

```
1 // Eugene Ngo
2 // 11/18/2022
3 // EE 271
4 // Lab 4
5
6 // userInput module takes clk, reset, and button as 1-Bit inputs and returns 1-Bit out
7 // This module instatiates the buttons for user input.
8
9 module userInput(out, clk, reset, button);
10     output logic out;
11     input logic clk, reset, button;
12
13     enum {on, off} ps, ns;
14
15     always_comb begin
16         case(ps)
17             on: if(button) ns = on;
18                 else ns = off;
19             off: if(button) ns = on;
20                  else ns = off;
21         endcase
22     end
23
24     assign out = (ps == on & ns == off);
25     //assign pressed = (ns == on);
26
27     always_ff @(posedge clk) begin
28         if(reset)
29             ps <= off;
30         else
31             ps <= ns;
32     end
33 endmodule
34
35 // userInput_testbench tests all expected, unexpected, and edgecase behaviors
36 module userInput_testbench();
37     logic clk, reset;
38     logic button;
39     logic out;
40
41     userInput dut (.out, .clk, .reset, .button);
42
43     parameter CLOCK_PERIOD = 100;
44     initial begin
45         clk <= 0;
46         forever #(CLOCK_PERIOD / 2)
47             clk <= ~clk;
48     end
49
50     initial begin
51         reset <= 1; @ (posedge clk);
52         reset <= 0; @ (posedge clk);
53         button <= 1; @ (posedge clk);
54         button <= 0; @ (posedge clk);
55         button <= 1; @ (posedge clk);
56         button <= 0; @ (posedge clk);
57         button <= 1; @ (posedge clk);
58         button <= 0; @ (posedge clk);
59         button <= 1; @ (posedge clk);
60         button <= 0; @ (posedge clk);
61         button <= 1; @ (posedge clk);
62         button <= 0; @ (posedge clk);
63         button <= 1; @ (posedge clk);
64         button <= 0; @ (posedge clk);
65         button <= 1; @ (posedge clk);
66         button <= 0; @ (posedge clk);
67         button <= 1; @ (posedge clk);
68         button <= 0; @ (posedge clk);
69         button <= 1; @ (posedge clk);
70         button <= 0; @ (posedge clk);
71         button <= 1; @ (posedge clk);
72         button <= 0; @ (posedge clk);
73         button <= 1; @ (posedge clk);
```

```
74         button <= 0;
75
76         button <= 1;
77
78         $stop;
79     end
80 endmodule
```

```
@(posedge clk);
@(posedge clk);
@(posedge clk);
@(posedge clk);
@(posedge clk);
```