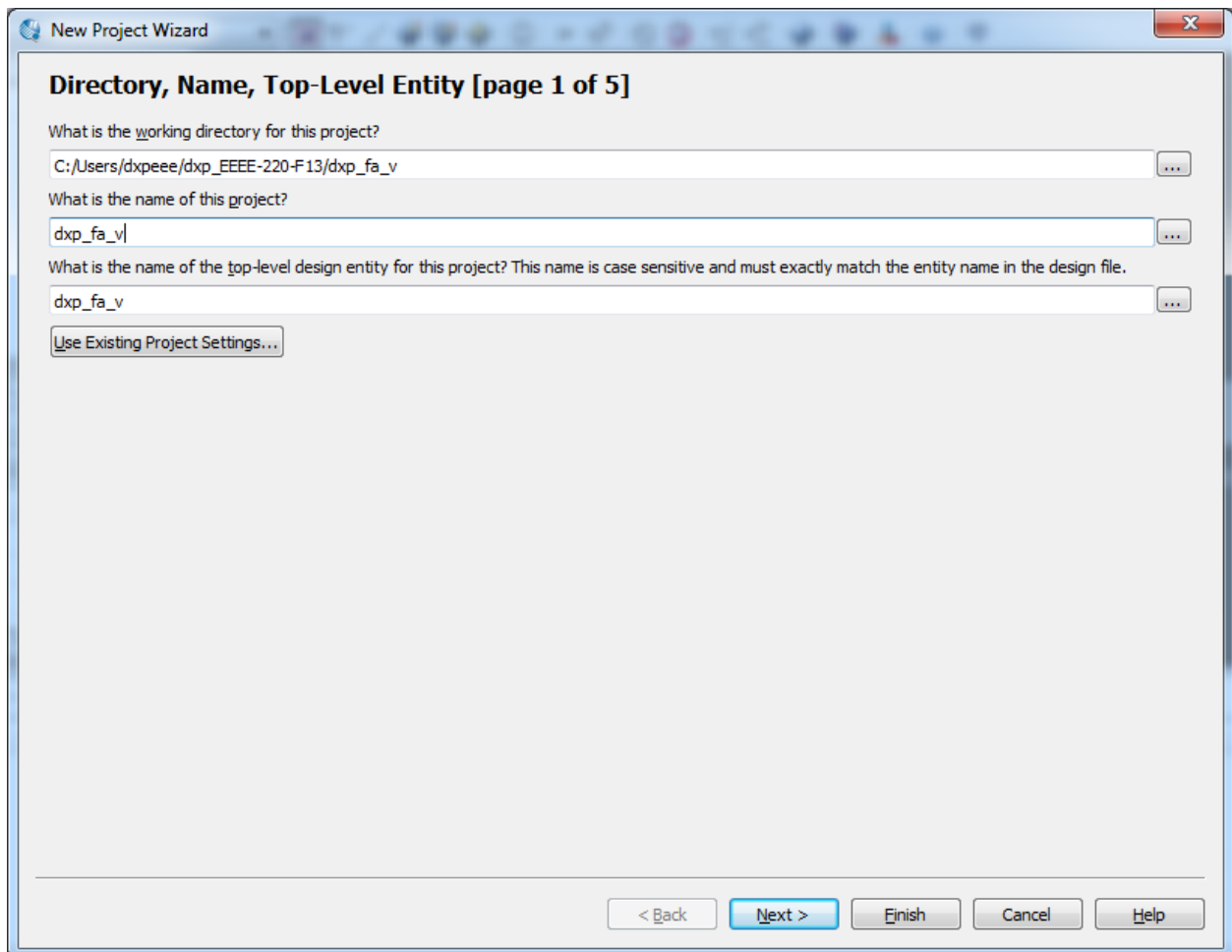


EEEE-220-Lab4

- 1) From mycourses download on your desktop this pdf file. This lab consists of four parts:
 - a. A full-adder implementation in VERILOG,
 - b. A full-adder VERILOG testbench and simulation,
 - c. A 4-bit ripple carry adder/subtractor implementation in VERILOG,
 - d. A 4-bit ripple carry adder/subtractor testbench and simulation.
- 2) Start by opening Altera Quartus II and creating a new project. File > New Project Wizard.
- 3) Select to create a new project directory, project name, and top-level entity called *fml_fa_v*. These should be created in the directory: *fml_EEEE-220-F13*, as shown below.



