```
// Eugene Ngo
// 1/20/2023
// EE 371
// Lab 2 Task 3

// This input_process module is used to makes user button inputs 1 clock cycle long

module input_process (A, out, clk);

   input logic A, clk;
   output logic out;

   logic buffer, in;

   // put the input logic through 2 D_FFs to clean up
   always_ff @(posedge clk) begin
      buffer <= A;
      in <= buffer;
   end

   enum {none, hold} ps, ns;

   // Next state logic
   always_comb begin
      case(ps)
         none: if (in) ns = hold;
               else ns = none;
         hold: if (in) ns = hold;
               else ns = none;
      endcase
   end

   // Output logic
   always_comb begin
      case (ps)
         none: if (in) out = 1'b1;
               else out = 1'b0;
         hold: out = 1'b0;
      endcase
   end

   // DFFs
   always_ff @(posedge clk) begin
         ps <= ns;
   end
endmodule

// input_process_testbench tests all expected, unexpected, and edgecase behaviors
module input_process_testbench();
   logic A, clk, out;
   logic CLOCK_50;

   input_process dut (.A, .clk(CLOCK_50), .out);

   // Setting up a simulated clock.
   parameter CLOCK_PERIOD = 100;
   initial begin
      CLOCK_50 <= 0;
      forever #(CLOCK_PERIOD/2) CLOCK_50 <= ~CLOCK_50; // Forever toggle the clock
   end

   initial begin
      // press and release for random duration of time
      A <= 0; repeat(2) @(posedge CLOCK_50);
      A <= 1; repeat(3) @(posedge CLOCK_50);
      A <= 0; repeat(2) @(posedge CLOCK_50);
      A <= 1; repeat(5) @(posedge CLOCK_50);
      A <= 0; repeat(3) @(posedge CLOCK_50);
      A <= 1; repeat(8) @(posedge CLOCK_50);
      A <= 0; repeat(2) @(posedge CLOCK_50);
      $stop;
   end
endmodule
```