

Anatomy of a Verilog Testbench - Overview

```
`timescale 1 ns/1 ns
module TestBench();
  reg [17:0] SW;
  reg [3:0] KEY;
  wire [6:0] LEDR, HEX0;

  Lab1 selector(.KEY(KEY),.SW(SW),.LEDR(LEDR),.HEX0(HEX0))

  initial begin
    KEY[3]=0; SW[10]=0; SW[0]=0 ; #5;
    KEY[3]=0; SW[10]=0; SW[0]=1 ; #5;
    KEY[3]=0; SW[10]=1; SW[0]=0 ; #5;
    KEY[3]=0; SW[10]=1; SW[0]=1 ; #5;
    KEY[3]=1; SW[10]=0; SW[0]=0 ; #5;
    KEY[3]=1; SW[10]=0; SW[0]=1 ; #5;
    KEY[3]=1; SW[10]=1; SW[0]=0 ; #5;
    KEY[3]=1; SW[10]=1; SW[0]=1 ; #5;
  end
endmodule
```

Inputs signals to the module we're testing. Use **reg** keyword (we'll learn why later)

Output signals from the module we're testing. Use **wire** keyword.

Instance of your module **Lab1**: Module name
selector: Instance name

initial begin: Code is executed once at the start of simulation
This block sets the inputs to various values.
First line: set KEY[3]=0, SW[10]=0, etc.

Anatomy of a Verilog Testbench – Inputs/Outputs

```
timescale 1 ns/1 ns
module TestBench();
  reg [17:0] SW;
  reg [3:0] KEY;
  wire [6:0] LEDR, HEX0;

  Lab1 selector (.KEY(KEY), .SW(SW), LEDR(LED), .HEX0(HEX0))

  initial begin
    KEY[3]=0; SW[10]=0; SW[0]=0 ; #5;
    KEY[3]=0; SW[10]=0; SW[0]=1 ; #5;
    KEY[3]=0; SW[10]=1; SW[0]=0 ; #5;
    KEY[3]=0; SW[10]=1; SW[0]=1 ; #5;
    KEY[3]=1; SW[10]=0; SW[0]=0 ; #5;
    KEY[3]=1; SW[10]=0; SW[0]=1 ; #5;
    KEY[3]=1; SW[10]=1; SW[0]=0 ; #5;
    KEY[3]=1; SW[10]=1; SW[0]=1 ; #5;
  end
endmodule
```



.KEY(KEY): Connect the **register** named “KEY” to the **input** on the Lab1 selector module named KEY

.LEDR(LED): Connect the wire named “LEDR” to the **output** named LEDR

Anatomy of a Verilog Testbench - Timing

```
`timescale 1 ns/1 ns
```

`timescale: Defines what 1 time step (#1) means in terms of real time.
<time_units>/<time_precision>

```
module TestBench();  
  reg [17:0] SW;  
  reg [3:0] KEY;  
  wire [6:0] LEDR, HEX0;  
  
  Lab1 selector(.KEY(KEY),.SW(SW),.LEDR(LEDR),.HEX0(HEX0))  
  
  initial begin  
    KEY[3]=0; SW[10]=0; SW[0]=0 ; #5;  
    KEY[3]=0; SW[10]=0; SW[0]=1 ; #5;  
    KEY[3]=0; SW[10]=1; SW[0]=0 ; #5;  
    KEY[3]=0; SW[10]=1; SW[0]=1 ; #5;  
    KEY[3]=1; SW[10]=0; SW[0]=0 ; #5;  
    KEY[3]=1; SW[10]=0; SW[0]=1 ; #5;  
    KEY[3]=1; SW[10]=1; SW[0]=0 ; #5;  
    KEY[3]=1; SW[10]=1; SW[0]=1 ; #5;  
  end  
endmodule
```

#5: After each test pattern.
delay sim by 5 time units (5ns)

Example: SW[0] is set to 0 initially, then changes to 1 after 5ns

