```
`timescale 1ns/1ns

module count_testbench();
    reg clk;
    reg en;
    reg up;
    reg reset;
    wire [15:0] out;
    initial clk = 0;
    initial en = 1'b0;
    initial up = 1'b0;
    initial reset = 1'b0;
    always #(10) clk <= ~clk;

    count tu(
        .clk(clk),
        .en(en),
        .up(up),
        .reset(reset),
        .out_data(out)
    );

    initial begin
        $dumpfile("count_testbench.vcd");
        $dumpvars(0, count_testbench);
        @(posedge clk); #1;
        reset = 1;
        @(posedge clk); #1;
        en = 1;
        up = 1;
        reset = 0;
        repeat (50) @(posedge clk); #1;
        $display("Count: %0d", out);
        @(posedge clk); #1;
        up = 0;
        repeat (26) @(posedge clk); #1;
        $display("Count: %0d", out);
        if (out == 25) begin
            $display("Pass");
        end else begin
            $display("Fail");
        end
        $finish();
    end
endmodule
```