Visvesvaraya National Institute of Technology (VNIT), Nagpur

Digital Hardware Design (ECL313)

Simulation Report

Submitted by :
Rajesh Nagula (BT18ECE059)
Shruti Murarka (BT18ECE099)
Dhruvi Shah (BT18ECE115)
Semester VI

Group No (9)

Submitted to :
Dr. Anamika Singh and Dr. Neha Nawandar
(Course Instructors)
Department of Electronics and Communication Engineering,
VNIT Nagpur

Contents

Problem Statement and Theory

Problem Statement: Design various models for 4-bit Excess-3 to binary converter.

Theory: The excess 3 code is a way of biased coding. In this each digit is expressed as individual binary equivalent + 3. The major advantage is that a decimal can be easily nine's complimented as simple as 1's compliment in binary. Other advantage is is doesn't use 0000 and 1111 which are the signal failure cases. Also it is difficult to write 0000 exactly in magnetic media. There are the following steps to convert the Excess-3 code to binary:

- Get the decimal equivalent of the excess 3 code using simple binary equivalence.
- Subtract 3 in each digit of the decimal number.
- From this newly generated decimal number digits get binary number.

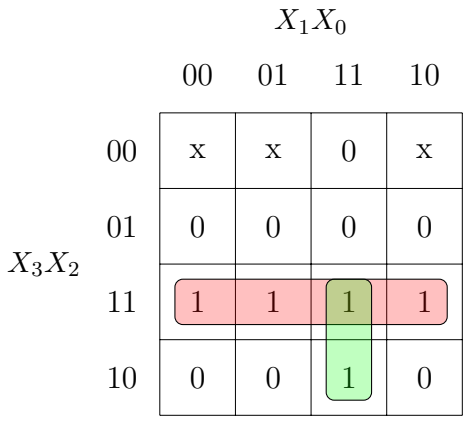
We can also subtract 0011 in each 4-bit excess-3 code to get binary equivalence. By following the duality of each step we can get excess 3 code from binary.

Truth Table: Desired Truth table is given below:

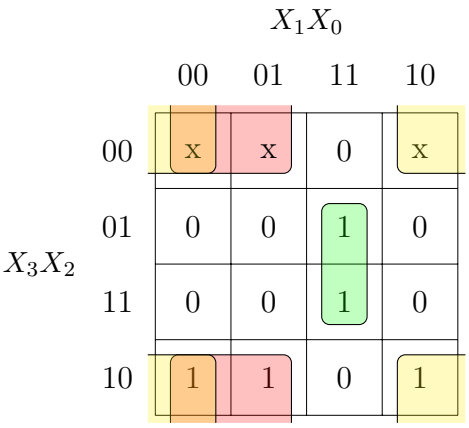
Input ($x_3x_2x_1x_0$)	Output ($b_3b_2b_1b_0$)	Valid Bit (v)
0000	xxxx	1
0001	xxxx	1
0010	xxxx	1
0011	0000	0
0100	0001	0
0101	0010	0
0110	0011	0
0111	0100	0
1000	0101	0
1001	0110	0
1010	0111	0
1011	1000	0
1100	1001	0
1101	1010	0
1110	1011	0
1111	1100	0

K-Maps: K-Maps for output are given below:

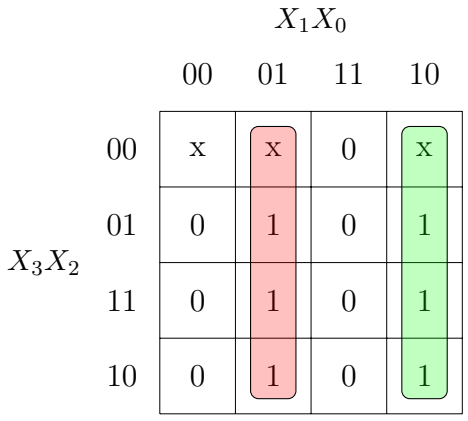
b_3



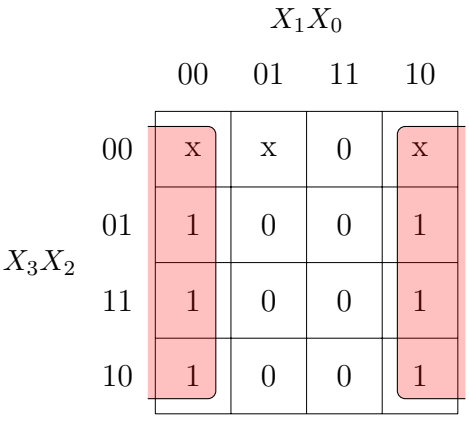
b_2



b_1



b_0



$$b_3 = x_3x_2 + x_0x_1x_3$$

$$b_2 = \overline{x_2}.\overline{x_1} + \overline{x_0}.\overline{x_2} + x_2x_1x_0$$

$$b_1 = \overline{x_1}x_0 + x_1\overline{x_0}$$

$$b_0 = \overline{x_0}$$

K-Map for valid bit is given below:

X_3X_2 \ X_1X_0	00	01	11	10
00	1	1	0	1
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

$v = \overline{x_2}.\overline{x_3}.\overline{x_1} + \overline{x_2}.\overline{x_3}.\overline{x_0}$

- Here x is used for don't care.
- The Karnaugh map is used to reduce the equation in terms of minterms, that is in sum of product form.

And final relations are:

- $b_3 = x_3x_2 + x_0x_1x_3$
- $b_2 = \overline{x_2}.\overline{x_1} + \overline{x_0}.\overline{x_2} + x_2x_1x_0$
- $b_1 = \overline{x_1}x_0 + x_1\overline{x_0}$
- $b_0 = \overline{x_0}$
- $v = \overline{x_2}.\overline{x_3}.\overline{x_1} + \overline{x_2}.\overline{x_3}.\overline{x_0}$

Simulation Screen Shots of Code - 1

Quartus Code and RTL View:

- Open Quartus and click on Create New Project.

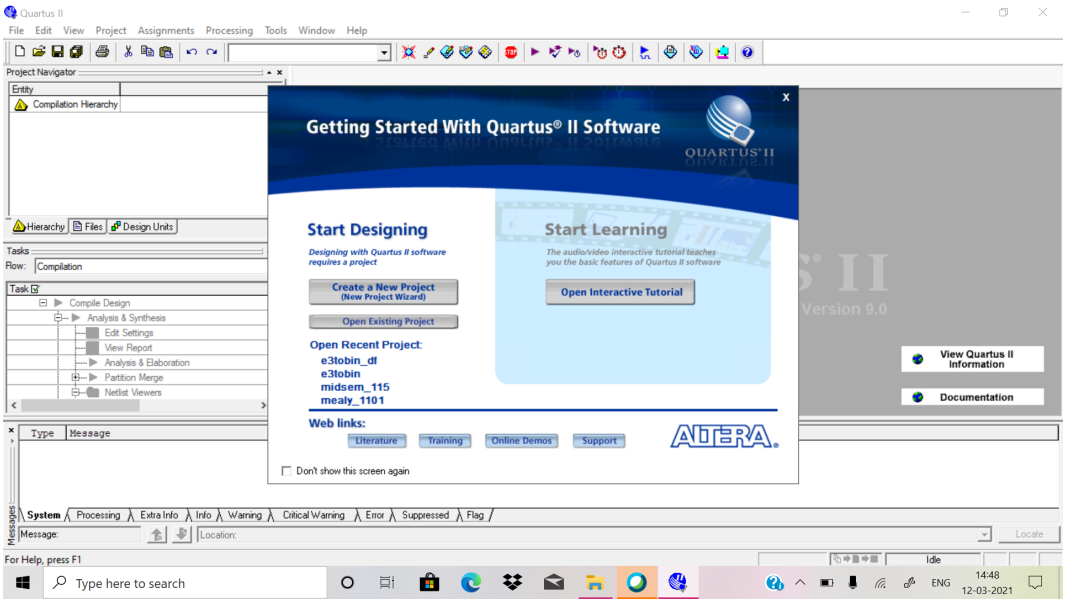


Figure 1: Create project window.

- Click Next till you get working directory window.

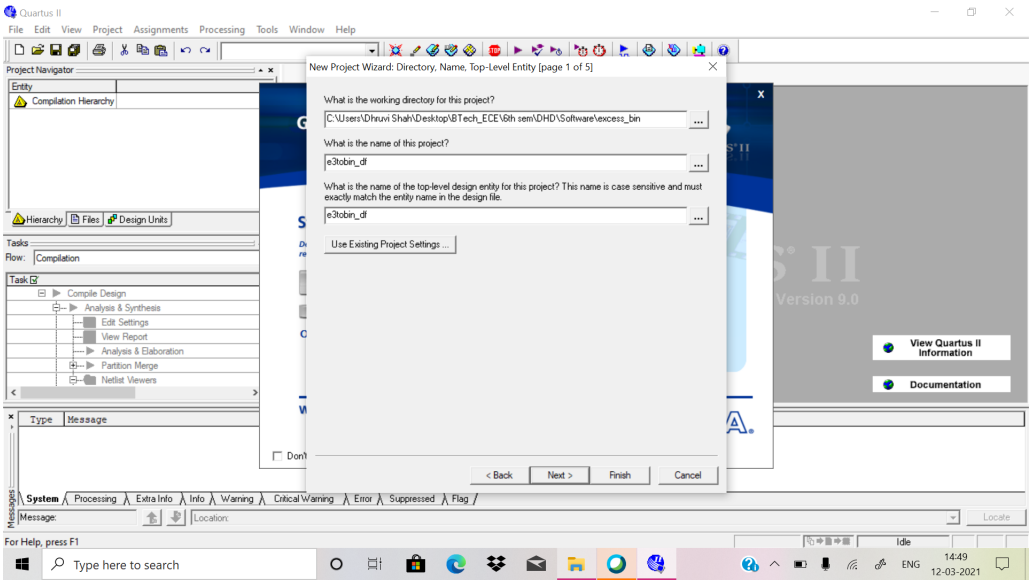


Figure 2: Working Directory window.

Now browse your working directory and name your project (this nae will be your entity name as well).

- Click next till finish window and finish the creating project procedure.
  Now open new file by clicking FILE –> NEW or white page icon on top left. Once a new VHDL file has been created write your code.

