```
1   // Eugene Ngo
2   // 1/20/2023
3   // EE 371
4   // Lab 2 Task 3
5
6   // This input_process module is used to makes user button inputs 1 clock cycle long
7
8   module input_process (A, out, clk);
9
10     input logic A, clk;
11     output logic out;
12
13     logic buffer, in;
14
15     // put the input logic through 2 D_FFs to clean up
16     always_ff @(posedge clk) begin
17        buffer <= A;
18        in <= buffer;
19     end
20
21     enum {none, hold} ps, ns;
22
23     // Next state logic
24     always_comb begin
25        case(ps)
26           none: if (in) ns = hold;
27                 else ns = none;
28           hold: if (in) ns = hold;
29                 else ns = none;
30        endcase
31     end
32
33     // Output logic
34     always_comb begin
35        case (ps)
36           none: if (in) out = 1'b1;
37                 else out = 1'b0;
38           hold: out = 1'b0;
39        endcase
40     end
41
42     // DFFs
43     always_ff @(posedge clk) begin
44           ps <= ns;
45     end
46  endmodule
47
48  // input_process_testbench tests all expected, unexpected, and edgecase behaviors
49  module input_process_testbench();
50     logic A, clk, out;
51     logic CLOCK_50;
52
53     input_process dut (.A, .clk(CLOCK_50), .out);
54
55     // Setting up a simulated clock.
56     parameter CLOCK_PERIOD = 100;
57     initial begin
58        CLOCK_50 <= 0;
59        forever #(CLOCK_PERIOD/2) CLOCK_50 <= ~CLOCK_50; // Forever toggle the clock
60     end
61
62     initial begin
63        // press and release for random duration of time
64        A <= 0; repeat(2) @(posedge CLOCK_50);
65        A <= 1; repeat(3) @(posedge CLOCK_50);
66        A <= 0; repeat(2) @(posedge CLOCK_50);
67        A <= 1; repeat(5) @(posedge CLOCK_50);
68        A <= 0; repeat(3) @(posedge CLOCK_50);
69        A <= 1; repeat(8) @(posedge CLOCK_50);
70        A <= 0; repeat(2) @(posedge CLOCK_50);
71        $stop;
72     end
73  endmodule
```