Week 1 RISC V Tape-Out Report on Tool Flow

By

Dr M A. Raheem,

Department of Electronics and Communication Engineering

Muffakham Jah College of Engineering and Technology

Banjara Hills, Hyderabad-500 034

Tool Flow and Directory Preparation

```
root@ubuntuvsd:/home/mjcet/Desktop/raheem/week1/day1# ls
vsdflow
root@ubuntuvsd:/home/mjcet/Desktop/raheem/week1/day1# cd vsdflow/
```

```
root@ubuntuvsd:/home/mjcet/Desktop/raheem/week1/day1# ls
vsdflow
root@ubuntuvsd:/home/mjcet/Desktop/raheem/week1/day1# cd vsdflow/
```

```
git clone https://github.com/kunalg123/vsdflow
```

```
root@ubuntuvsd:/home/mjcet/Desktop/raheem/week1/day1/vsdflow# git clone https:/
/github.com/kunalg123/sky130RTLDesignAndSynthesisWorkshop
Cloning into 'sky130RTLDesignAndSynthesisWorkshop'...
remote: Enumerating objects: 417, done.
remote: Counting objects: 100% (69/69), done.
remote: Compressing objects: 100% (52/52), done.
remote: Total 417 (delta 19), reused 47 (delta 12), pack-reused 348 (from 1)
Receiving objects: 100% (417/417), 7.79 MiB | 6.84 MiB/s, done.
Resolving deltas: 100% (242/242), done.
root@ubuntuvsd:/home/mjcet/Desktop/raheem/week1/day1/vsdflow# ls
sky130RTLDesignAndSynthesisWorkshop
```

- Go to the GitHub page for "Sky130 RTL Design and Synthesis Workshop" under kunalg123.
- Copy the repository link.

```
git clone https://github.com/kunalg123/sky130RTLDesignAndSynthesisWorkshop
```

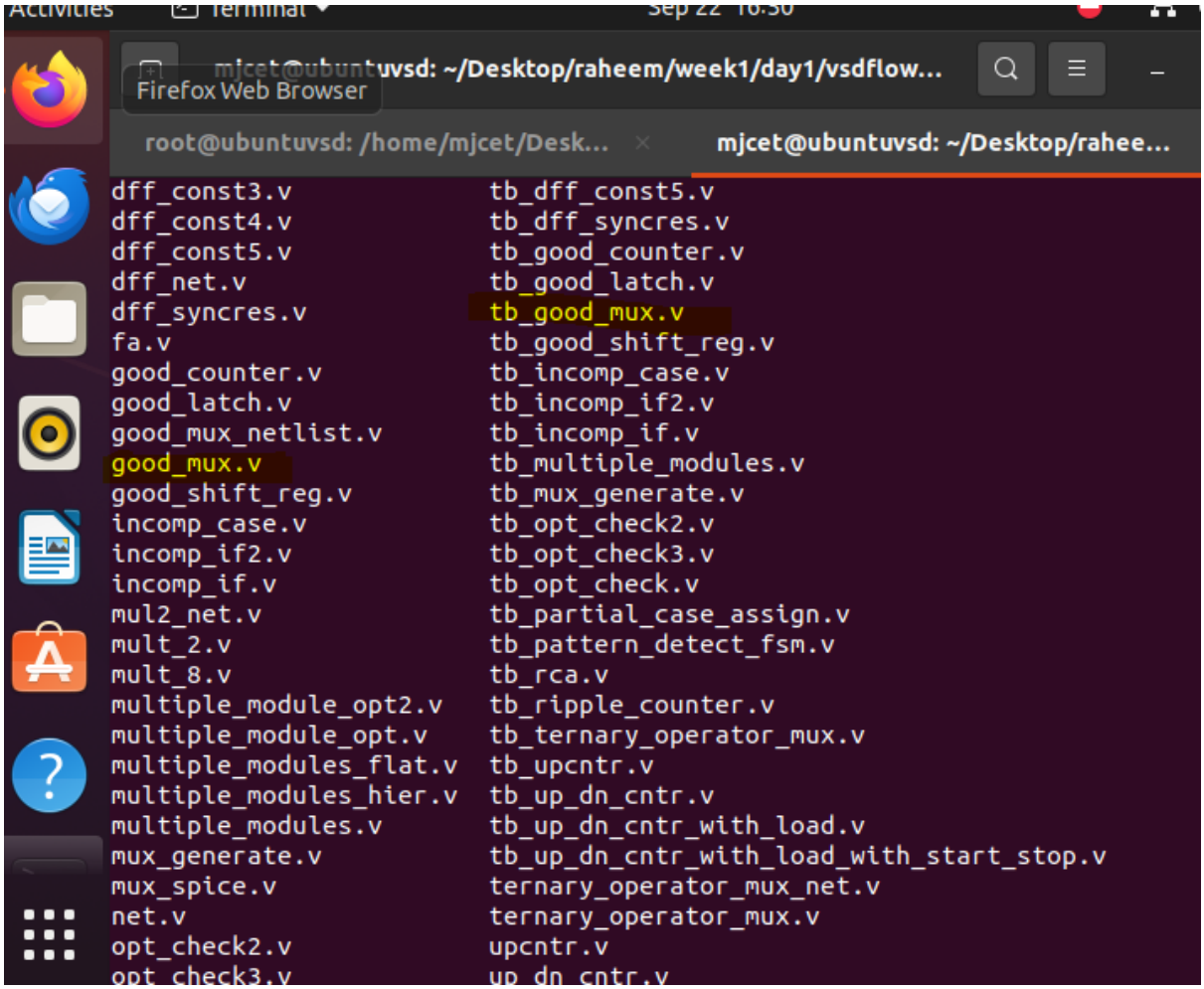
- This creates a `Sky130 RTL Design and Synthesis Workshop` directory for lab exercises.

```
nAndSynthesisWorkshop/
root@ubuntuvsd:/home/mjcet/Desktop/raheem/week1/day1/vsdflow/sky130RTLDesignAnd
SynthesisWorkshop# ls
DC_WORKSHOP  lib  my_lib  README.md  verilog_files  yosys_run.sh
root@ubuntuvsd:/home/mjcet/Desktop/raheem/week1/day1/vsdflow/sky130RTLDesignAnd
SynthesisWorkshop#
```

