```
// An Nguyen, Darren Do
// 3/7/24
// EE 371
// LAB 6, task#2
//
// Inputs:
// clk: Timer to indicate the functionality of our module.
// reset: Not used in this module.
// in: input that's going into the double D flip flops.
//
// Outputs:
// out: output from the double D flip flops.
//
// logic:
// out_ff1: take signal from first D flip flop to the second.
module flipFlop (clk, reset, in, out);
   input logic clk, reset;
   input logic in;
   output logic out;
   logic out_ff1;

   // First Flip Flop
   always_ff @(posedge clk) begin
      out_ff1 <= in;
   end

   // Second Flip Flop
   always_ff @(posedge clk) begin
      out <= out_ff1;
   end
endmodule

// This testbench's only job is to show how an input is going through the double
// flip flop.
module flipFlop_testbench();
   logic clk, reset;
   logic in;
   logic out;

   flipFlop dut (.clk, .reset, .in, .out);

   parameter CLOCK_PERIOD = 100;
   initial begin
     clk <= 0;
     forever #(CLOCK_PERIOD / 2) clk <= ~clk;
   end
   // reset and then show the input signal coming in
   initial begin
      in <= 1'b0; reset <= 1'b1; repeat(4) @(posedge clk);
      reset <= 1'b0;         @(posedge clk);
      in <= 1'b1;  repeat(2) @(posedge clk);
      in <= 1'b0;  repeat(4) @(posedge clk);
      in <= 1'b1;  repeat(6) @(posedge clk);
      in <= 1'b0;  repeat(4) @(posedge clk);
      in <= 1'b1;  repeat(1) @(posedge clk);
      in <= 1'b0;  repeat(3) @(posedge clk);
      $stop;
   end

endmodule
```