```
/*
   Ben Davis
   Made on 11/17/23
   Re used on 1/12/23
   EE 371, was EE 271
   Lab 1

   This is a module that converts a long button press by a user
   into a single pulse, instead of a long pulse equivalent to
   the user input.

*/

module singlePress (
   input logic clk,        // input of the internal clock
   input logic rst,        // reset button for the machine
   input logic key,        // user input button for game
   output logic onePress   // the output of the button press, always a single pulse
   );

   enum {off, on} ps, ns, temp;  // on and off states of the user input button, with
                           // present and next states as well


   always_ff @(posedge clk or posedge rst) begin
      if (rst) begin  // if the reset button is pushed, game starts over
         temp <= off; // and user input is off
         ps <= temp;
      end else begin
         temp <= ns;
         ps <= temp;    // otherwise, at every posedge of the clk the present
      end             // state is changed to the next state
   end

   // converting long pulse to single
   always_comb begin
      case(ps)
         off: if(key) ns <= on;  // only moves to on state when key is pressed
               else ns <= off;

         on: if (key) ns <= on;  // only reverts back to off when key is let go
               else ns <= off;

      endcase
   end

   assign onePress = (ps == off) & key;  // the output is only 1 when the present
                                         // state is off and key is on.
                                         // ensures output only is 1 at the positive
                                         // edge of the user input pulse, when button
                                         // is first pressed and not just when its held

endmodule

//testbench for module
module singlePress_testbench();

   logic clk, rst, key, onePress; // repeating same logic variables

   singlePress dut (.clk, .rst, .key, .onePress);

   // clock setup
   parameter clock_period = 100;

   initial begin
      clk <= 0;
      forever #(clock_period /2) clk = ~clk;
   end // of clock setup

   initial begin  // tests in module three instances of when the button is held and let go

      rst <= 1;            @(posedge clk);
      rst <= 0; key <= 0;  @(posedge clk);
```

```
                           @(posedge clk);
                           @(posedge clk);
                key <= 1;  @(posedge clk);
                           @(posedge clk);
                           @(posedge clk);
                           @(posedge clk);
                           @(posedge clk);
                key <= 0;  @(posedge clk);
                           @(posedge clk);
                           @(posedge clk);
                key <= 1;  @(posedge clk);
                key <= 0;  @(posedge clk);
                           @(posedge clk);
                key <= 1;  @(posedge clk);
                           @(posedge clk);
                           @(posedge clk);
                key <= 0;  @(posedge clk);
                           @(posedge clk);

      $stop;
   end
endmodule
```