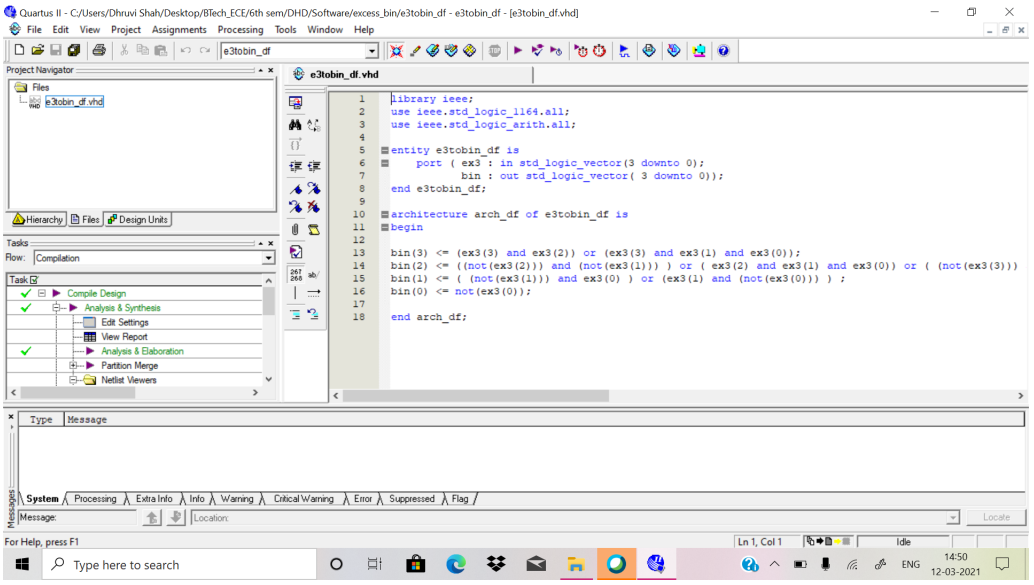


Figure 3: New file.

- Save the file (Press Ctrl+S, it will automatically provide entity name for your file click Save.), then
  Select processing –> start compilation.

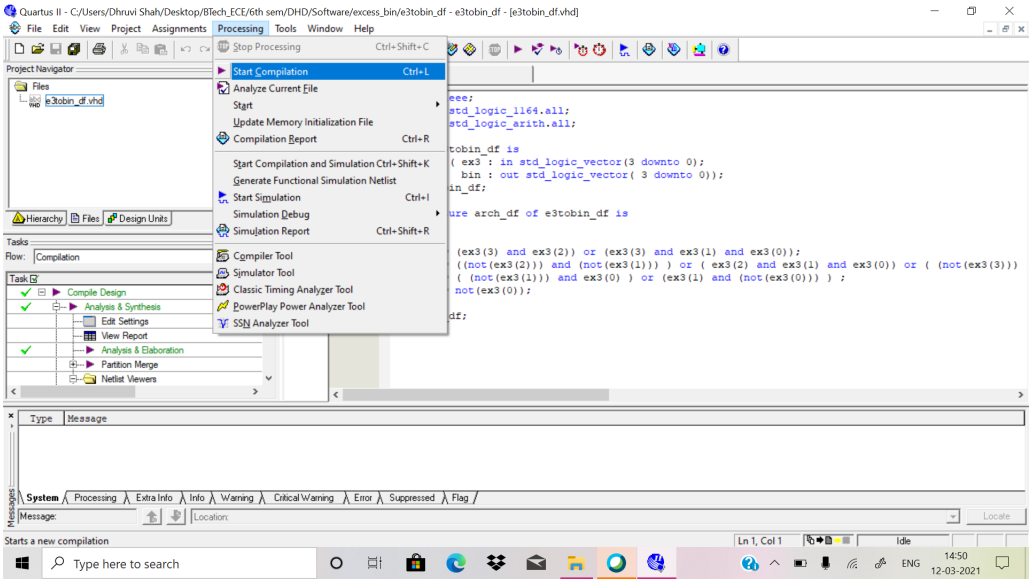


Figure 4: Compilation File

- Tools –> Netlist viewers –> RTL view.
  This opens a window thats shows the RTL schematic of the written code.

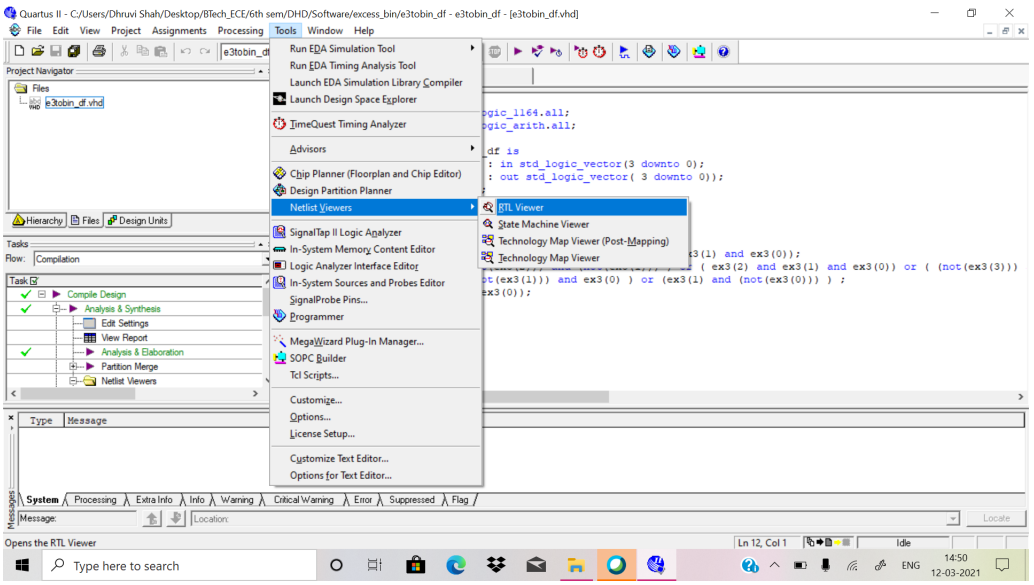


Figure 5: RTL view

- Now open ModelSim to view the waveform.
  Select JumpStart.

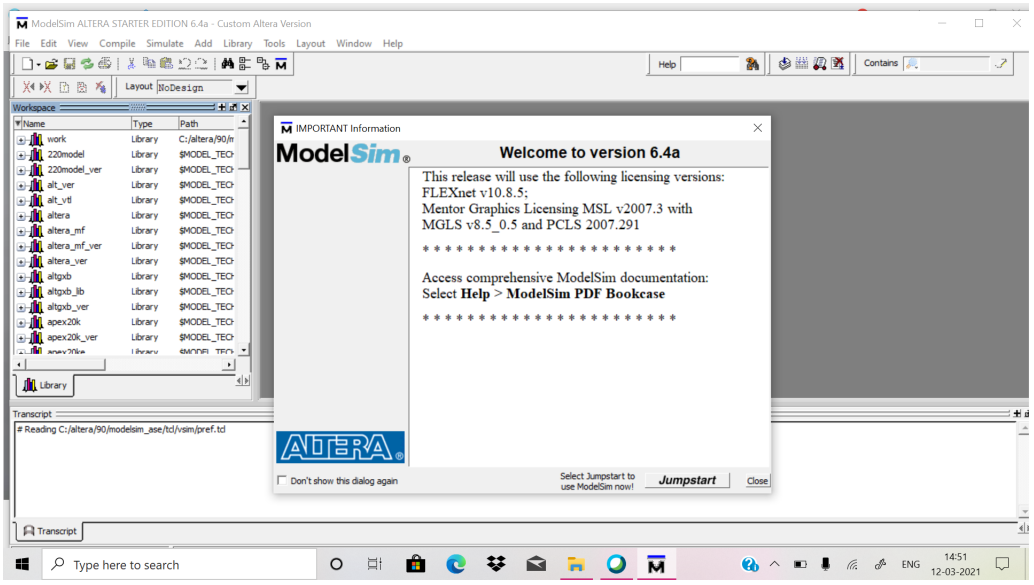


Figure 6: modelsim.

- Create a New project.

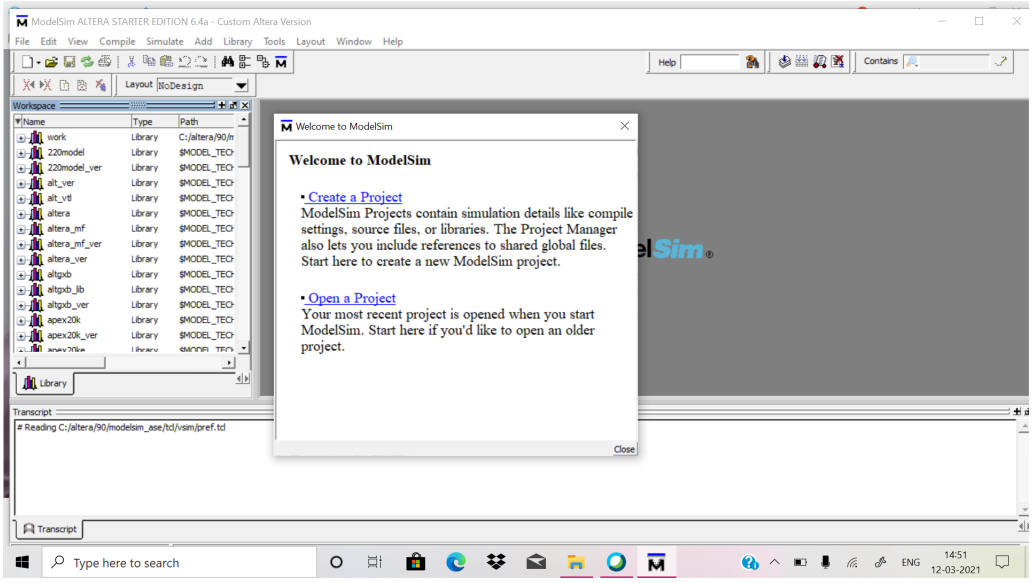


Figure 7: Create a new project.

- Right click in the workspace –> Add the VHDL file already saved previously –> Compile the file

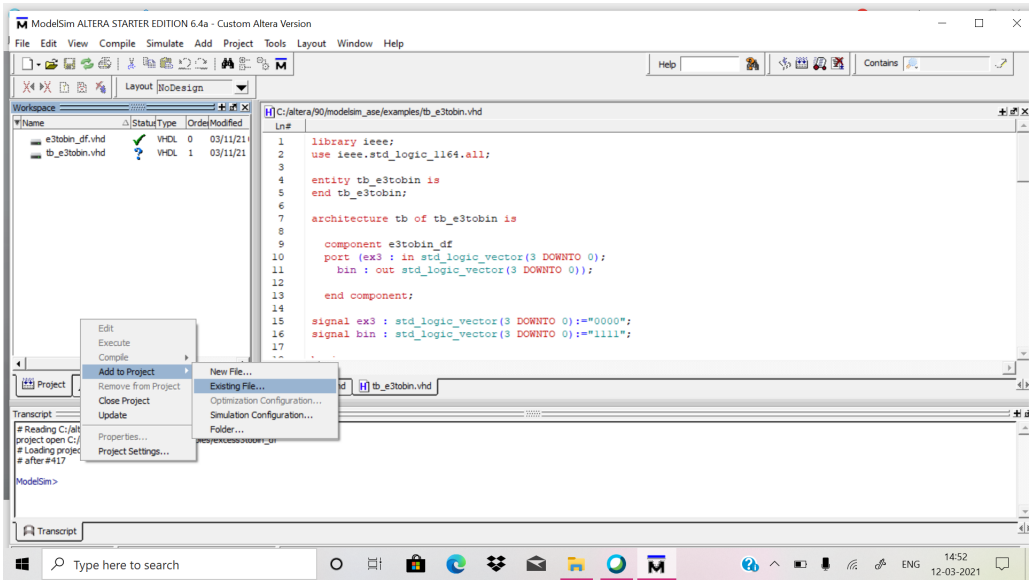


Figure 8: workspace of modelsim.

