```
1   // Eugene Ngo
2   // 1/20/2023
3   // EE 371
4   // Lab 2 Task 1
5
6   // seg7 is a hex display driver that outputs the corresponding hexadecimal value to the hex
    display.
7
8   module seg7 (addr_r, addr_w, data_in, data_out, HEX5, HEX4, HEX3, HEX2, HEX1, HEX0);
9
10     input logic [4:0] addr_r, addr_w;
11     input logic [3:0] data_in, data_out;
12     output logic [6:0] HEX5, HEX4, HEX3, HEX2, HEX1, HEX0;
13
14     // Initiating a hex ram to drive the hex displays
15     logic [15:0][6:0] hex_ram;
16     assign hex_ram[0]  = 7'b1000000; // 0
17     assign hex_ram[1]  = 7'b1111001; // 1
18     assign hex_ram[2]  = 7'b0100100; // 2
19     assign hex_ram[3]  = 7'b0110000; // 3
20     assign hex_ram[4]  = 7'b0011001; // 4
21     assign hex_ram[5]  = 7'b0010010; // 5
22     assign hex_ram[6]  = 7'b0000010; // 6
23     assign hex_ram[7]  = 7'b1111000; // 7
24     assign hex_ram[8]  = 7'b0000000; // 8
25     assign hex_ram[9]  = 7'b0010000; // 9
26     assign hex_ram[10] = 7'b0001000; // a
27     assign hex_ram[11] = 7'b0000011; // b
28     assign hex_ram[12] = 7'b1000110; // c
29     assign hex_ram[13] = 7'b0100001; // d
30     assign hex_ram[14] = 7'b0000110; // e
31     assign hex_ram[15] = 7'b0001110; // f
32
33     // assigning values to output to hex displays
34     assign HEX5 = hex_ram[addr_w[4]];
35     assign HEX4 = hex_ram[addr_w[3:0]];
36     assign HEX3 = hex_ram[addr_r[4]];
37     assign HEX2 = hex_ram[addr_r[3:0]];
38     assign HEX1 = hex_ram[data_in];
39     assign HEX0 = hex_ram[data_out];
40  endmodule
41
42
43  // seg7_testbench tests all expected, unexpected, and edgecase behaviors
44  module seg7_testbench();
45     logic [4:0] addr_r, addr_w;
46     logic [3:0] data_in, data_out;
47     logic [6:0] HEX5, HEX4, HEX3, HEX2, HEX1, HEX0;
48
49     seg7 dut (.addr_r, .addr_w, .data_in, .data_out, .HEX5, .HEX4, .HEX3, .HEX2, .HEX1, .HEX0
    );
50
51     // testing random combinations of inputs
52     initial begin
53        addr_w = 5'b00001; addr_r = 5'b00001; data_in = 4'b0001; data_out = 4'b0001; #10;
54        addr_w = 5'b11111; addr_r = 5'b11111; data_in = 4'b1111; data_out = 4'b1111; #10;
55        addr_w = 5'b10101; addr_r = 5'b10101; data_in = 4'b0101; data_out = 4'b0101; #10;
56        $stop;
57     end
58  endmodule
```