

```
1  module up_counter (clk, reset, enter, exit, counter);
2
3      input logic clk, reset;
4      input logic enter, exit;
5      output logic [4:0] counter;
6
7      logic [4:0] counter_up;
8
9      always_ff @(posedge clk) begin
10         if (reset) counter_up <= 5'b00000;
11         else if (enter) counter_up <= counter_up + 5'd00001;
12         else if (exit) counter_up <= counter_up - 5'd00001;
13         else counter_up <= counter_up;
14     end
15
16     assign counter = counter_up;
17
18 endmodule
19
20 module up_counter_testbench();
21
22     logic clk, reset;
23     logic enter, exit;
24     logic [4:0] counter;
25
26     up_counter dut (.clk, .reset, .enter, .exit, .counter);
27
28     parameter CLOCK_PERIOD = 100;
29     initial clk = 1;
30     always begin
31         #(CLOCK_PERIOD / 2);
32         clk = ~clk;
33     end
34
35     initial begin
36         @(posedge clk);
37         reset <= 1;
38         @(posedge clk);
39         reset <= 0;
40         @(posedge clk);
41         enter = 0; exit = 0; @(posedge clk);
42         enter = 1; exit = 0; @(posedge clk);
43         enter = 0; exit = 0; @(posedge clk);
44         enter = 1; exit = 0; @(posedge clk);
45         enter = 1; exit = 0; @(posedge clk);
46         enter = 1; exit = 0; @(posedge clk);
47         enter = 1; exit = 0; @(posedge clk);
48         enter = 1; exit = 0; @(posedge clk);
49         enter = 1; exit = 0; @(posedge clk);
50         enter = 1; exit = 0; @(posedge clk);
51         enter = 1; exit = 0; @(posedge clk);
52         enter = 1; exit = 0; @(posedge clk);
53         enter = 1; exit = 0; @(posedge clk);
54         enter = 1; exit = 0; @(posedge clk);
55         enter = 1; exit = 0; @(posedge clk);
56         enter = 1; exit = 0; @(posedge clk);
57         enter = 1; exit = 0; @(posedge clk);
58         enter = 1; exit = 0; @(posedge clk);
59         enter = 1; exit = 0; @(posedge clk);
60         enter = 1; exit = 0; @(posedge clk);
61         enter = 1; exit = 0; @(posedge clk);
62         enter = 1; exit = 0; @(posedge clk);
63         enter = 1; exit = 0; @(posedge clk);
64         enter = 1; exit = 0; @(posedge clk);
65         enter = 1; exit = 0; @(posedge clk);
66         enter = 1; exit = 0; @(posedge clk);
67         enter = 1; exit = 0; @(posedge clk);
68         enter = 1; exit = 0; @(posedge clk);
69         enter = 0; exit = 1; @(posedge clk);
70         enter = 0; exit = 1; @(posedge clk);
71         enter = 0; exit = 1; @(posedge clk);
72         enter = 0; exit = 1; @(posedge clk);
73         enter = 0; exit = 1; @(posedge clk);
74         enter = 0; exit = 1; @(posedge clk);
75         enter = 0; exit = 1; @(posedge clk);
```

```
76     enter = 0; exit = 1; @(posedge clk);
77     enter = 0; exit = 1; @(posedge clk);
78     enter = 0; exit = 1; @(posedge clk);
79     enter = 0; exit = 1; @(posedge clk);
80     enter = 0; exit = 1; @(posedge clk);
81     enter = 0; exit = 1; @(posedge clk);
82     enter = 0; exit = 1; @(posedge clk);
83     enter = 0; exit = 1; @(posedge clk);
84     enter = 0; exit = 1; @(posedge clk);
85     enter = 0; exit = 1; @(posedge clk);
86     enter = 0; exit = 1; @(posedge clk);
87     enter = 0; exit = 1; @(posedge clk);
88     enter = 0; exit = 1; @(posedge clk);
89     enter = 0; exit = 1; @(posedge clk);
90     enter = 0; exit = 1; @(posedge clk);
91     enter = 0; exit = 1; @(posedge clk);
92     enter = 0; exit = 1; @(posedge clk);
93     enter = 0; exit = 1; @(posedge clk);
94     enter = 0; exit = 1; @(posedge clk);
95     $stop;
96     end
97
98 endmodule
99
```