```
1   /*
2      Ben Davis
3      Made on 11/17/23
4      Re used on 1/12/23
5      EE 371, was EE 271
6      Lab 1
7
8      This is a module that converts a long button press by a user
9      into a single pulse, instead of a long pulse equivalent to
10     the user input.
11
12  */
13
14  module singlePress (
15     input logic clk,        // input of the internal clock
16     input logic rst,        // reset button for the machine
17     input logic key,        // user input button for game
18     output logic onePress   // the output of the button press, always a single pulse
19     );
20
21     enum {off, on} ps, ns, temp;  // on and off states of the user input button, with
22                                   // present and next states as well
23
24
25     always_ff @(posedge clk or posedge rst) begin
26        if (rst) begin  // if the reset button is pushed, game starts over
27           temp <= off; // and user input is off
28           ps <= temp;
29        end else begin
30           temp <= ns;
31           ps <= temp;    // otherwise, at every posedge of the clk the present
32        end             // state is changed to the next state
33     end
34
35     // converting long pulse to single
36     always_comb begin
37        case(ps)
38           off: if(key) ns <= on;  // only moves to on state when key is pressed
39                 else ns <= off;
40
41           on: if (key) ns <= on;  // only reverts back to off when key is let go
42                 else ns <= off;
43
44        endcase
45     end
46
47     assign onePress = (ps == off) & key;  // the output is only 1 when the present
48                                           // state is off and key is on.
49                                           // ensures output only is 1 at the positive
50                                           // edge of the user input pulse, when button
51                                           // is first pressed and not just when its held
52
53  endmodule
54
55  //testbench for module
56  module singlePress_testbench();
57
58     logic clk, rst, key, onePress; // repeating same logic variables
59
60     singlePress dut (.clk, .rst, .key, .onePress);
61
62     // clock setup
63     parameter clock_period = 100;
64
65     initial begin
66        clk <= 0;
67        forever #(clock_period /2) clk = ~clk;
68     end // of clock setup
69
70     initial begin  // tests in module three instances of when the button is held and let go
71
72        rst <= 1;             @(posedge clk);
73        rst <= 0; key <= 0;   @(posedge clk);
```

```
74                                  @(posedge clk);
75                                  @(posedge clk);
76                      key <= 1;   @(posedge clk);
77                                  @(posedge clk);
78                                  @(posedge clk);
79                                  @(posedge clk);
80                                  @(posedge clk);
81                      key <= 0;   @(posedge clk);
82                                  @(posedge clk);
83                                  @(posedge clk);
84                      key <= 1;   @(posedge clk);
85                      key <= 0;   @(posedge clk);
86                                  @(posedge clk);
87                      key <= 1;   @(posedge clk);
88                                  @(posedge clk);
89                                  @(posedge clk);
90                      key <= 0;   @(posedge clk);
91                                  @(posedge clk);
92
93          $stop;
94      end
95  endmodule
```