- 4) In the Family and Device Settings step make sure you select the right device: EP4CE22F17C6.

New Project Wizard

Family & Device Settings [page 3 of 5]

Select the family and device you want to target for compilation.
You can install additional device support with the Install Devices command on the Tools menu.

Device family

Family: **Cyclone IV E**

Devices: **All**

Target device

☐ Auto device selected by the Fitter

☒ Specific device selected in 'Available devices' list

☐ Other: n/a

Show in 'Available devices' list

Package: **Any**

Pin count: **Any**

Speed grade: **Any**

Name filter:

☒ Show advanced devices ☐ HardCopy compatible only

Available devices:

Name	Core Voltage	LEs	User I/Os	Memory Bits	Embedded multiplier 9-bit elements	PLL	GL
EP4CE22E22...	1.0V	22320	80	608256	132	4	20
EP4CE22F17...	1.2V	22320	154	608256	132	4	20
EP4CE22F17...	1.2V	22320	154	608256	132	4	20
EP4CE22F17...	1.2V	22320	154	608256	132	4	20
EP4CE22F17...	1.2V	22320	154	608256	132	4	20
EP4CE22F17...	1.0V	22320	154	608256	132	4	20
EP4CE22E17...	1.0V	22320	154	608256	132	4	20

Companion device

HardCopy:

☐ Limit DSP & RAM to HardCopy device resources

< Back

Next >

Finish

Cancel

Help

5) In the EDA Tools Settings step select ModelSim-Altera and the Format VERILOG.

New Project Wizard

EDA Tool Settings [page 4 of 5]

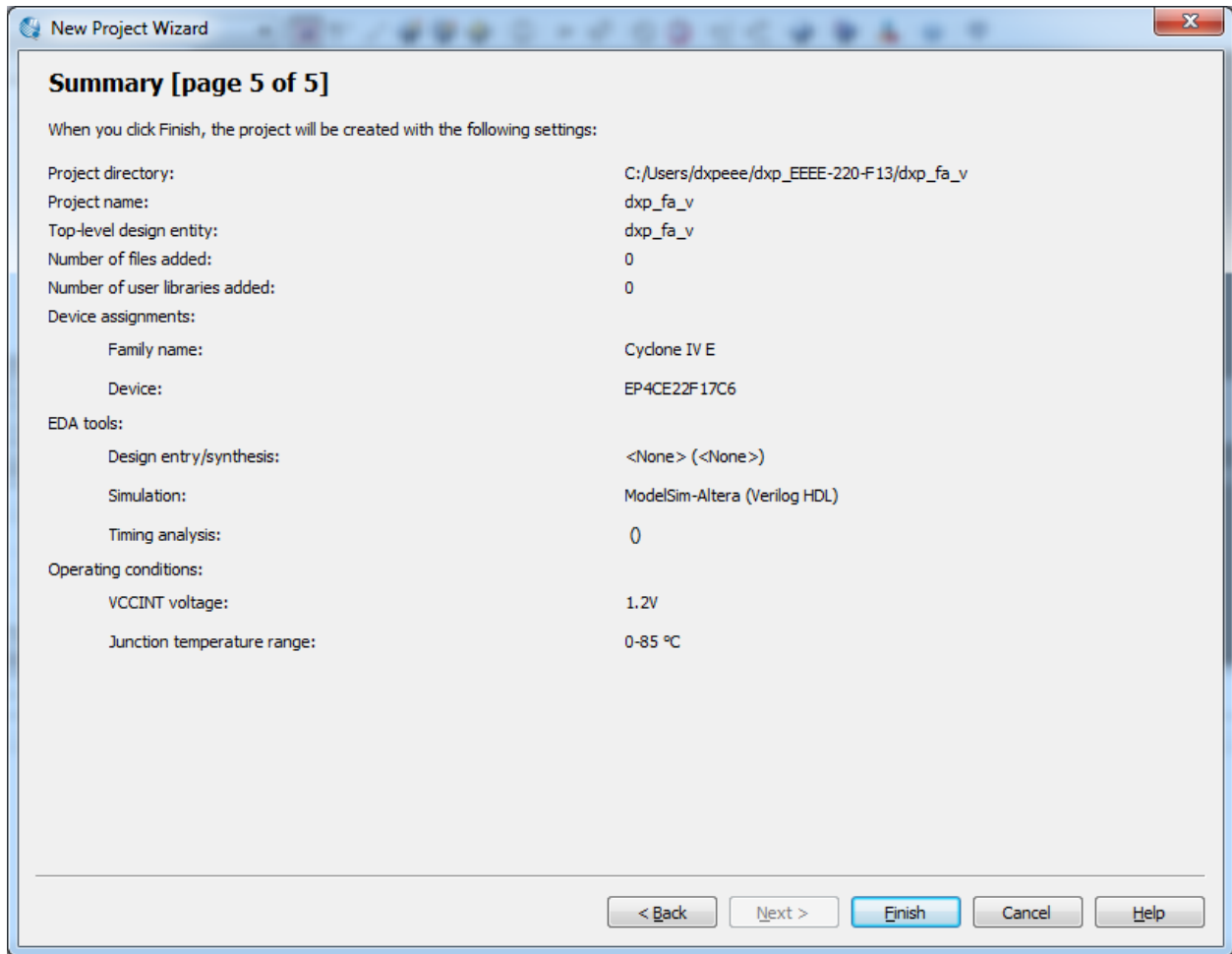
Specify the other EDA tools used with the Quartus II software to develop your project.