`mylib`: Contains library files needed for synthesis

- `lib`: Sky130 standard cell library files
- `verilog_model`: Verilog models of the standard cells

`Verilog_files`: Contains all lab-related Verilog source and test bench files



Understanding File Structure

- Each design file (e.g., `good_mux.v`) has a corresponding test bench file (e.g., `tb_good_mux.v`)
- Test bench files are named with `tb_` prefix matching the design's name
- All lab exercises are run inside the `Verilog_files` directory

Compiling and Simulating a Verilog Design

Navigate to the Verilog Files Directory

```
cd Verilog_files
```

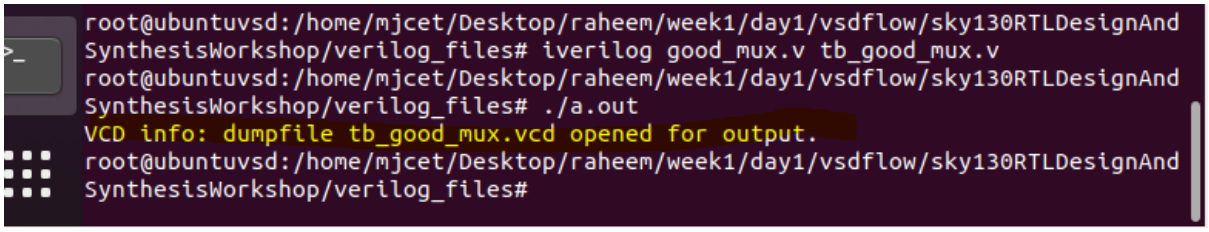
Compile with IVerilog

```
iverilog good_mux.v tb_good_mux.v
```

- Compiling both the design and test bench creates an executable file named `a.out`.