- Add a new source file (VHDL) for writing the testbench code.

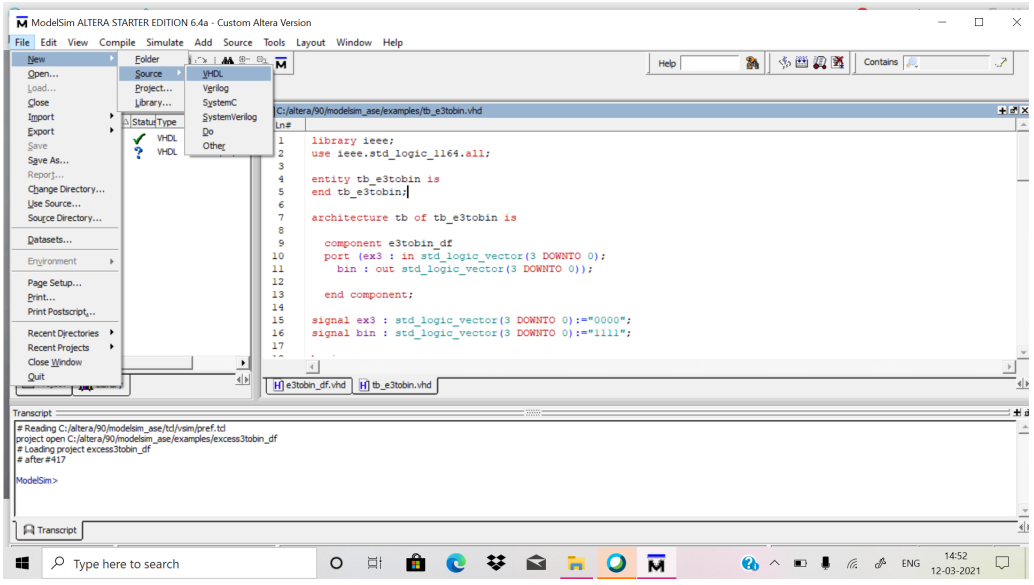


Figure 9: Test bench file.

- Once the code has been written , save it (ctrl + s and compile

the testbench file , make suitable changes in case it gives.

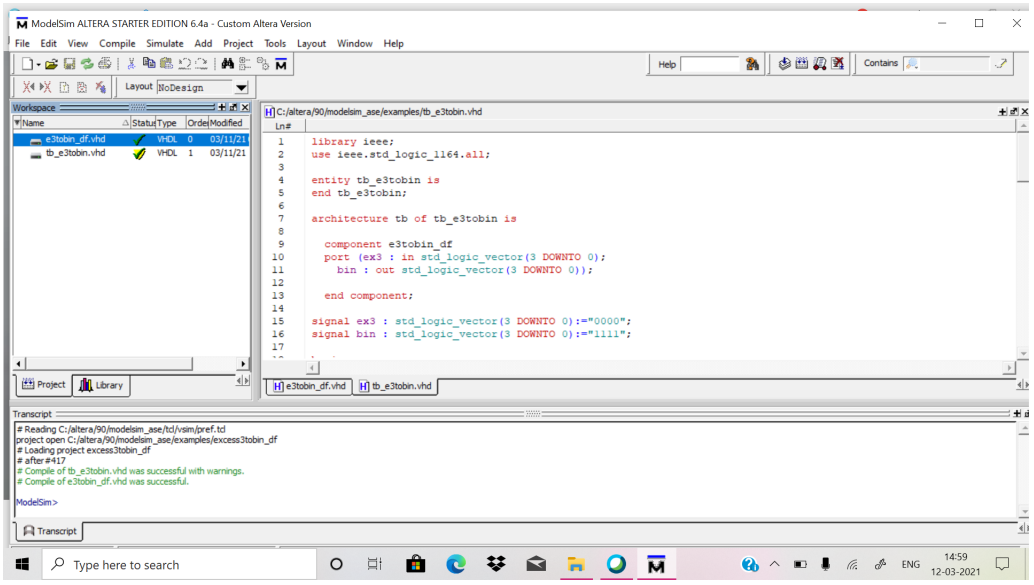


Figure 10: Compiled files .

- Once it has been compiled , select start simulation and select the testbench file just written.

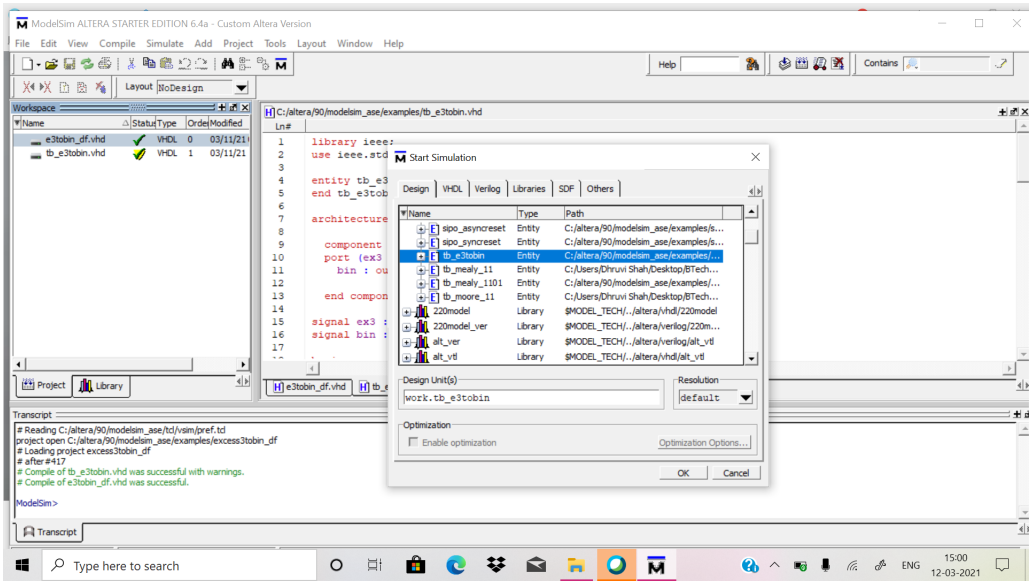


Figure 11: simulation

- The given window will open
  Right click in the blue space –> add –> to wave –> all in region
  Once this is done , a waveform window will appear.

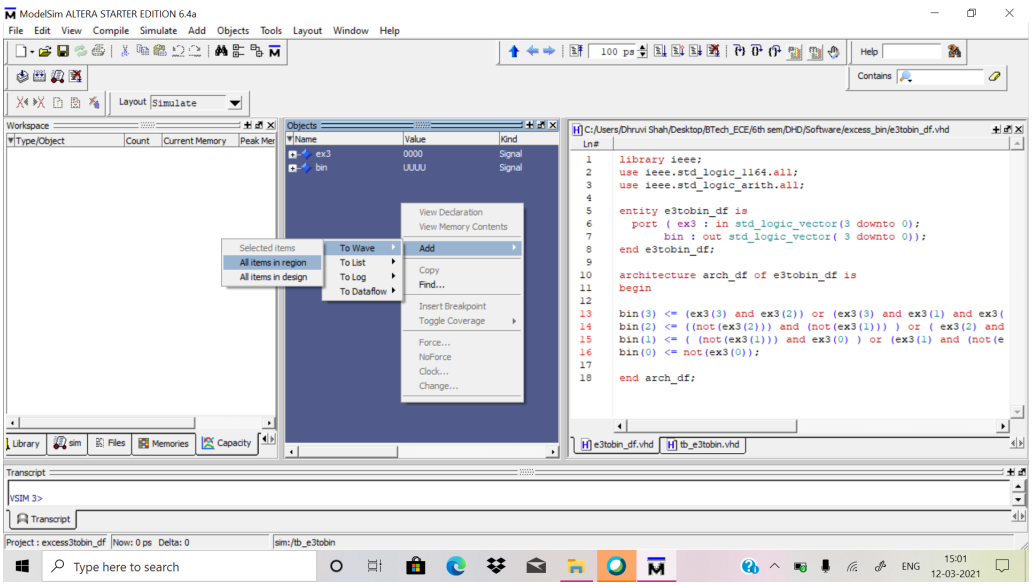


Figure 12: Simulation.

- Click on Run
  the waveform can be seen

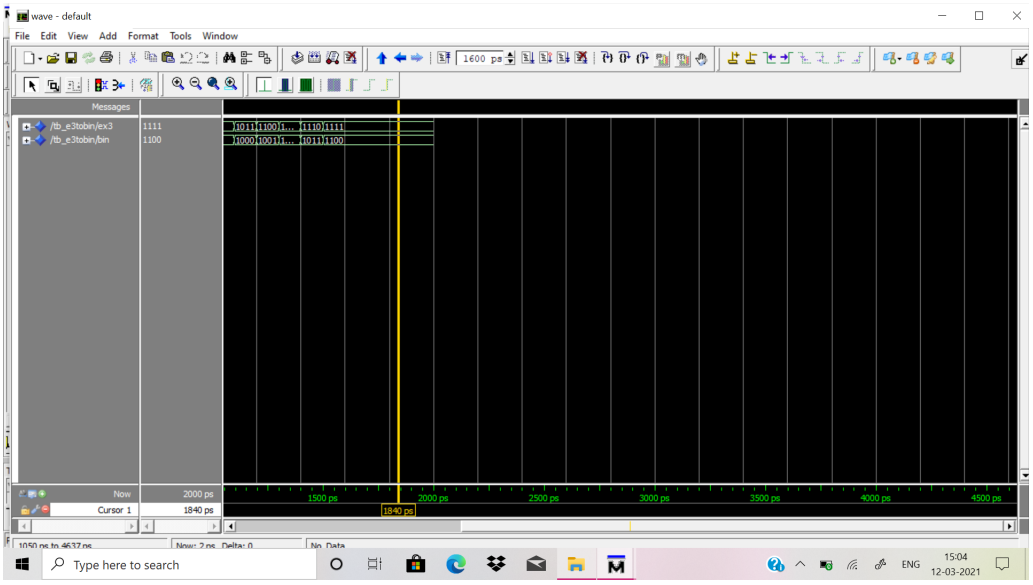


Figure 13: Open Existing File.

Simulation Screen Shots of Code - 2

Quartus Code and RTL View:

- Open Quartus and click on Create New Project.

Figure 14: Create project window.

- Click Next till you get working directory window.

