```
`timescale 1ns/1ns

module pattern2_testbench();
    reg clk;
    reg reset;
    reg sig;
    wire out;
    reg fi = 1'b0;
    initial clk = 0;
    initial reset = 1'b0;
    initial sig = 1'b0;

    always #(10) clk <= ~clk;

    pattern dut(
        .clk(clk),
        .rst(reset),
        .sig(sig),
        .out(out)
    );

    initial begin
        $dumpfile("pattern2_testbench.vcd");
        $dumpvars(0, pattern2_testbench);

        @(posedge clk); #1;
        reset = 1;
        @(posedge clk); #1;
        reset = 0;
        sig = 1;
        @(posedge clk); #1;
        sig = 0;
        repeat (2) @(posedge clk); #1;
        sig = 1;
        @(posedge clk); #1;
        sig = 0;
        @(posedge clk); #1;
        if (out == 1'b0) fi = 1'b1;
        repeat (50) @(posedge clk); #1;
        if (out == 1'b1) fi = 1'b1;
        @(posedge clk); #1;
        sig = 1;
        @(posedge clk); #1;
        sig = 0;
        @(posedge clk); #1;
        sig = 1;
        @(posedge clk); #1;
        sig = 0;
        @(posedge clk); #1;
        @(posedge clk); #1;
        sig = 1;
        @(posedge clk); #1;
        @(posedge clk); #1;
        sig = 0;
        @(posedge clk); #1;
        @(posedge clk); #1;
        sig = 1;
        @(posedge clk); #1;
        sig = 0;
        @(posedge clk); #1;
        if (out == 1'b0) fi = 1'b1;
        @(posedge clk); #1;
        sig = 1;
        @(posedge clk); #1;
        sig = 0;
        @(posedge clk); #1;
        if (out == 1'b0) fi = 1'b1;
        $display("Out: %0d", out);
        if (fi) $display("Fail");
        else $display("Pass");
        $finish();
    end
endmodule
```