

Come lanciare un testbench

```
// iverilog -g2005-sv -s sillyfunction_testbench lesson1.sv && a.out
```

Esempio **module endianness** con il suo testbench

```
// https://en.wikipedia.org/wiki/Endianness
// https://en.wikipedia.org/wiki/Row_and_column-major_order
// https://en.wikipedia.org/wiki/Matrix_calculus#Layout_conventions

module endianness(input logic [2:-1] a, output logic y);
    assign y = &a | a[-1];
endmodule

module endianness_testbench();
    logic [3:0] a;
    logic y;

    endianness dut(a, y);

    initial begin
        $monitor($time, " a=%b y=%b", a, y);
        #10
        a = 1;
        #10
        a = 0;
        #10
        $finish();
    end
endmodule
```

Esempio **module sillyfunction** con il suo testbench

```
module sillyfunction(
    input logic a, b, c,
    output logic y
);
    assign y = ~a & ~b & ~c
            | a & ~b & ~c
            | a & ~b & c;
endmodule

module sillyfunction_testbench();
    logic a, b, c, y;

    sillyfunction dut(a, b, c, y);

    initial begin
        $monitor("%4d a=%b b=%b c=%b y=%b", $time, a, b, c, y);
        a = 0; b = 0; c = 0;
        #10
        b = 1;
```

```

        #10
        $finish;
    end
endmodule

```

Esempio **module fulladder** con il suo testbench

```

module fulladder(
    input logic a, b, cin,
    output logic s, cout
);
    logic p, g;

    assign s = p ^ cin;
    assign cout = g | (p & cin);

    assign p = a ^ b;
    assign g = a & b;
endmodule

module fulladder_testbench();
    logic a, b, cin, s, cout;

    fulladder dut(a, b, cin, s, cout);

    initial begin
        $monitor("%4d a=%b b=%b cin=%b s=%b cout=%b", $time, a, b, cin, s, cout);
        a = 0; b = 0; cin = 0;
        #10
        a = 1;
        #10
        b = 1;
        #10
        cin = 1;
        #10
        b = 0;
        #10
        $finish;
    end
endmodule

```

Esempio **module tristate** con il suo testbench

```

module tristate(
    input logic [3:0] a,
    input logic      en,
    output tri     [3:0] y
);
    assign y = en ? a : 4'bz;
endmodule

```

```
module tristate_testbench();
```

```

logic [3:0] a;
logic en;
tri [3:0] y;

tristate dut(a, en, y);

initial begin
    $monitor("%4d a=%b en=%b y=%b", $time, a, en, y);
    a = 0; en = 0;
    #10
    en = 1;
    #10
    a = 1;
    #10
    en = 0;
    #10
    $finish;
end
endmodule

```

Esempio module mux2 con il suo testbench

```

module mux2(
    input logic [3:0] d0, d1,
    input logic      s,
    output tri     [3:0] y
);
    tristate t0(d0, ~s, y), t1(d1, s, y);
endmodule

module mux2_testbench();
    logic [3:0] d0, d1;
    logic s;
    tri [3:0] y;

    mux2 dut(d0, d1, s, y);

    initial begin
        $monitor($time, " d0=%b d1=%b s=%b y=%b", d0, d1, s, y);
        #10
        d0 = 0; d1 = 1; s = 0;
        #10
        s = 1;
        #10
        $finish();
    end
endmodule

```

Esempio module example con il suo testbench

```

module example(
    input logic a, b, c,
    output logic y

```

```

);
logic ab, bb, cb, n1, n2, n3;

assign #1 {ab, bb, cb} = ~{a, b, c};
assign #2 n1 = ab & bb & cb;
assign #2 n2 = a & bb & cb;
assign #2 n3 = a & bb & c;
assign #4 y = n1 | n2 | n3;
endmodule

module example_testbench();
    logic a, b, c, y;

    example dut(a, b, c, y);

    initial begin
        $monitor("%4d a=%b b=%b c=%b y=%b ab=%b", $time, a, b, c, y,
dut.ab);
        $dumpfile("example.vcd");
        $dumpvars;
        a = 0; b = 0; c = 0;
        #10
        a = 1;
        #10
        b = 1;
        #10
        c = 1;
        #10
        $finish;
    end
endmodule

```