```
1   // Eugene Ngo
2   // 1/20/2023
3   // EE 371
4   // Lab 2 Task 1
5
6   // counter scrolls through the 5-bit address one by one, stopping for approximately one
    second at each address.
7   // The parameter FREQ specifies the number of clock cycles for the counter to increment by
    1. It is set to
8   // 50M by default for the 50MHz clock on the DE1_SoC board.
9
10  module counter #(parameter FREQ=50000000)(addr_r, clk, reset);
11
12     input logic clk, reset;
13     output logic [4:0] addr_r;
14
15     integer count;
16
17     always_ff @(posedge clk) begin
18        // Implementing reset
19        if (reset) begin
20           addr_r <= '0;
21           count <= 0;
22        end
23        // Update addr_r when 1 second has passed
24        else if (count == FREQ) begin
25           if (addr_r == 5'b11111)
26              addr_r <= '0;
27           else
28              addr_r <= addr_r + 5'b00001;
29           count <= 0;
30        end
31        // Increment count when less than 1 second has passed
32        else
33           count <= count + 1;
34     end
35
36  endmodule
37
38  // counter_testbench tests all expected, unexpected, and edgecase behaviors
39  module counter_testbench();
40     logic clk, reset;
41     logic [4:0] addr_r;
42     logic CLOCK_50;
43
44     // Instantiating a 1-clock-cycle counter for testbench purpose
45     counter #(0) dut (.addr_r, .clk(CLOCK_50), .reset);
46
47     // Setting up a simulated clock.
48     parameter CLOCK_PERIOD = 100;
49     initial begin
50        CLOCK_50 <= 0;
51        forever #(CLOCK_PERIOD/2) CLOCK_50 <= ~CLOCK_50; // Forever toggle the clock
52     end
53
54     // Reset the module, then let the counter run
55     initial begin
56        reset <= 1; repeat(5) @(posedge CLOCK_50);
57        reset <= 0; repeat(40) @(posedge CLOCK_50);
58        $stop;
59     end
60  endmodule
```