EDA tools:

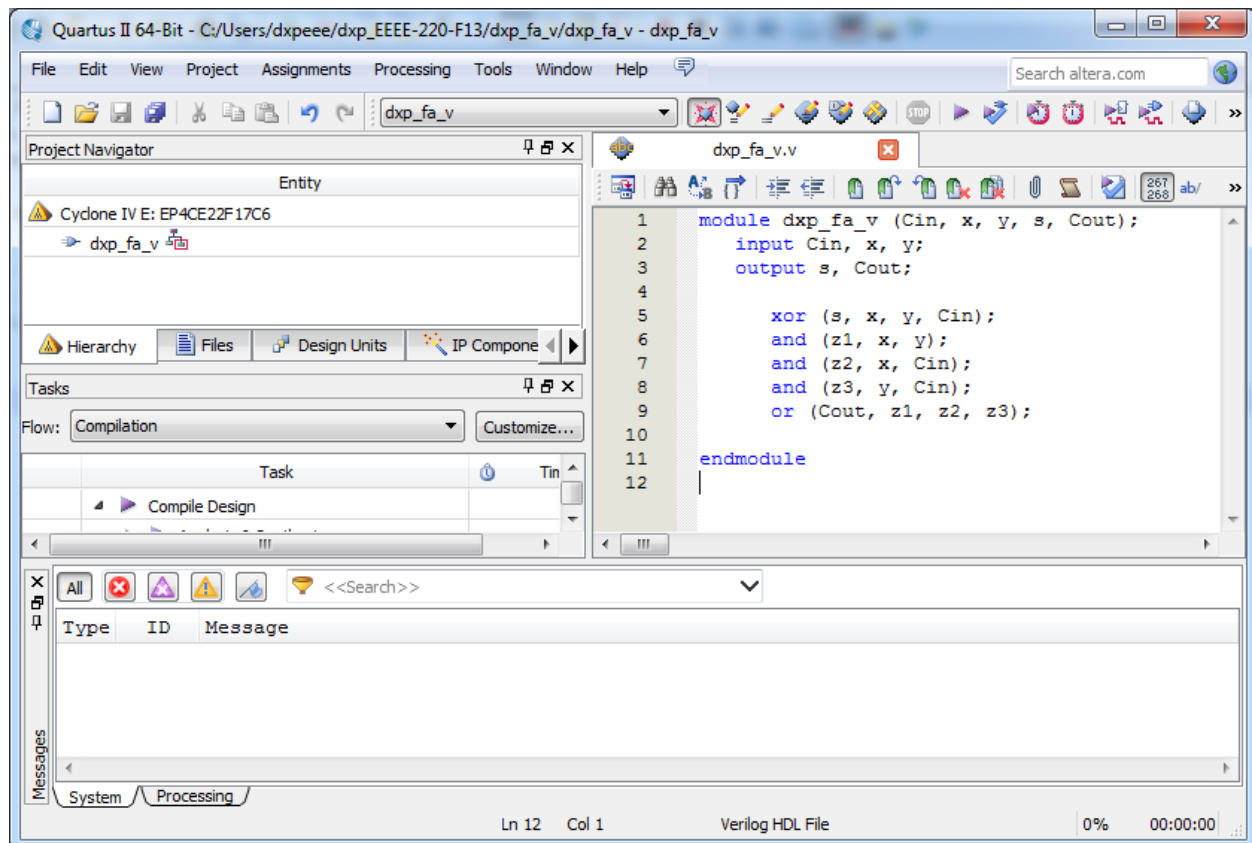
Tool Type	Tool Name	Format(s)	Run Tool Automatically
Design Entry/Synthesis	<None>	<None>	<input type="checkbox"/> Run this tool automatically to synthesize the current design
Simulation	ModelSim-Altera	Verilog HDL	<input type="checkbox"/> Run gate-level simulation automatically after compilation
Formal Verification	<None>		
Board-Level	Timing	<None>	
	Symbol	<None>	
	Signal Integrity	<None>	
	Boundary Scan	<None>	