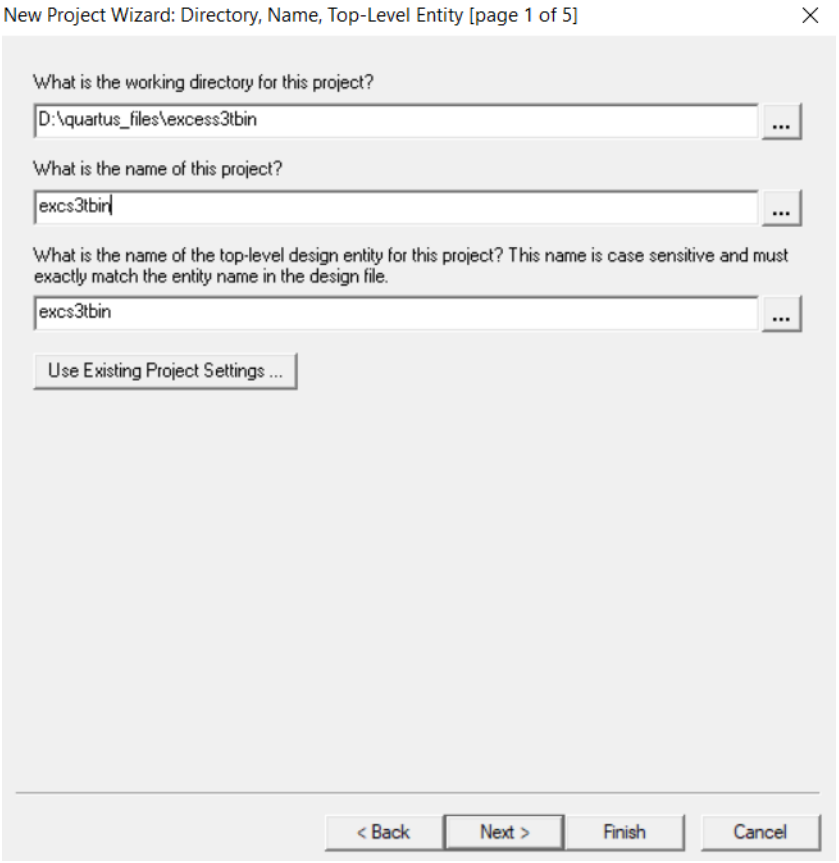


Figure 15: Working Directory window.

Now browse your working directory and name your project (this nae will be your entity name as well).

- Select the option file to create a new file.
  Now open new file by clicking FILE –> NEW or white page icon on top left.

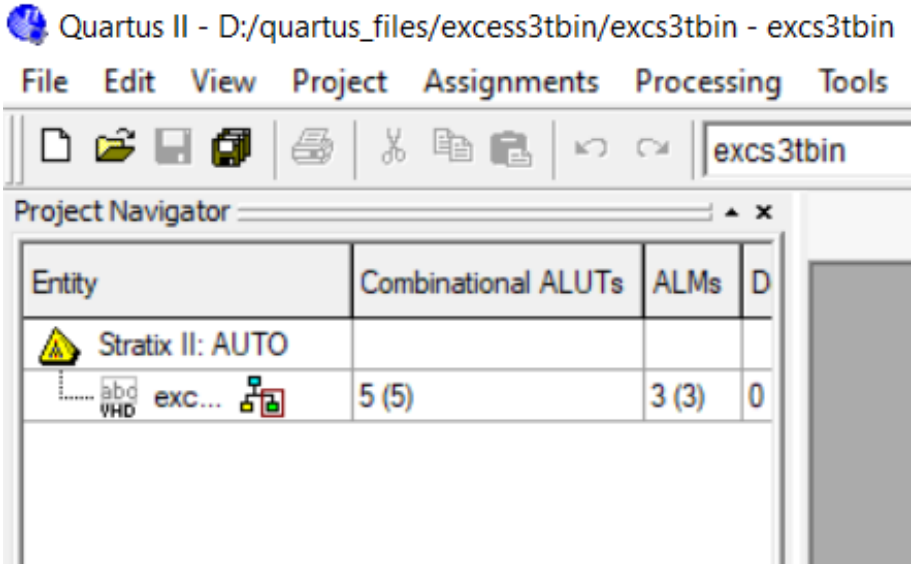


Figure 16: New file.

- Open VHDL file.

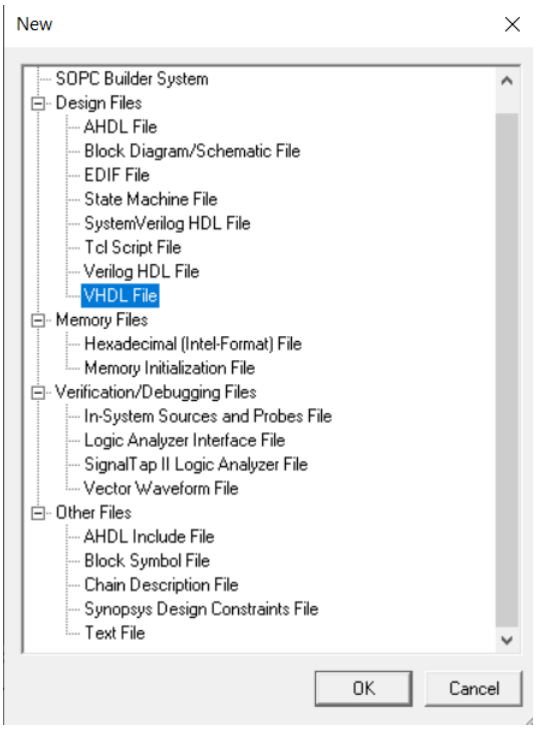


Figure 17: New VHDL file.

- Write Your VHDL Code.

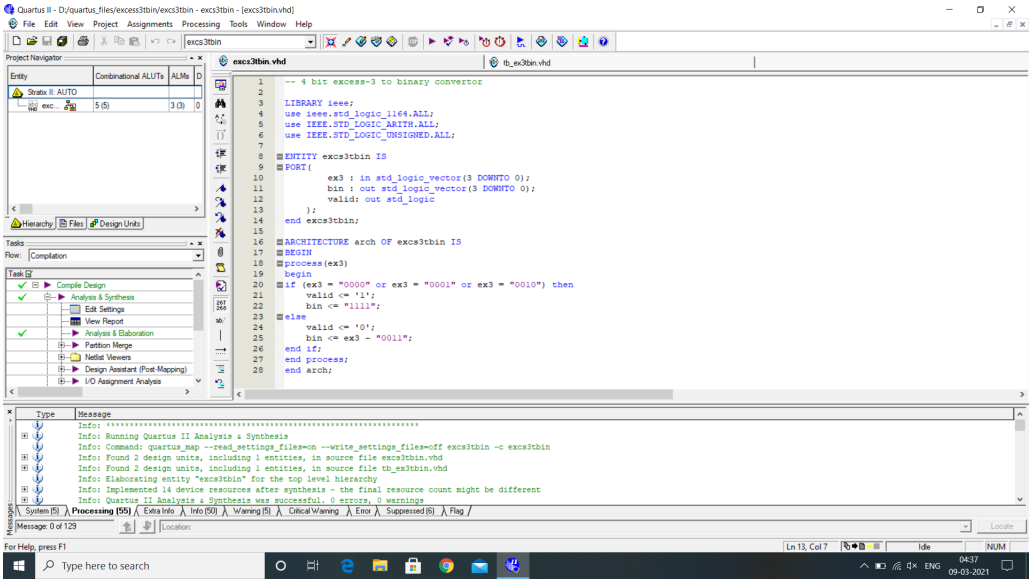


Figure 18: Code in your file.

- Save your file now. Press Ctrl+S, it will automatically provide entity name for your file click Save.

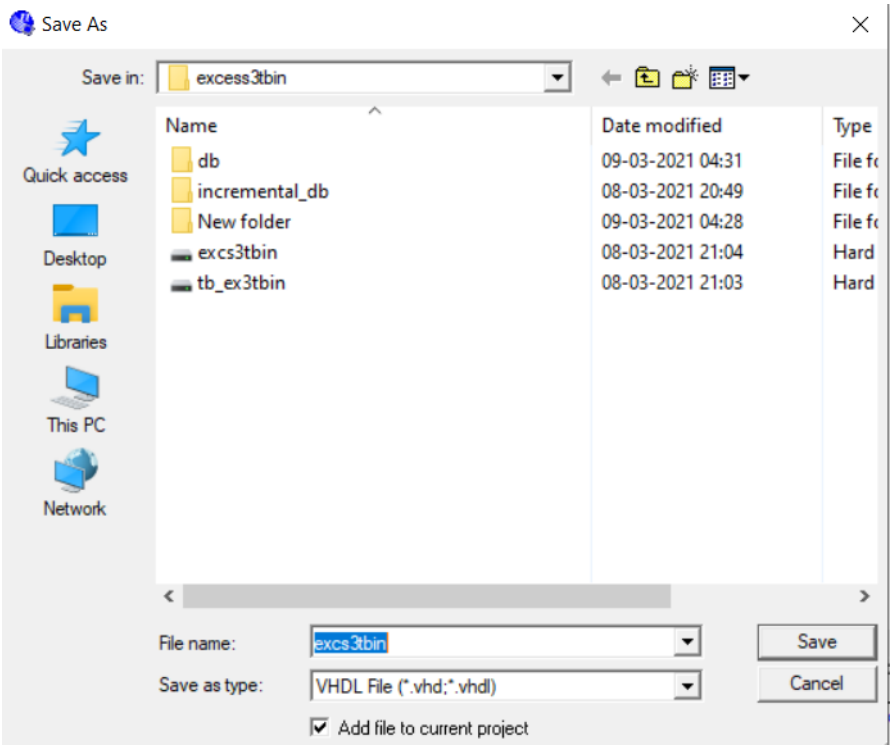


Figure 19: Save your file.

- Time for COMPILATION !! Compile your file by clicking Processing –> Start Compilation.

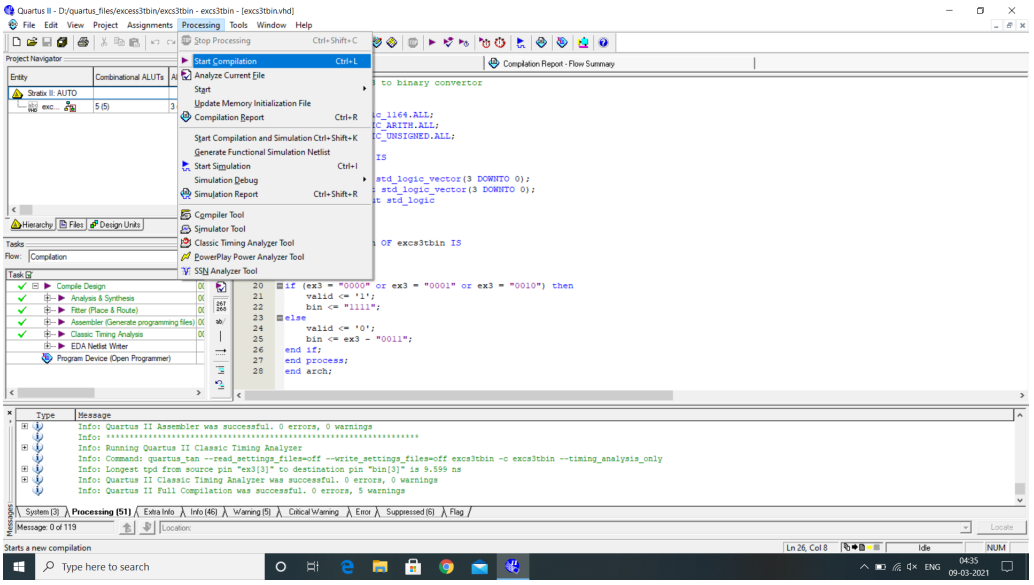


Figure 20: Compile your file.

