```
1   // Eugene Ngo
2   // 12/2/2022
3   // EE 271
4   // Lab 5
5
6   // compButton takes 1-Bit clk, 1-Bit reset, 10-Bit Q, and 9-Bit SW as inputs and returns
    1-Bit out
7   // This module extends comparator and formats the switch values to be used in comparator
8   module compButton(clk, reset, Q, SW, out);
9      output logic out;
10     input logic clk, reset;
11     input logic [9:0] Q;
12     input logic [8:0] SW;
13
14     logic [9:0] SW_extend;
15
16     assign SW_extend = {1'b0, SW};
17
18     // The comparator module takes in clk, reset, SW_extend to A, Q to B, and returns out to
    value_final
19     // The comparator takes in the selected switch and determines the difficulty and value
    of the CPU
20     comparator computer(.clk, .reset, .A(SW_extend), .B(Q), .value_final(out));
21  endmodule
22
23  // compButton_testbench tests all expected, unexpected, and edgecase behaviors
24  module compButton_testbench();
25     logic out;
26     logic clk, reset;
27     logic [9:0] Q;
28     logic [8:0] SW;
29     logic [9:0] SW_extend;
30
31     compButton dut(.clk, .reset, .Q, .SW, .out);
32
33     parameter CLOCK_PERIOD = 100;
34     initial begin
35        clk <= 0;
36        forever #(CLOCK_PERIOD / 2)
37        clk <= ~clk;
38     end
39
40     initial begin
41        reset <= 1;                                    @(posedge clk);
42                                                       @(posedge clk);
43        reset <= 0;                                    @(posedge clk);
44                                                       @(posedge clk);
45        Q = 10'b0000000001;  SW = 9'b000000010;        @(posedge clk);
46                                                       @(posedge clk);
47        Q = 10'b0000000011;                            @(posedge clk);
48                                                       @(posedge clk);
49        $stop;
50     end
51  endmodule
```