< Back Next > Finish Cancel Help

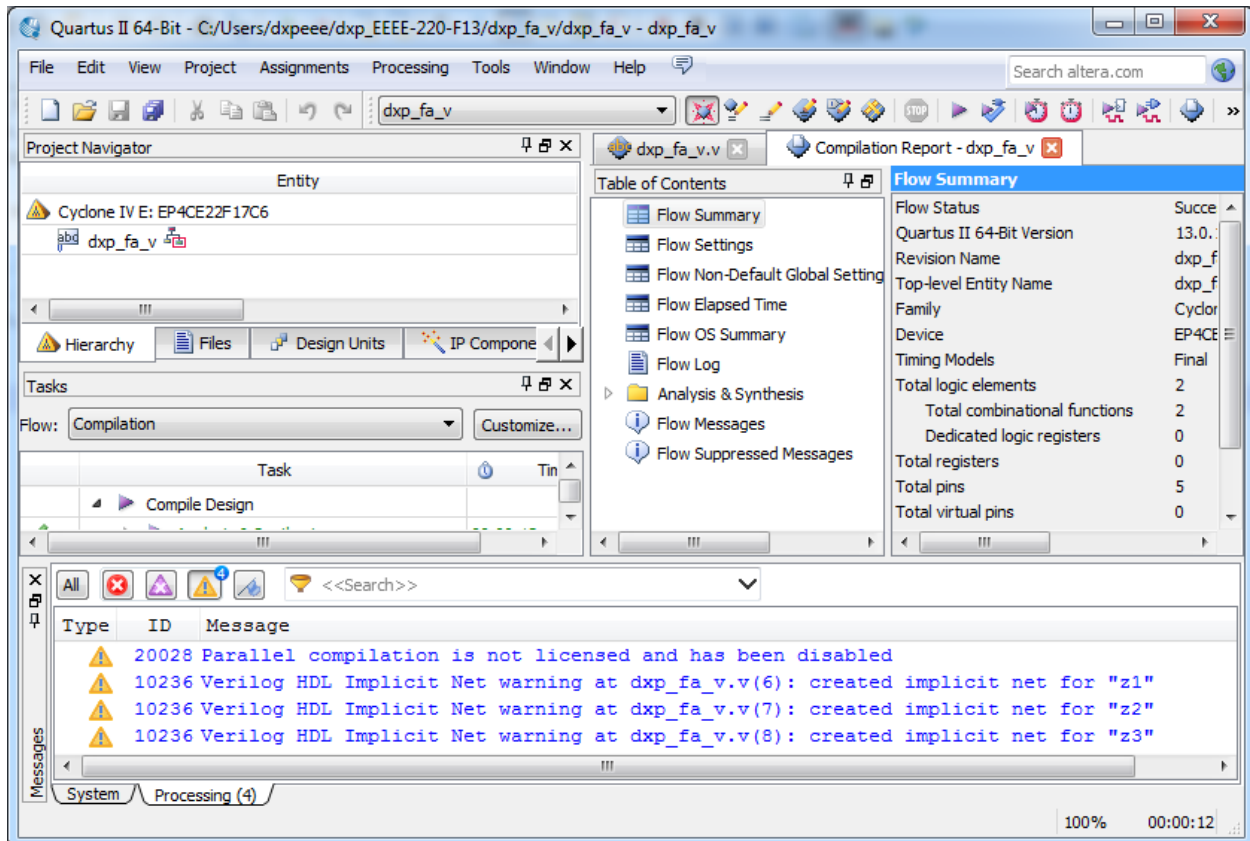
6) Before you click Finish review the summary report.