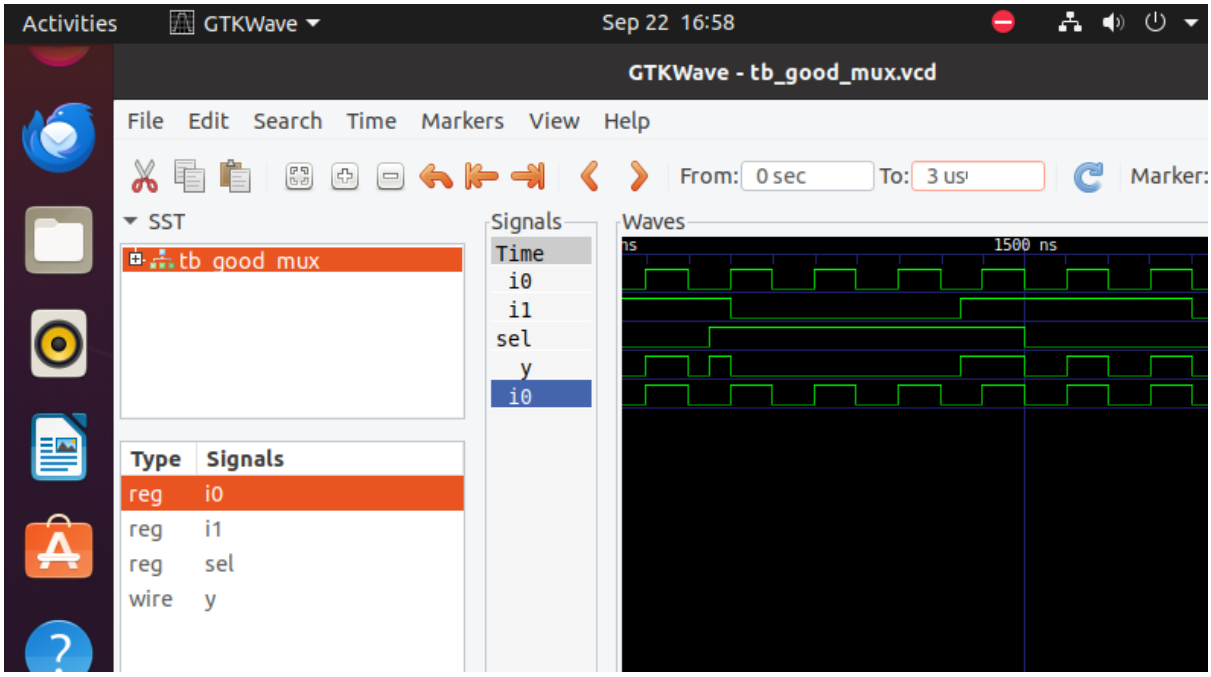
Run the Simulation

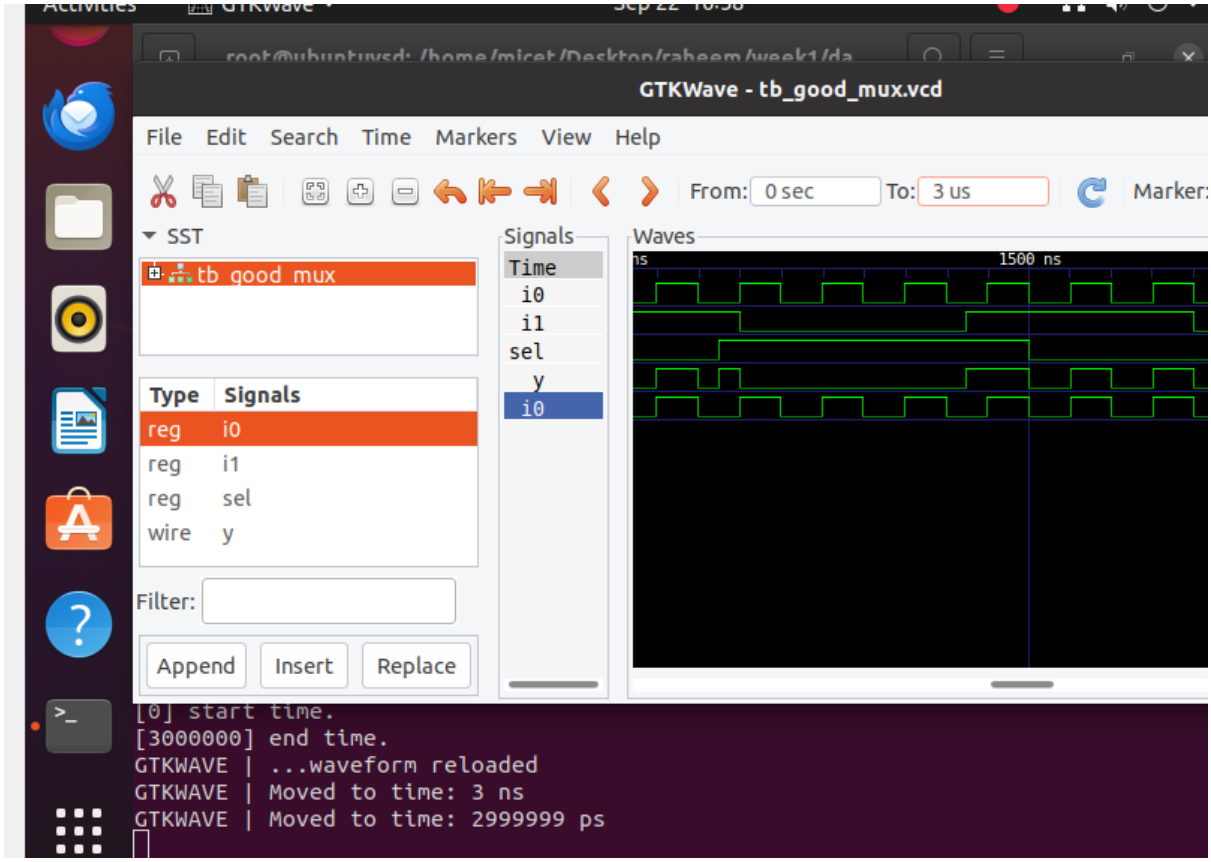
```
./a.out
```

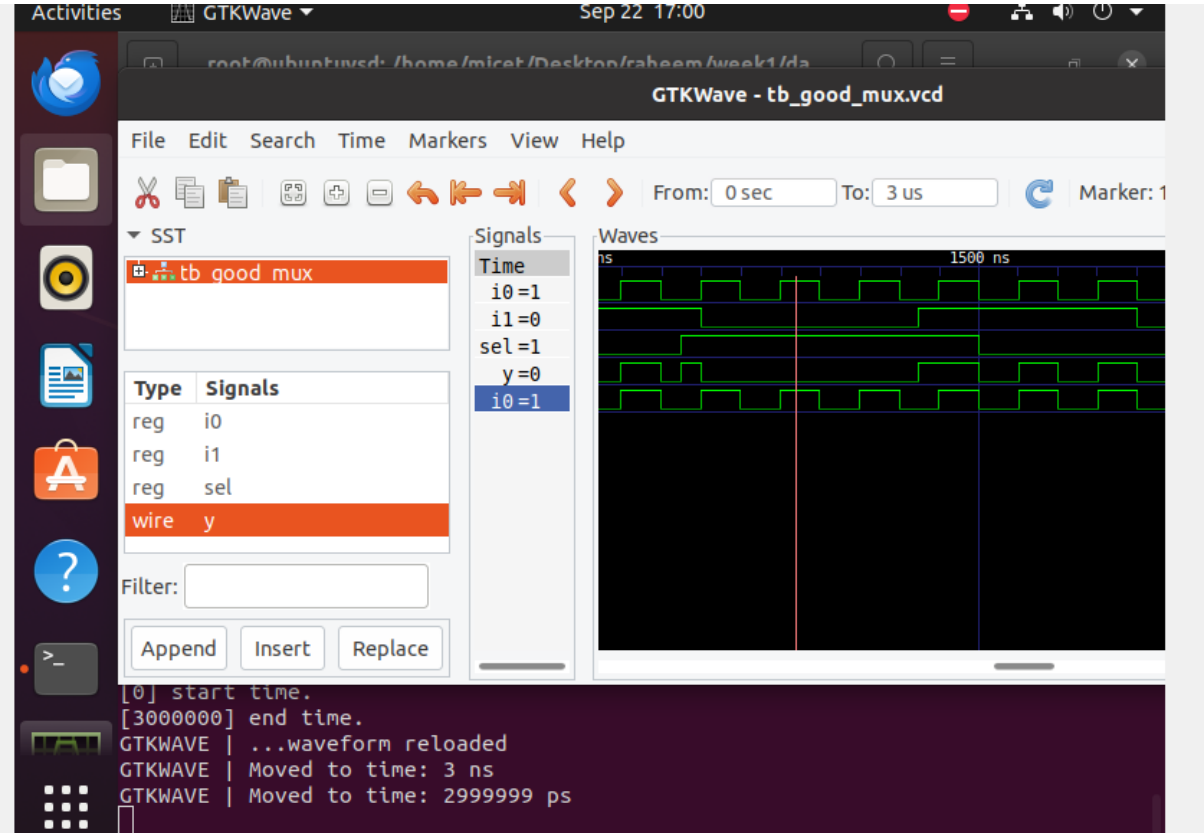


Launch GTKWave to View Waveforms

```
gtkwave tb_good_mux.vcd
```

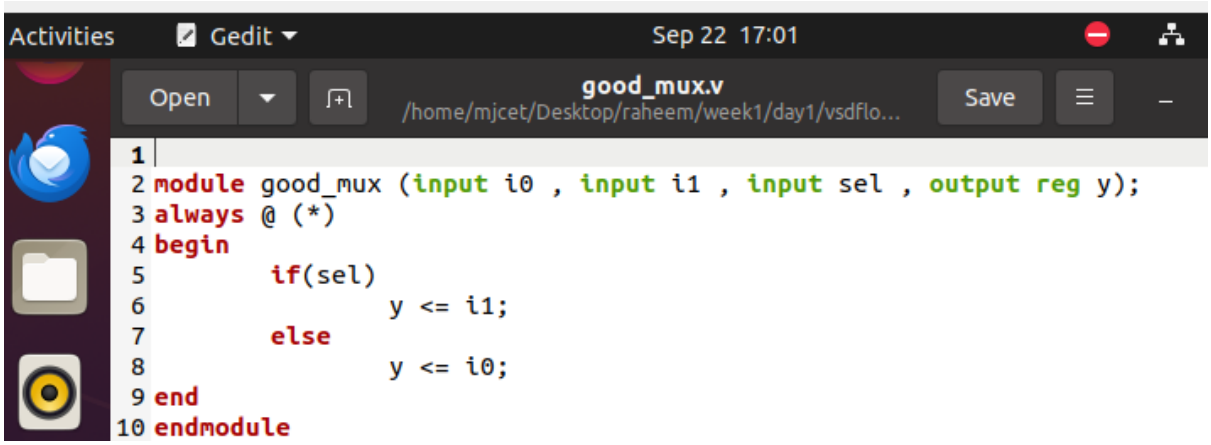

Activities
GTKWave
Sep 22 16:58
GTKWave - tb_good_mux.vcd
File Edit Search Time Markers View Help
From: 0 sec
To: 3 us
Marker:
SST
tb_good_mux
Type Signals
reg i0
reg i1
reg sel
wire y
Signals
Time
i0
i1
sel
y
i0
Waves
1500 ns


GTKWave - tb_good_mux.vcd
File Edit Search Time Markers View Help
From: 0 sec
To: 3 us
Marker:
SST
tb_good_mux
Type Signals
reg i0
reg i1
reg sel
wire y
Filter:
Append Insert Replace
Signals
Time
i0
i1
sel
y
i0
Waves
1500 ns
[0] start time.
[3000000] end time.
GTKWAVE | ...waveform reloaded
GTKWAVE | Moved to time: 3 ns
GTKWAVE | Moved to time: 2999999 ps



