```
1   module flipflop (clk, reset, D, Q);
2
3      // Just a basic flipflop
4
5      input logic clk, reset, D;
6      output logic Q;
7
8      //initial Q <= 0;
9
10     always_ff @(posedge clk) begin
11        if (reset)
12           Q <= 0;
13        else
14           Q <= D;
15     end
16
17  endmodule
18
19  module flipflop_testbench();
20     logic clk, reset, D, Q;
21
22     flipflop dut (clk, reset, D, Q);
23     // Set up the clock.
24     parameter CLOCK_PERIOD=100;
25     initial begin
26     clk <= 0;
27     forever #(CLOCK_PERIOD/2) clk <= ~clk;
28     end
29
30     // Set up the inputs to the design. Each line is a clock cycle.
31     initial begin
32                                 @(posedge clk);
33        reset <= 1;              @(posedge clk);
34        reset <= 0; D <= 0;      @(posedge clk);
35                                 @(posedge clk);
36                                 @(posedge clk);
37                                 @(posedge clk);
38        D <= 1;                  @(posedge clk);
39                                 @(posedge clk);
40                                 @(posedge clk);
41                                 @(posedge clk);
42        D <= 0;                  @(posedge clk);
43                                 @(posedge clk);
44                                 @(posedge clk);
45                                 @(posedge clk);
46        $stop; // End the simulation.
47     end
48  endmodule
49
```