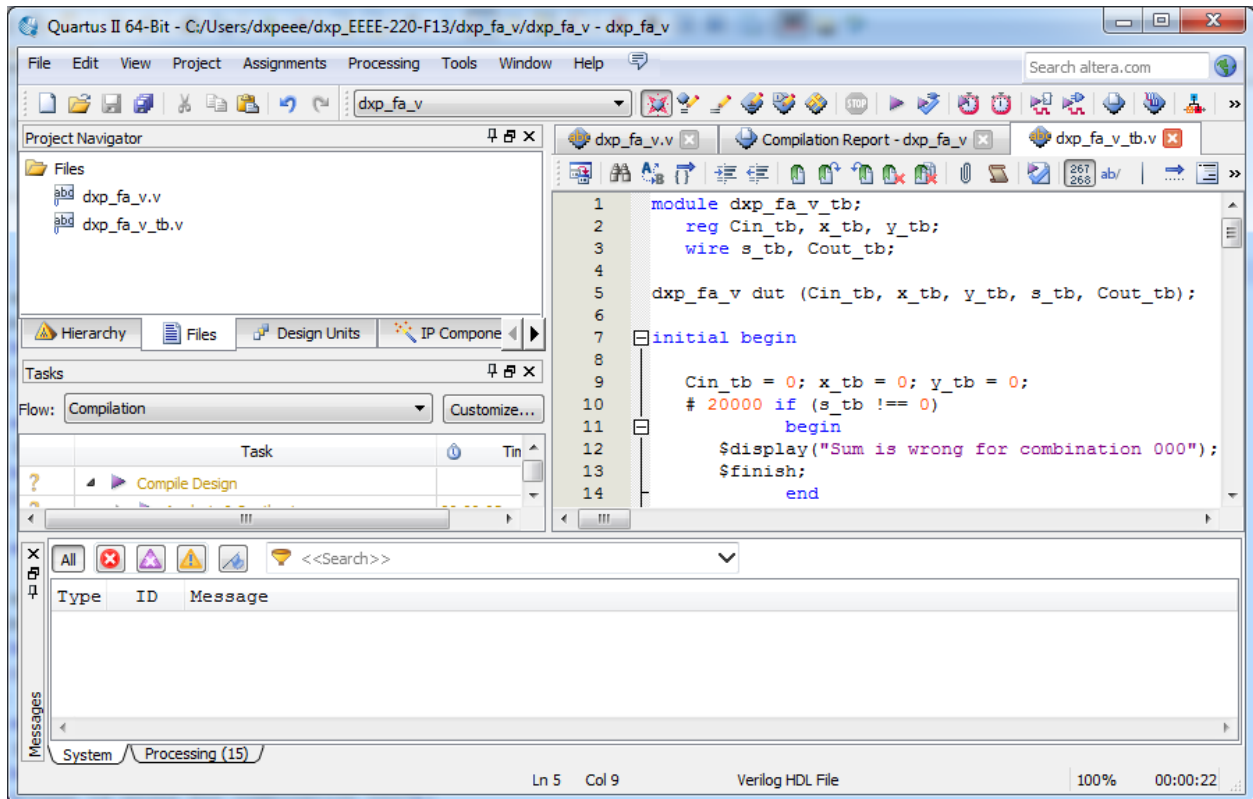
- 7) At this point we have a top-level entity name, but not a file to support it.
- 8) Go to File > New > Verilog HDL file. Copy/paste or even better type the code of the full-adder captured in the file fulladd.v.
- 9) Click Save and accept to save as recommended: fml_fa_v. Leave the *add file to project* box checked.
- 10) At this point your main window should look as follows:



