```
module UserInput(clk, reset, key, out);

    input logic clk, reset, key;
    output logic out;

    enum {on, off} ps, ns;

    always_comb begin
        case(ps)
            on:     if(key) ns = on;
                    else ns = off;

            off: if(key) ns = on;
                    else ns = off;
        endcase
    end

    assign out = (ps == off & key == 1);

    always_ff @(posedge clk) begin
        if(reset)
            ps <= off;
        else
            ps <= ns;
    end
endmodule

module UserInput_testbench();
    logic clk, reset, key;
    logic out;

    UserInput dut (.clk, .reset, .key, .out);

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

    initial begin
        reset <= 1;     repeat(2) @(posedge clk);
        reset <= 0;     repeat(2) @(posedge clk);
        key <= 1;   repeat(3) @(posedge clk);
        key <= 0;   repeat(3) @(posedge clk);
        key <= 1;   repeat(3) @(posedge clk);
        key <= 0;   repeat(3) @(posedge clk);
        key <= 1;   repeat(3) @(posedge clk);
        key <= 0;   repeat(3) @(posedge clk);
        key <= 1;   repeat(3) @(posedge clk);
        $stop;
    end
endmodule
```