Week 1 RISC V Tape-Out Report

By

Dr M A. Raheem,

Department of Electronics and Communication Engineering

Muffakham Jah College of Engineering and Technology

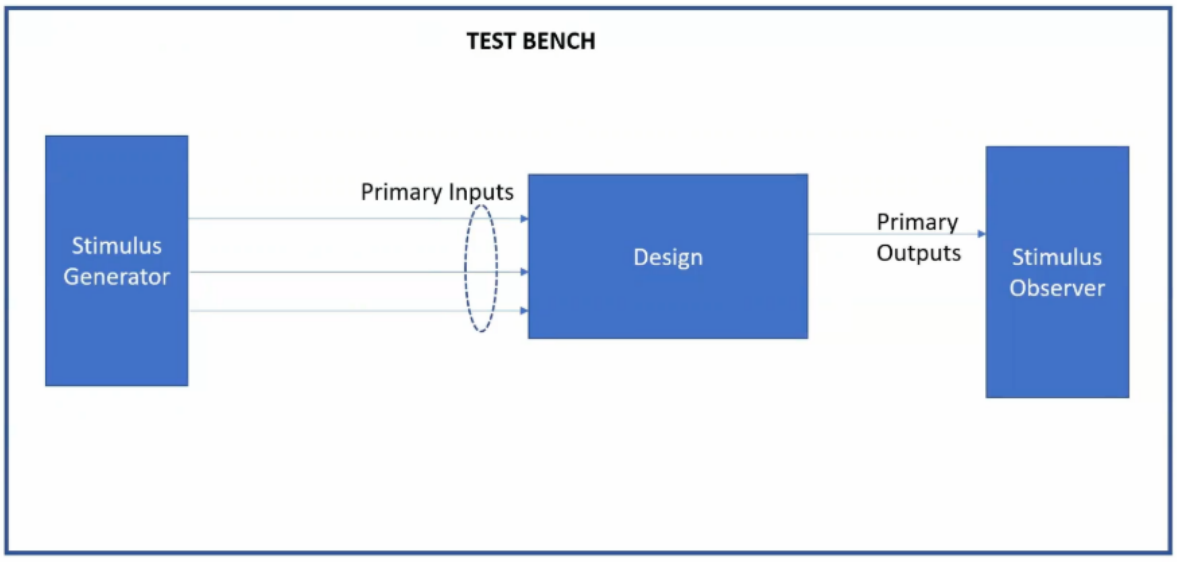
Banjara Hills, Hyderabad-500 034

Role of a Simulator

- A **simulator** is used to verify that an RTL (Register Transfer Level) design adheres to its intended specifications.
- Simulation applies test inputs (stimulus) to the design and observes the outputs, allowing designers to confirm correctness before hardware implementation.