- Resolve errors if any resave(Ctrl+S) and compile again till you get 0 errors. You can ignore warnings!

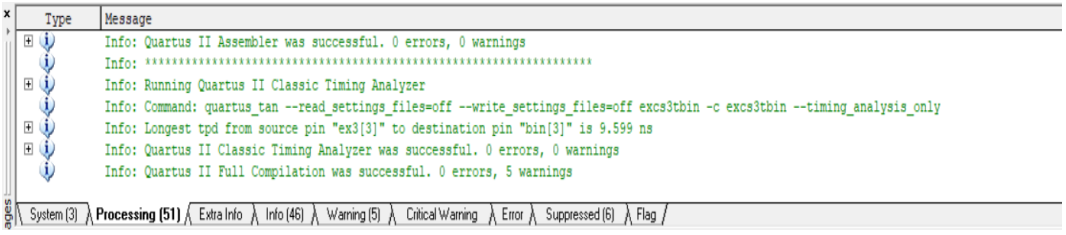


Figure 21: Error-free Compiled file.

- For RTL View Click TOOLS –> Netlist Viewers –> RTL Viewer

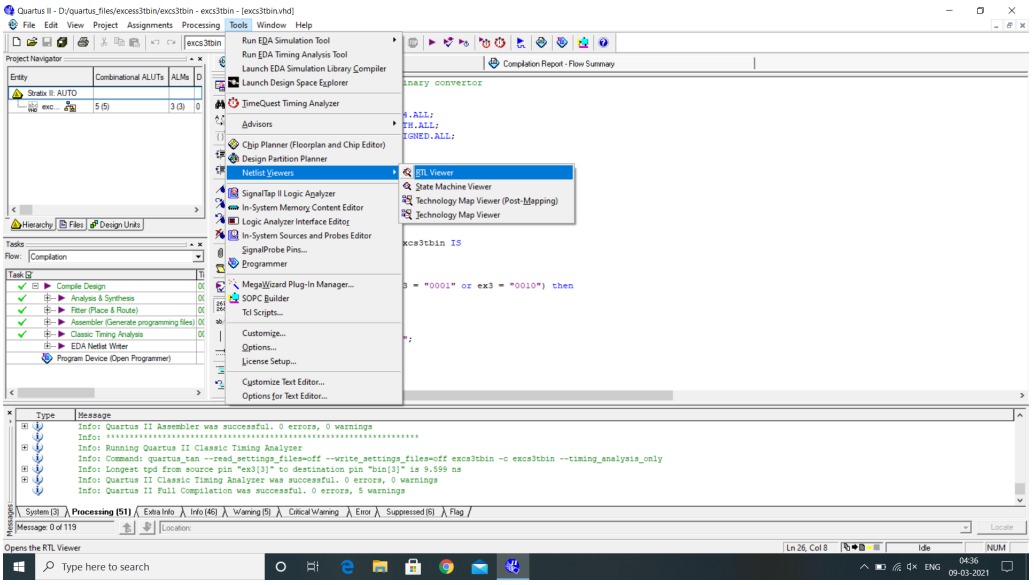


Figure 22: RTL Viewer window.

- RTL View

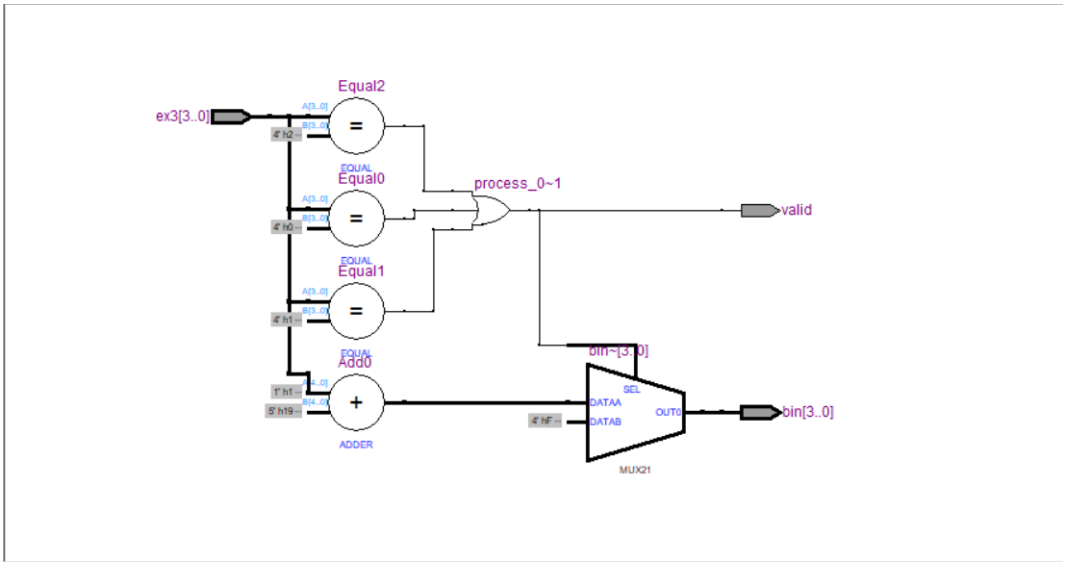


Figure 23: RTL View.

- Now you are good to go !! Create new VHDL file for Testbench, code it and save it with your testbench entity name. Compile till

you get 0 errors. We will be using this file in Modelsim Simulation.

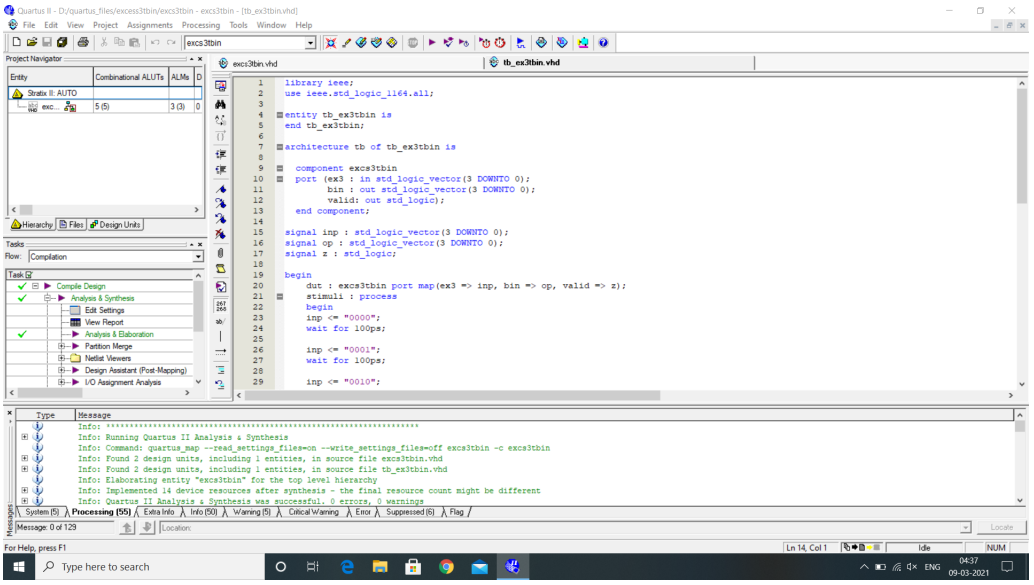


Figure 24: Testbench VHDL file.

ModelSim Simulation:

- Open Modelsim and click JUMPSTART to welcome window. Select Create New Project.

Figure 25: ModelSim create project window.

- Browse Working Directory and name your project preferably entityname-sim.

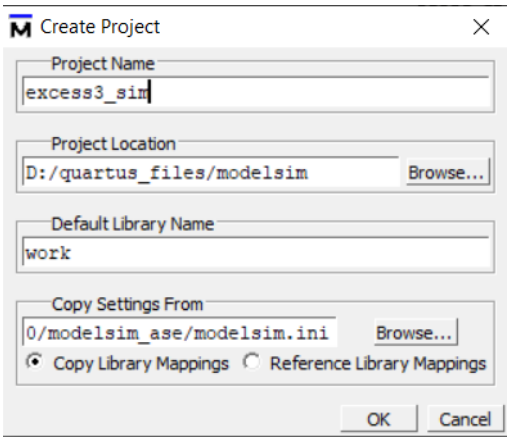


Figure 26: Name project and browse working directory.

- Now select Open a Existing File and browse the VHDL entity

and tb files.

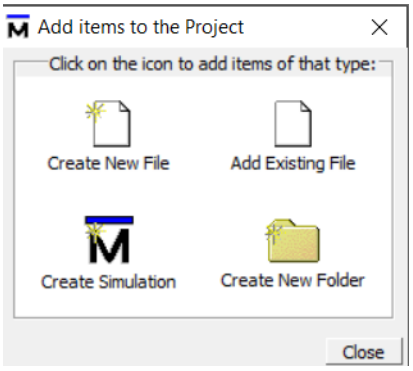


Figure 27: Open Existing File.

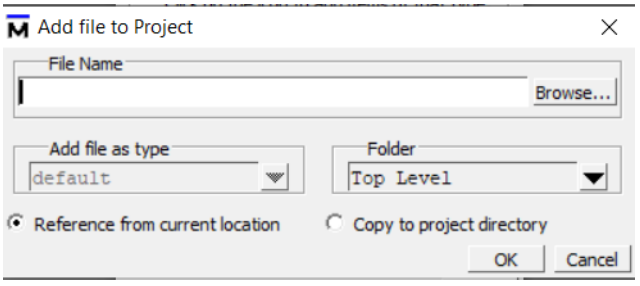


Figure 28: Browse Existing entity and testbench VHDL file.

- Compile files by Right clicking on one of the –> Compile –> Commpile ALL.