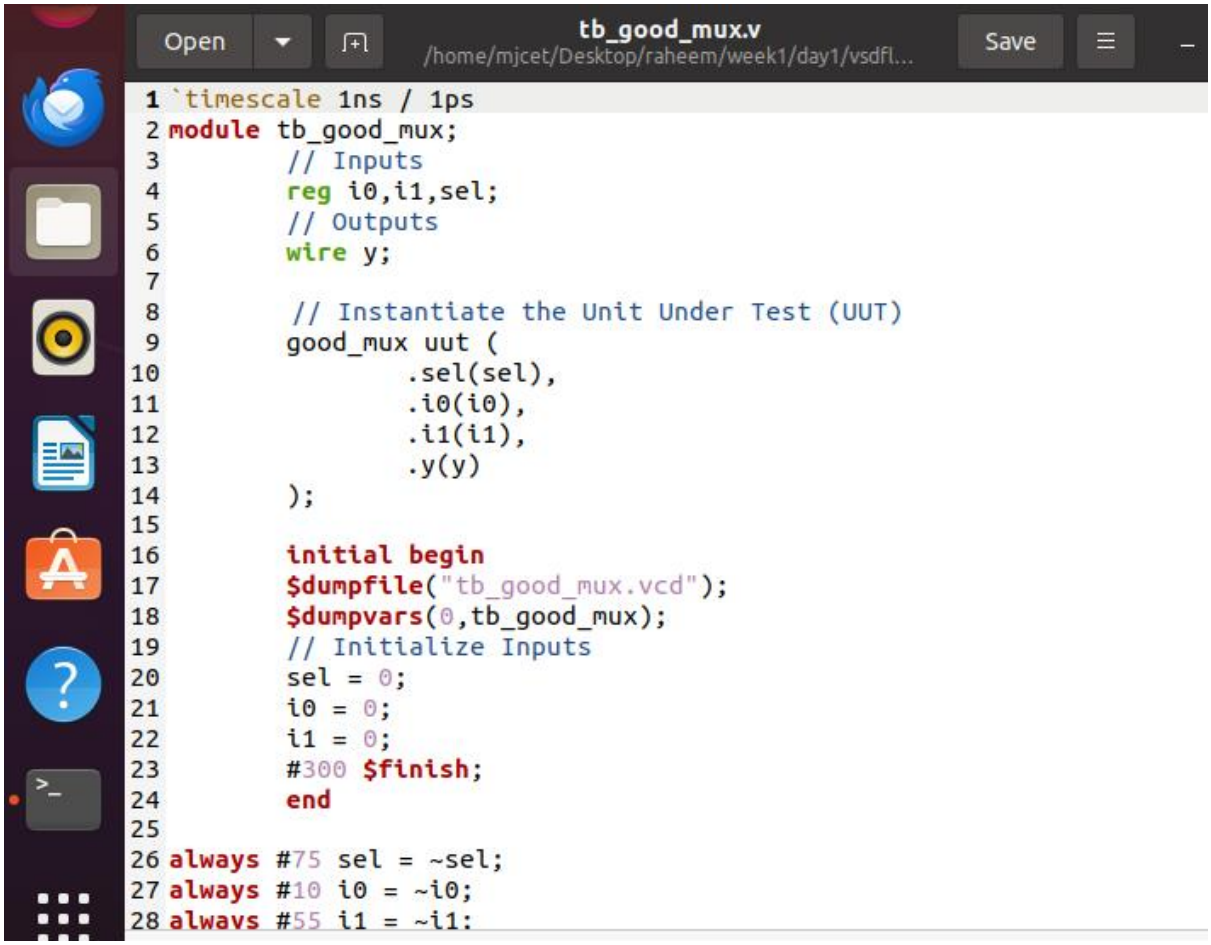
Design and Test Bench File Overview

Good_mux.v and tb_good_mux.v




```
module good_mux (input i0 , input i1 , input sel , output reg y);
always @ (*)
begin
        if(sel)
```

```
                y <= i1;
        else
                y <= i0;
end
endmodule
```




```
`timescale 1ns / 1ps
module tb_good_mux;
        // Inputs
        reg i0,i1,sel;
        // Outputs
        wire y;

       // Instantiate the Unit Under Test (UUT)
        good_mux uut (
                .sel(sel),
                .i0(i0),
                .i1(i1),
                .y(y)
        );

        initial begin
        $dumpfile("tb_good_mux.vcd");
        $dumpvars(0,tb_good_mux);
        // Initialize Inputs
```

```
        sel = 0;
        i0 = 0;
        i1 = 0;
        #300 $finish;
        end

always #75 sel = ~sel;
always #10 i0 = ~i0;
always #55 i1 = ~i1;
endmodule
```