2, HEX3, HEX4, HEX5;
55     logic [9:0] LEDR;
56     logic [3:0] KEY;
57     logic [9:0] SW;
58
59     DE1_SoC dut (.HEX0, .HEX1, .HEX2, .HEX3, .HEX4, .HEX5, .KEY, .LEDR,
60  .SW);
61
62     // Try all combinations of inputs.
63     integer i;
64     initial begin
65        SW[9] = 1'b0;
66        SW[8] = 1'b0;
67        for(i = 0; i <256; i++) begin
68           SW[7:0] = i; #10;
69        end
70     end
71  endmodule
```