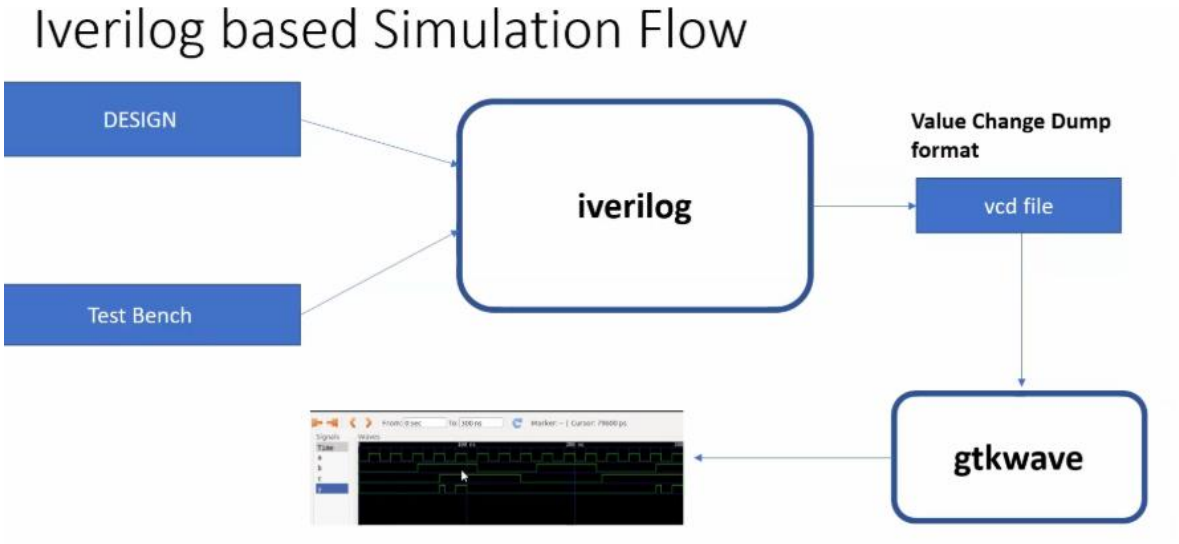
Design and Test Bench

- **Design**: Actual Verilog code (may span multiple files) representing the hardware's intended functionality.
- **Test Bench (TB)**: A Verilog module that *instantiates* the design, applies stimulus (test vectors), and observes outputs to validate the design's behavior.
    - The test bench does **not** have primary inputs or outputs; these are properties of the design under test.
    - The TB generates input stimuli and monitors output responses internally.



Simulator Operation

- The simulator evaluates outputs **only when input signals change**.
- Simulation tools generate a **VCD (Value Change Dump) file**, capturing signal changes over time — essential for timing and functional analysis.



- **Design and test bench files** are compiled together using iVerilog.
- The compiled design is executed, producing simulation results.
- **GTKWave** or similar tools open the VCD file to visually inspect input and output waveforms.

Lab Steps

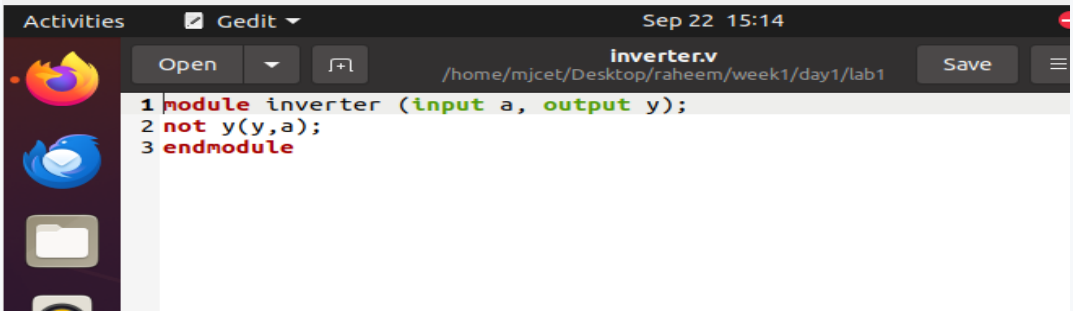
1. Prepare Your Environment

- Ensure iVerilog and GTKWave are installed on your system.

2. Write the Design

Example: An **Inverter** (NOT gate), saved as `inverter.v`:

