```systemverilog
module flipflop (clk, reset, D, Q);

   // Just a basic flipflop

   input logic clk, reset, D;
   output logic Q;

   //initial Q <= 0;

   always_ff @(posedge clk) begin
      if (reset)
         Q <= 0;
      else
         Q <= D;
   end

endmodule

module flipflop_testbench ();
   logic clk, reset, D, Q;

   flipflop dut (clk, reset, D, Q);
   // Set up the clock.
   parameter CLOCK_PERIOD=100;
   initial begin
   clk <= 0;
   forever #(CLOCK_PERIOD/2) clk <= ~clk;
   end

   // Set up the inputs to the design. Each line is a clock cycle.
   initial begin
                              @(posedge clk);
      reset <= 1;             @(posedge clk);
      reset <= 0; D <= 0;     @(posedge clk);
                              @(posedge clk);
                              @(posedge clk);
                              @(posedge clk);
      D <= 1;                 @(posedge clk);
                              @(posedge clk);
                              @(posedge clk);
                              @(posedge clk);
      D <= 0;                 @(posedge clk);
                              @(posedge clk);
                              @(posedge clk);
                              @(posedge clk);
      $stop; // End the simulation.
   end
endmodule

```