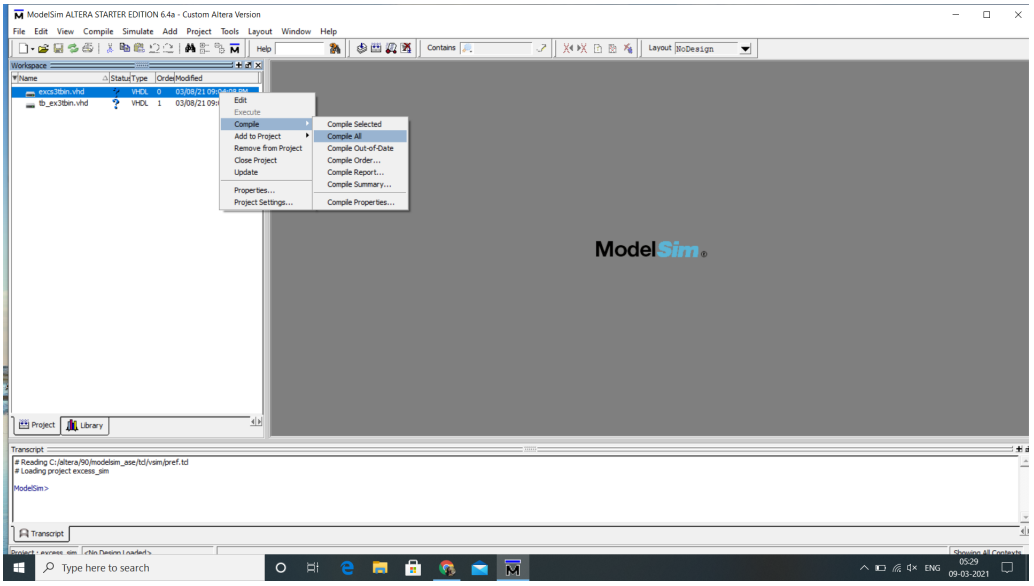


Figure 29: Compile VHDL files.

- For simulation click Simulation –> Start Simulation.

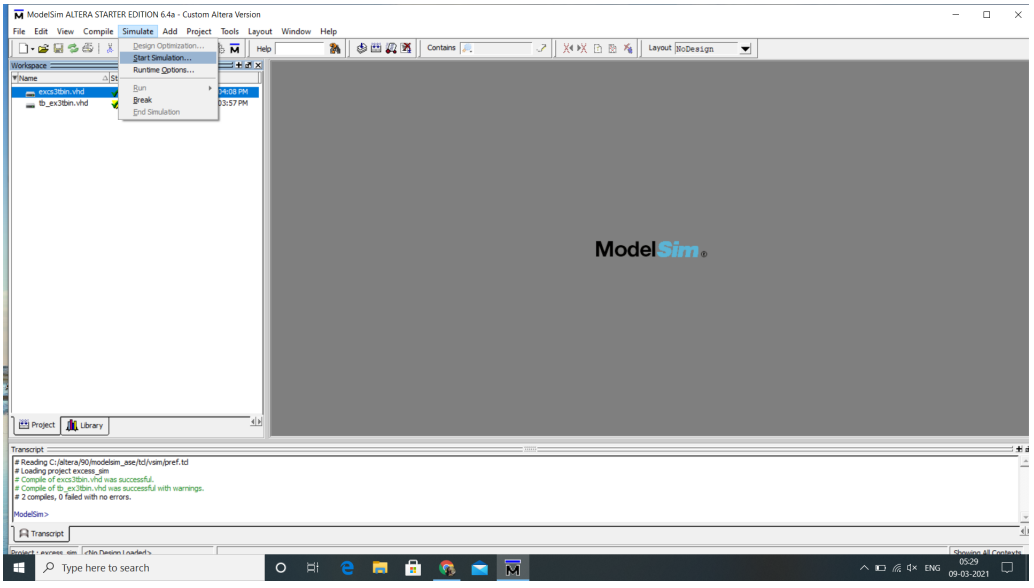


Figure 30: Start Simulation.

- Add tb VHDL file to work by clicking + symbol infront of work and adding tb file then click ok.

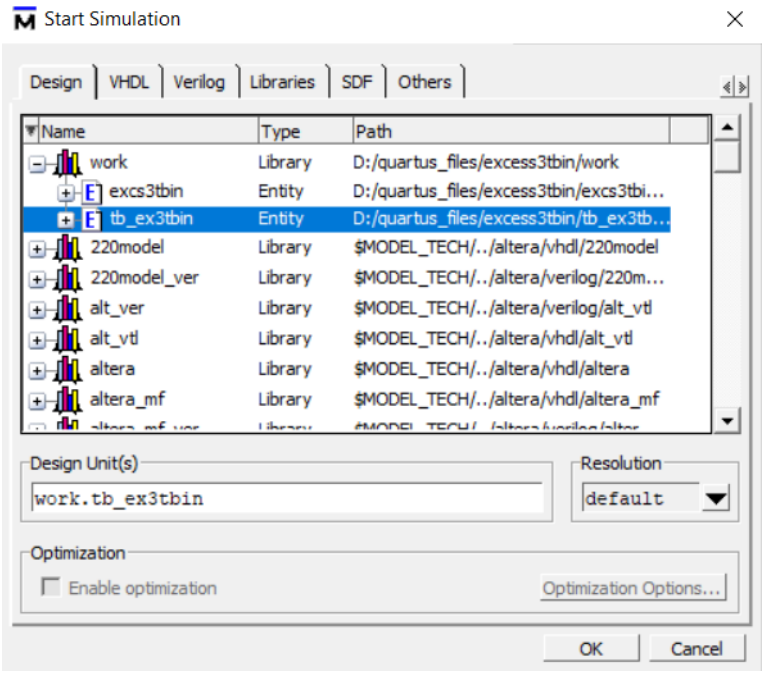


Figure 31: Add Testbench file to work.

- Now add all the object in the wave simulation by right click on object window –> ADD –> To Wave –> All items in the region.

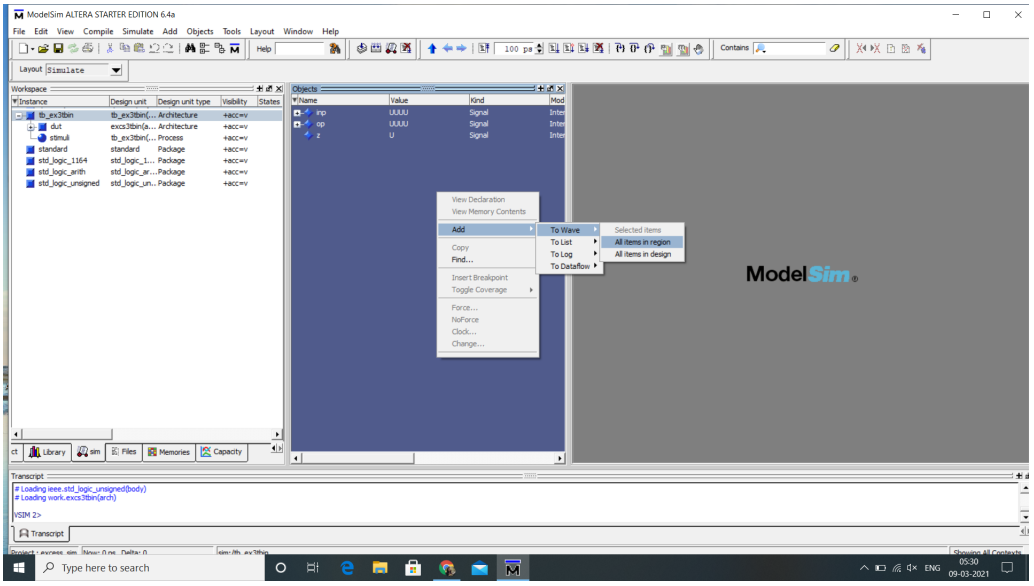


Figure 32: Add all objects to wave siulation window.

- Time to run the simulation. Click icon run simulation icon (next to 100ps). This will run simulation for 100ps.

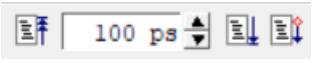


Figure 33: Run simulation.

- Simulation

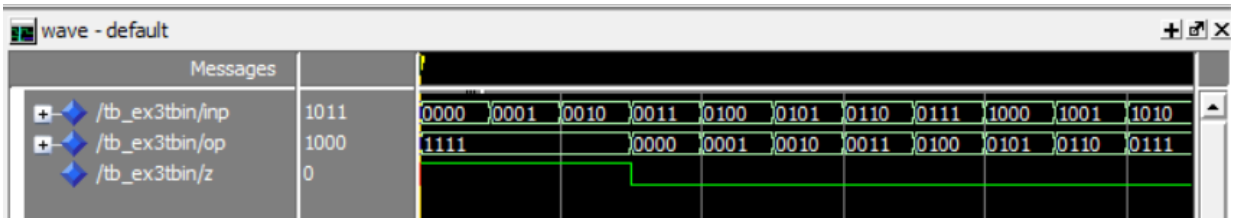


Figure 34: Simulation.

Simulation Screen Shots of Code - 3

Quartus Code and RTL View:

- Open Quartus

```
rajesh@rajesh-Strix-GL504GM-GL504GM:~/intelFPGA_lite/20.1/quartus/bin$ ./quartus
 &
[2] 24925
[1]   Done                    ./quartus
```

Figure 35: Open quartus from terminal.

- create new project

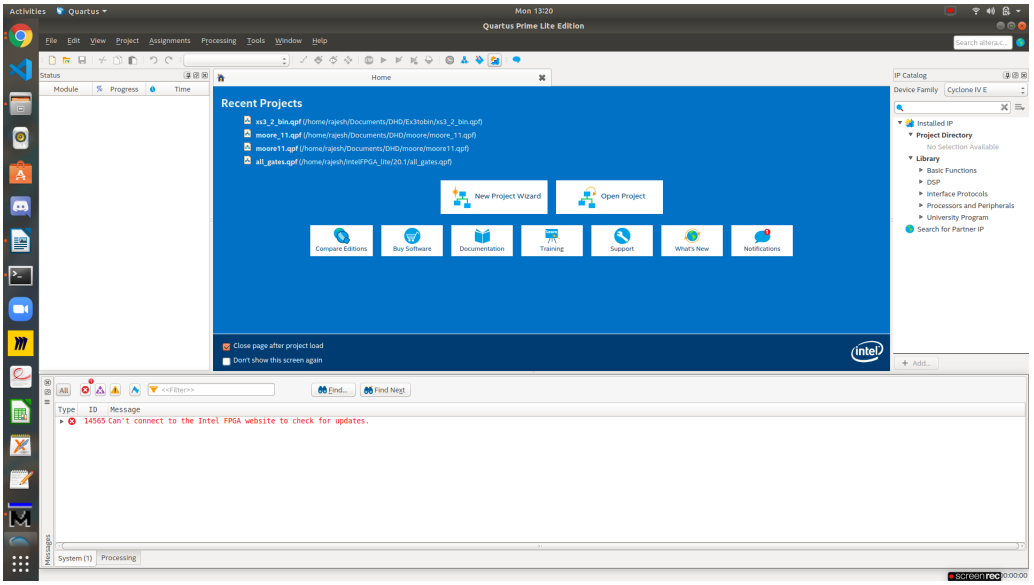


Figure 36: Create new project

- Browse your working directory window and name your project (this will be your entity name as well). Click next till finish window and finish the creating project procedure.

