```
module UserInput(clk, reset, buttons, out);
   input  logic clk, reset;
   input  logic [1:0] buttons;
   output logic [1:0] out;

   logic [1:0] ps, ns;

   always_comb begin
         ns = buttons;
   end

   assign out = (buttons & ~ps);

   always_ff @(posedge clk) begin
      if (reset)
         ps <= 2'b00;
      else
         ps <= ns;
   end

endmodule

module UserInput_testbench();
   logic clk, reset;
   logic [1:0] buttons, out;

   UserInput dut (clk, reset, buttons, out);

   parameter CLOCK_PERIOD=100;
   initial begin
      clk <= 0;
      forever #(CLOCK_PERIOD/2) clk <= ~clk;
   end

   initial begin                              @(posedge clk);
      reset <= 1; buttons <= 2'b0;            @(posedge clk);
      reset <= 0;                             @(posedge clk);
                                              @(posedge clk);
      buttons <= 2'b01;                       @(posedge clk);
      buttons <= 2'b00;                       @(posedge clk);
                                              @(posedge clk);
      buttons <= 2'b10;                       @(posedge clk);
      buttons <= 2'b00;                       @(posedge clk);
                                              @(posedge clk);
      buttons <= 2'b11;                       @(posedge clk);
      buttons <= 2'b00;                       @(posedge clk);
                                              @(posedge clk);
      buttons <= 2'b01;                       @(posedge clk);
                                              @(posedge clk);
                                              @(posedge clk);
      buttons <= 2'b10;                       @(posedge clk);
                                              @(posedge clk);
      buttons <= 2'b01;                       @(posedge clk);
                                              @(posedge clk);
      buttons <= 2'b11;                       @(posedge clk);
                                              @(posedge clk);
      buttons <= 2'b10;                       @(posedge clk);
                                              @(posedge clk);
      buttons <= 2'b11;                       @(posedge clk);
                                              @(posedge clk);
      buttons <= 2'b01;                       @(posedge clk);
      buttons <= 2'b00;                       @(posedge clk);
      $stop;
   end
endmodule

```