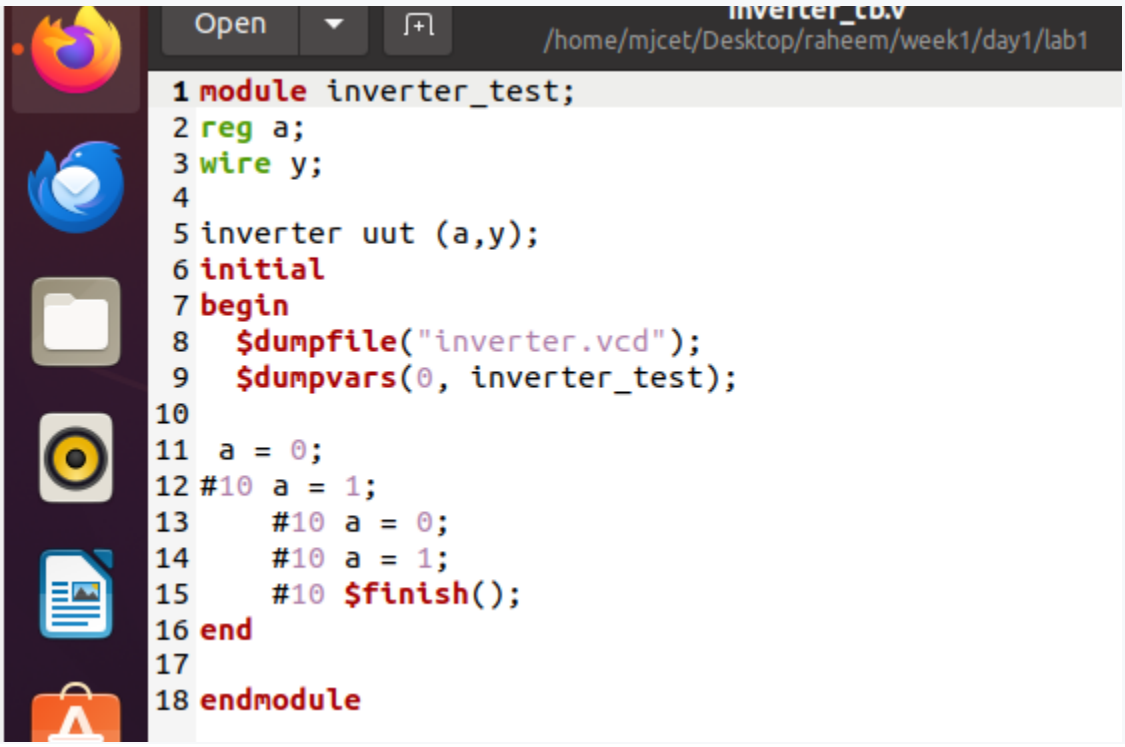
```
module inverter(
    input a,
    output y
);
    not(y, a);
endmodule
```



3. Write the Test Bench

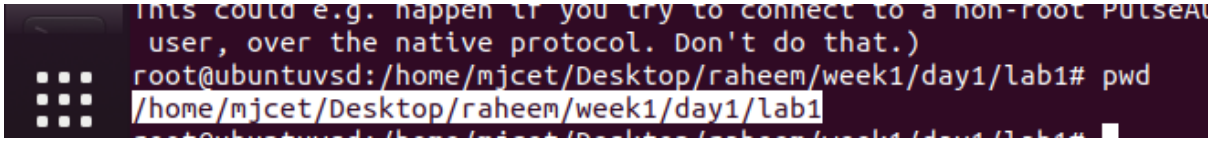
Save the following as `inverter_tb.v` in the same directory:

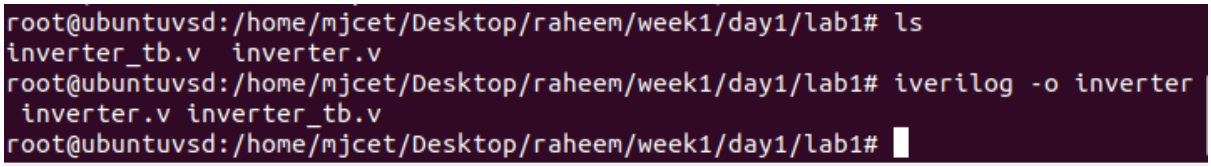
```
module inverter_test;

reg a;
wire y;

inverter uut(a, y); // Instantiate inverter

initial begin
    $dumpfile("inverter.vcd");  // Set VCD output filename
    $dumpvars(0, inverter_test); // Dump all variables in scope

    a = 0;
    #10 a = 1;
    #10 a = 0;
    #10 a = 1;
    #10 $finish();
end

endmodule
```



Open Terminal in Project Directory

Navigate to the directory containing both `inverter.v` and `inverter_tb.v`. On Linux/Mac:

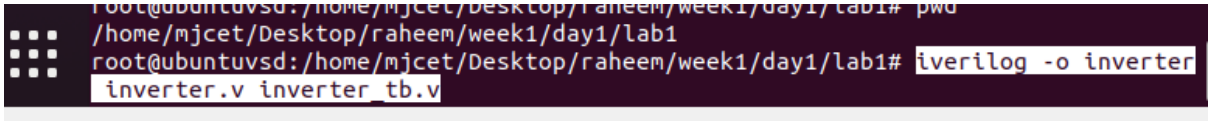




. Compile the Design and Test Bench

Use iVerilog to compile both files:

```
iverilog -o inverter inverter.v inverter_tb.v
```