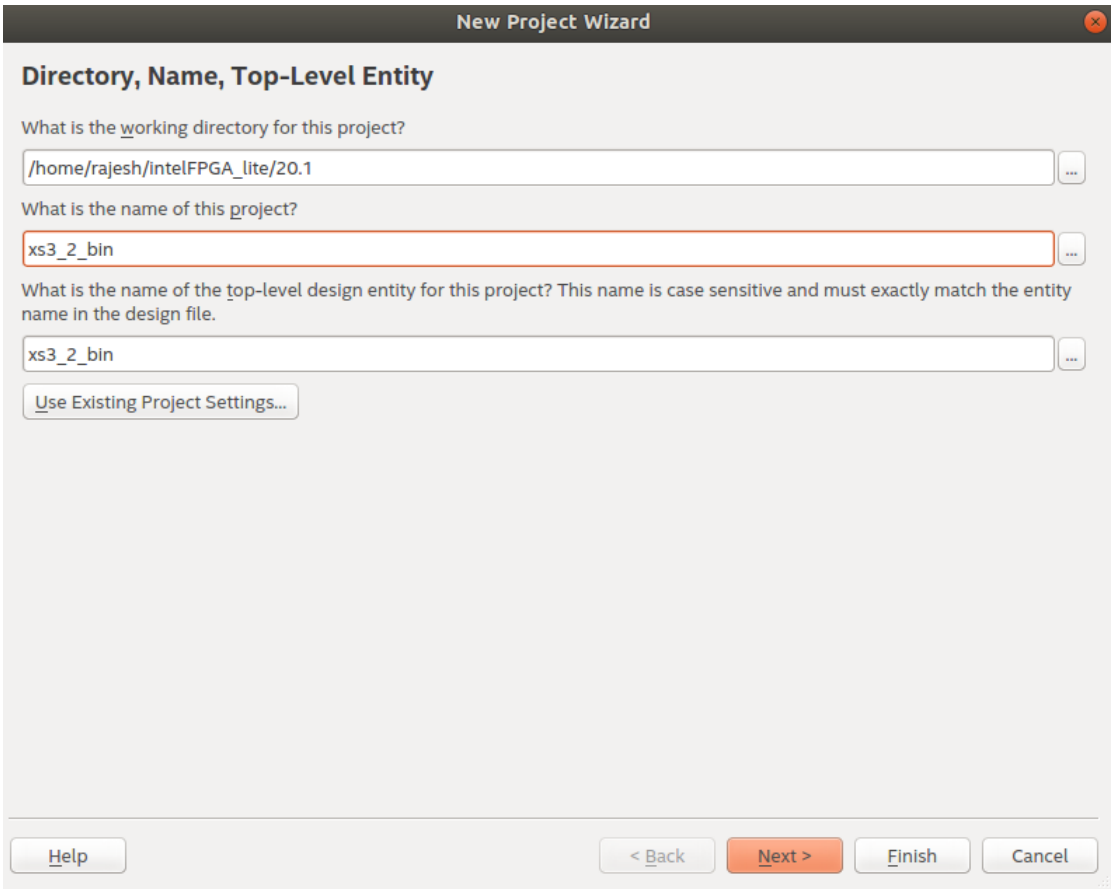


Figure 37: Working Directory window.

- Now open new file by clicking FILE –> NEW or white page icon on top left. Once a new VHDL file has been created write your code.

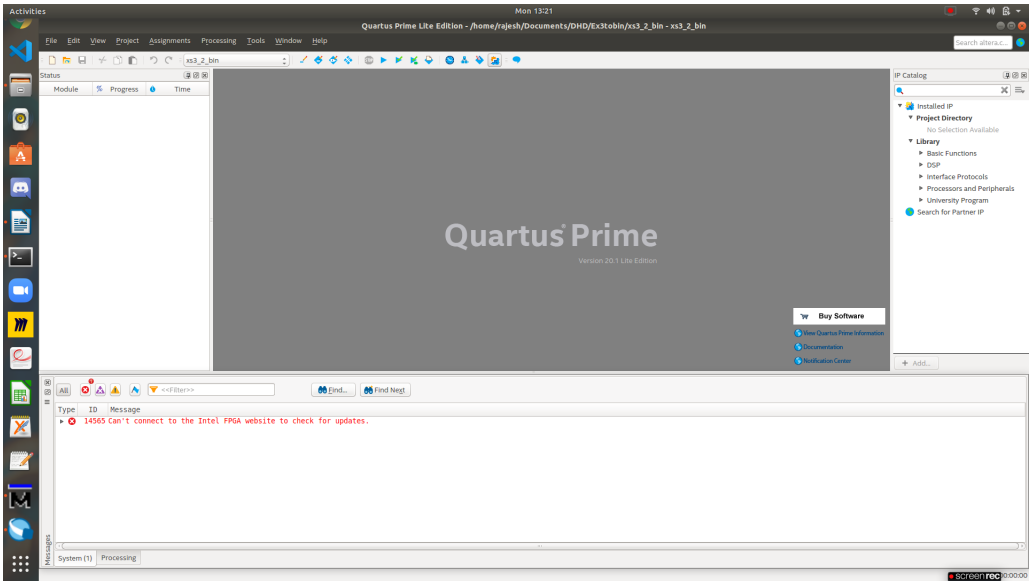


Figure 38: New file.

- Save the file (Press Ctrl+S, it will automatically provide entity name for your file click Save.), then
  Select processing –> start compilation.

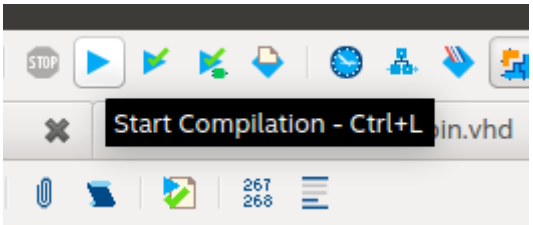


Figure 39: Compilation File

- Wait till all status is 100% if errors pop resolve them.

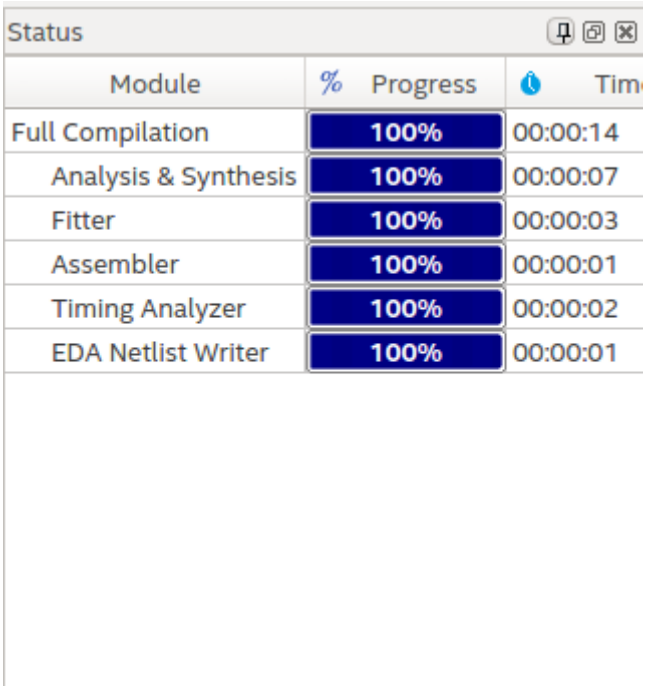


Figure 40: Status

- Now open ModelSim to view the waveform.

.

```
rajesh@rajesh-Strix-GL504GM-GL504GM:~/intelFPGA_lite/20.1/modelsim_ase/bin$ ./vsim &
[1] 26233
```

Figure 41: Modelsim from teerminal

- Select Jumpstart

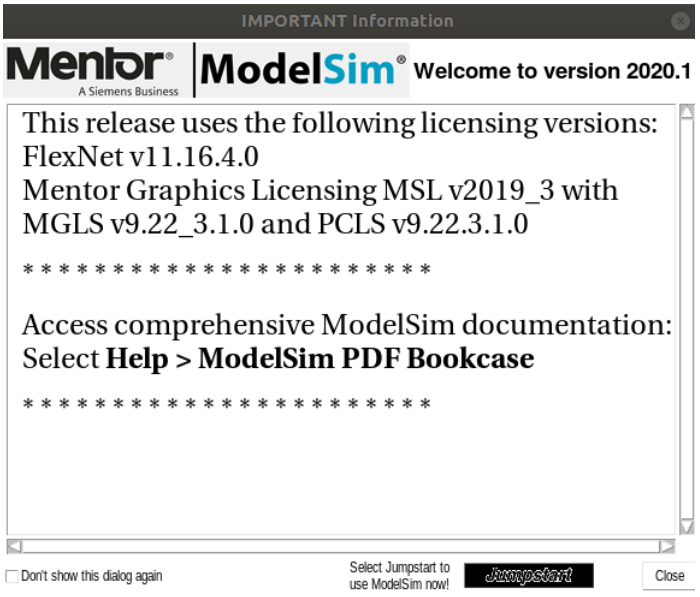


Figure 42: Select jumpstart.

- Create a New project.

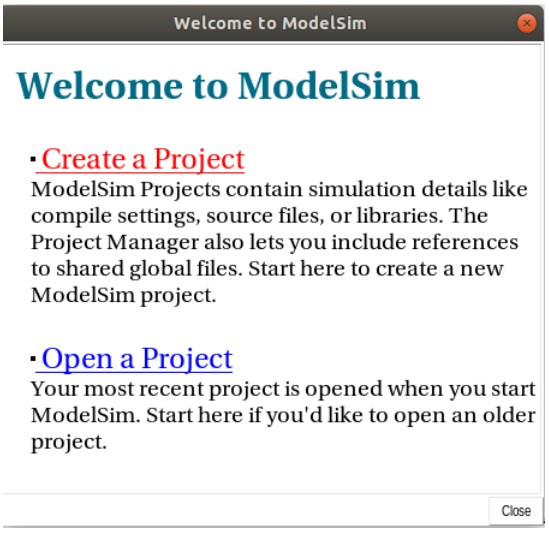


Figure 43: Create a new project.