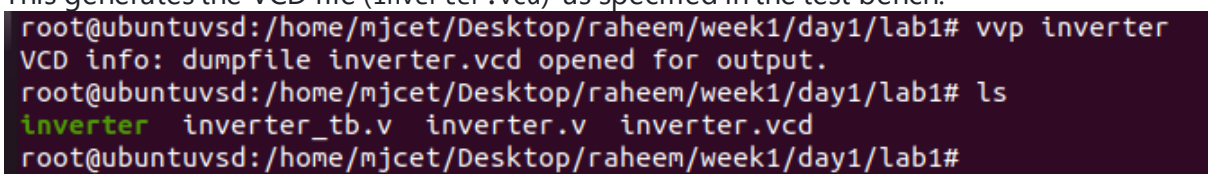


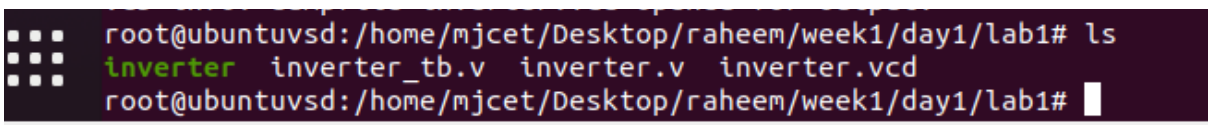
- `-o inverter` sets the name of the compiled output.
- Both source files are specified.

Run the Simulation

Execute the compiled simulation:

```
vvp inverter
```

- This generates the VCD file (`inverter.vcd`) as specified in the test bench.
- 

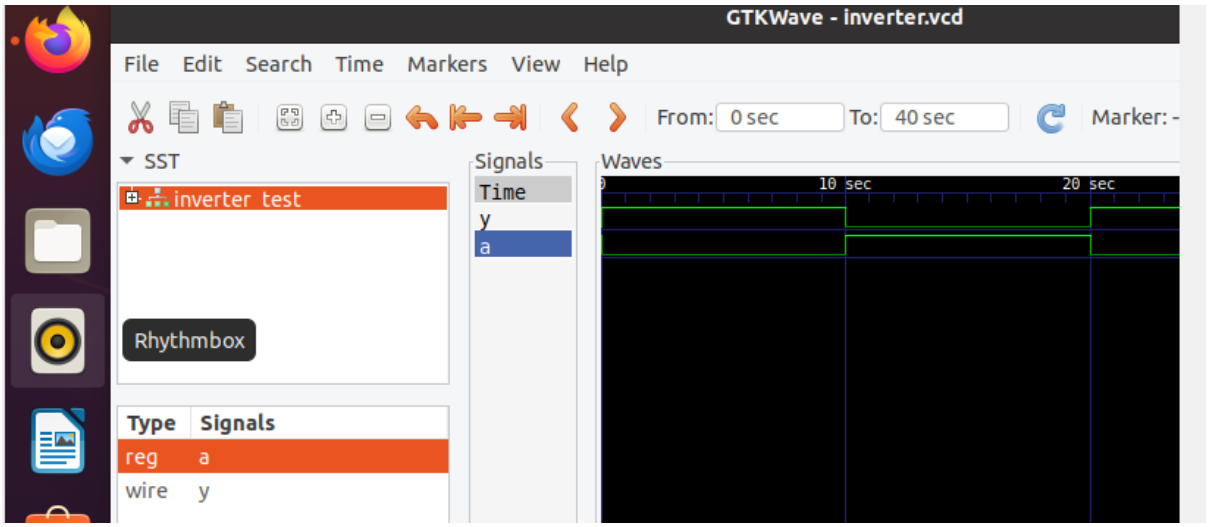


View the Waveforms

Open the VCD file with GTKWave to visualize input and output signal changes:

```
gtkwave inverter.vcd
```

- Use GTKWave's GUI to inspect signal transitions and verify they match expected results.



- Designs can span multiple files; compile by listing all files or using a file list.
- The test bench may be named differently for various modules but always *instantiates* the design under test, sets up stimulus, and observes outputs.
- **VCD files** are standard output for digital simulation, supporting post-simulation waveform analysis.
- No actual hardware is produced; simulation strictly checks logical correctness at the RTL level.