- Add the VHDL file already saved previously

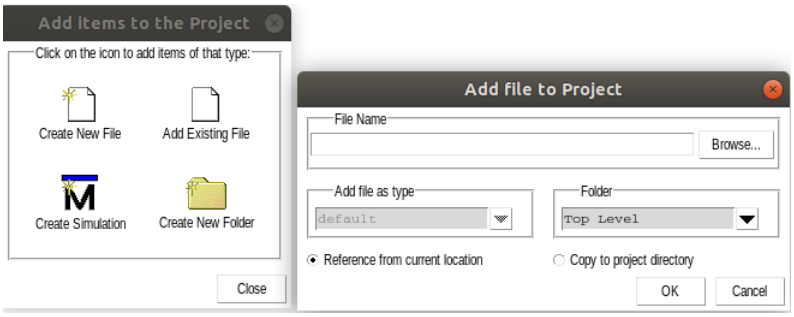


Figure 44: Add testbench and other files

- Right click –> compile –> compile all

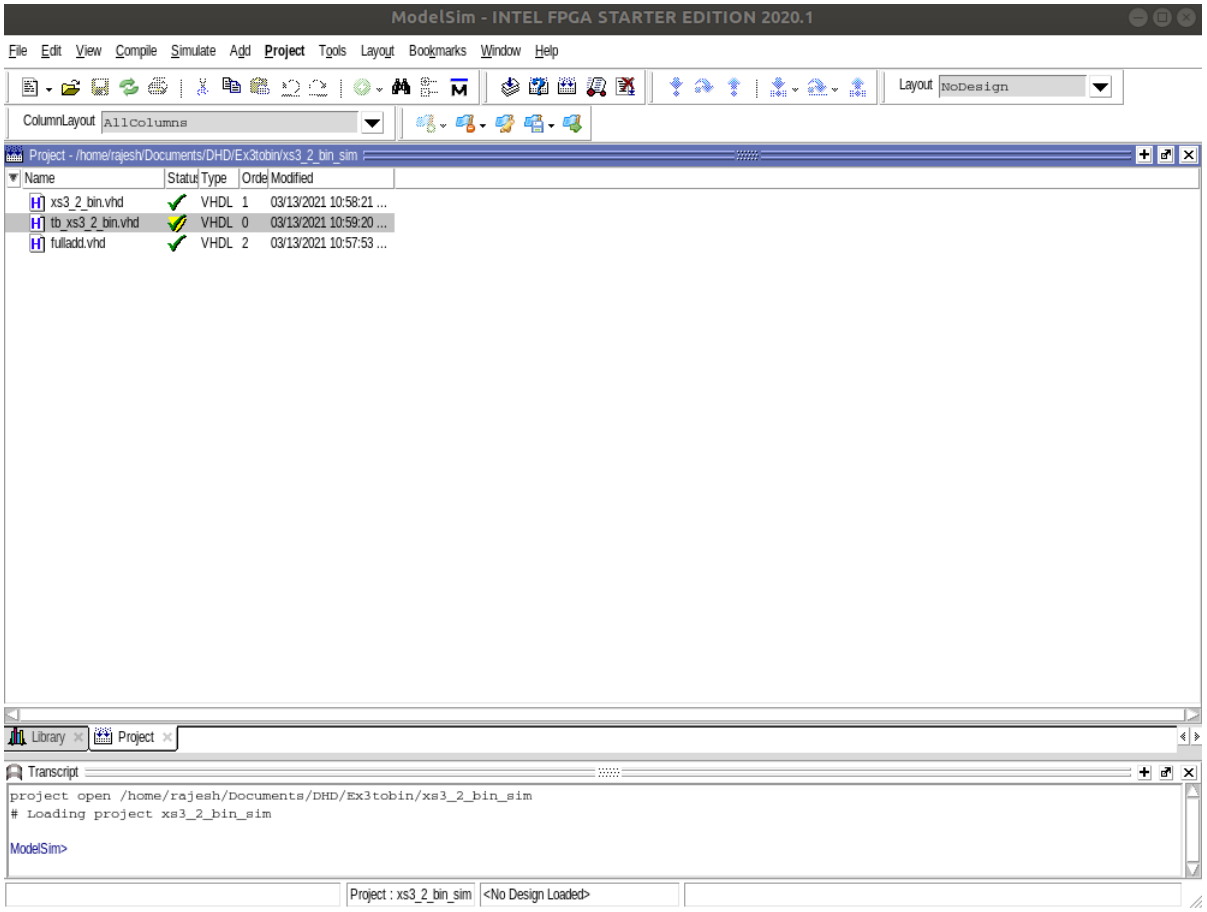


Figure 45: Compiled files

- Once it has been compiled , select simulation –> Start Simulation.

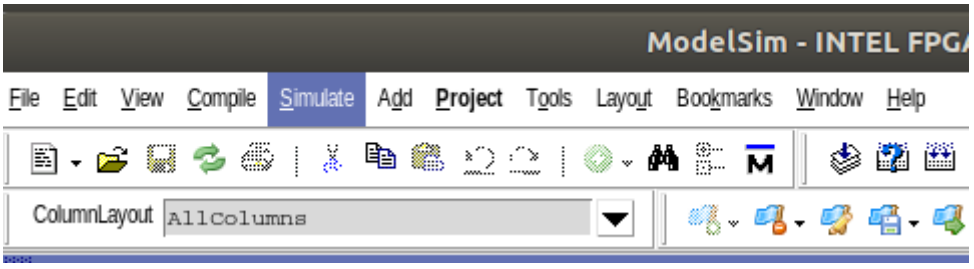


Figure 46: simulation

- Select the testbench file

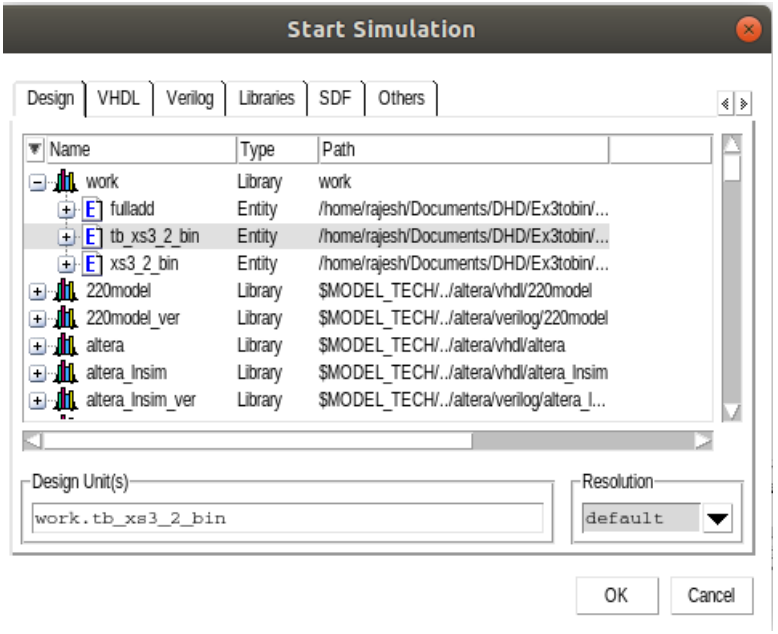


Figure 47: Select testbench file

- The given window will open
  Right click in the blue space –> add –> to wave –> all in region
  Once this is done , a waveform window will appear.

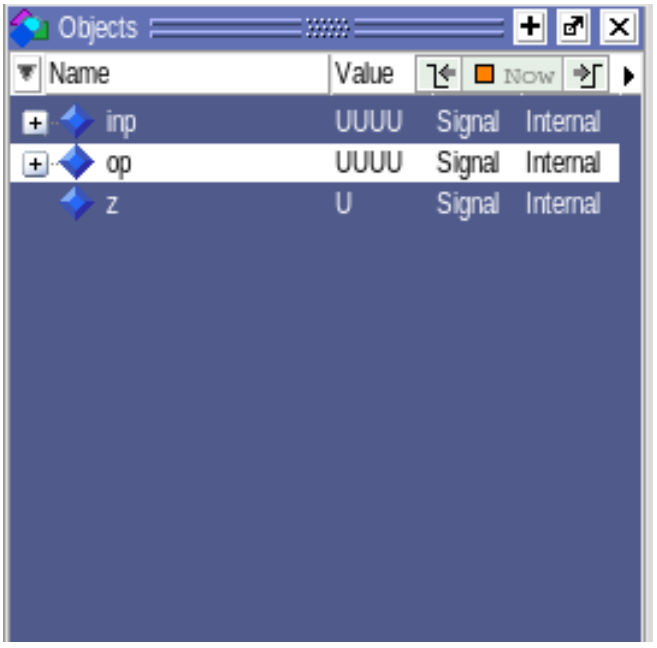


Figure 48: Add to wave

- Click on Run all

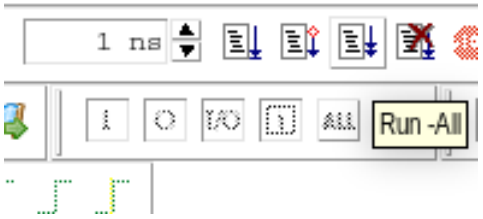


Figure 49: Run all

- Waveform is ready

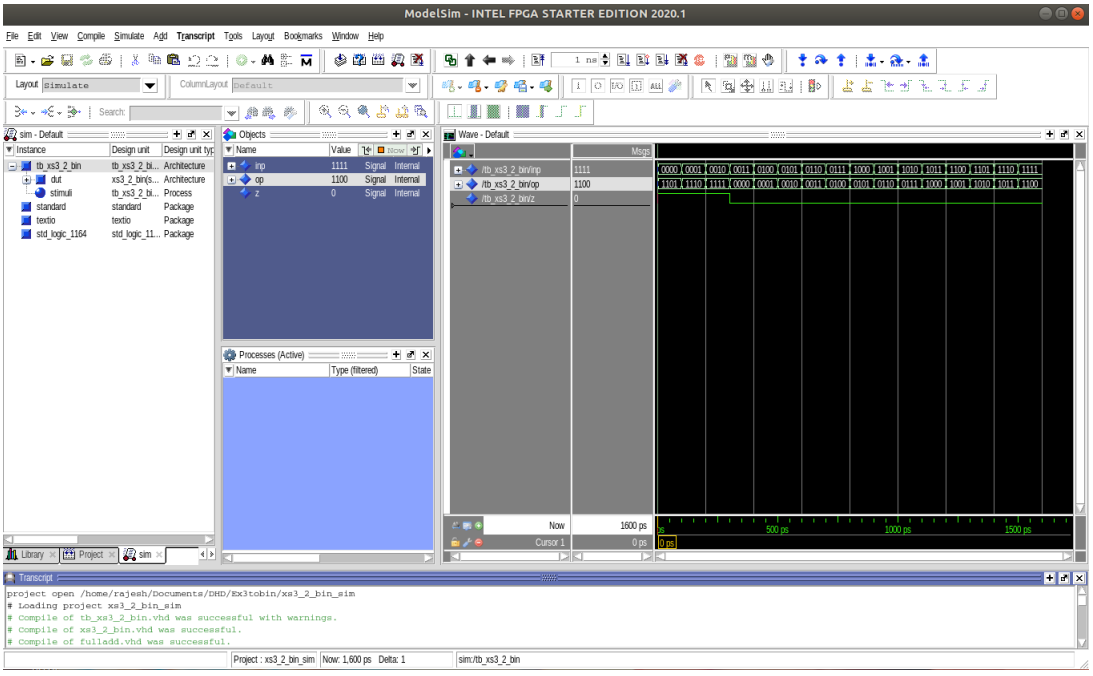


Figure 50: Waveform