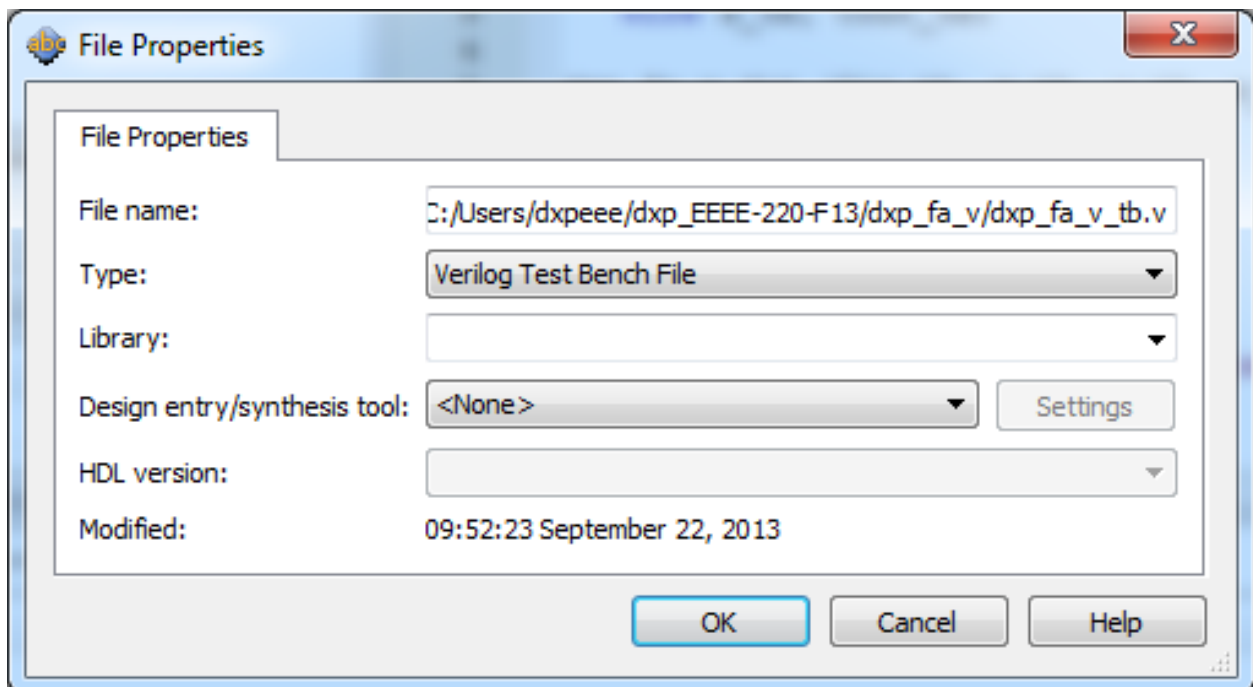
- 11) **Now Remember: the top-level file and module names have to be identical!** Therefore, change the module name inside the code to `fml_fa_v`. Otherwise, the compiler will give you an error saying that it cannot find the top-level module.
- 12) Normally, at this point you would compile by clicking on the magenta play button. However, because we won't use the board in this lab either, and because for RTL Simulation purposes analysis and synthesis are sufficient, click on Analysis and Synthesis, i.e. the button to the right of the magenta play button. This will analyze and synthesize the circuit, but it will not proceed to fitting and FPGA assembly, saving you some time in the process. If all goes well you should only get some 4 warnings.



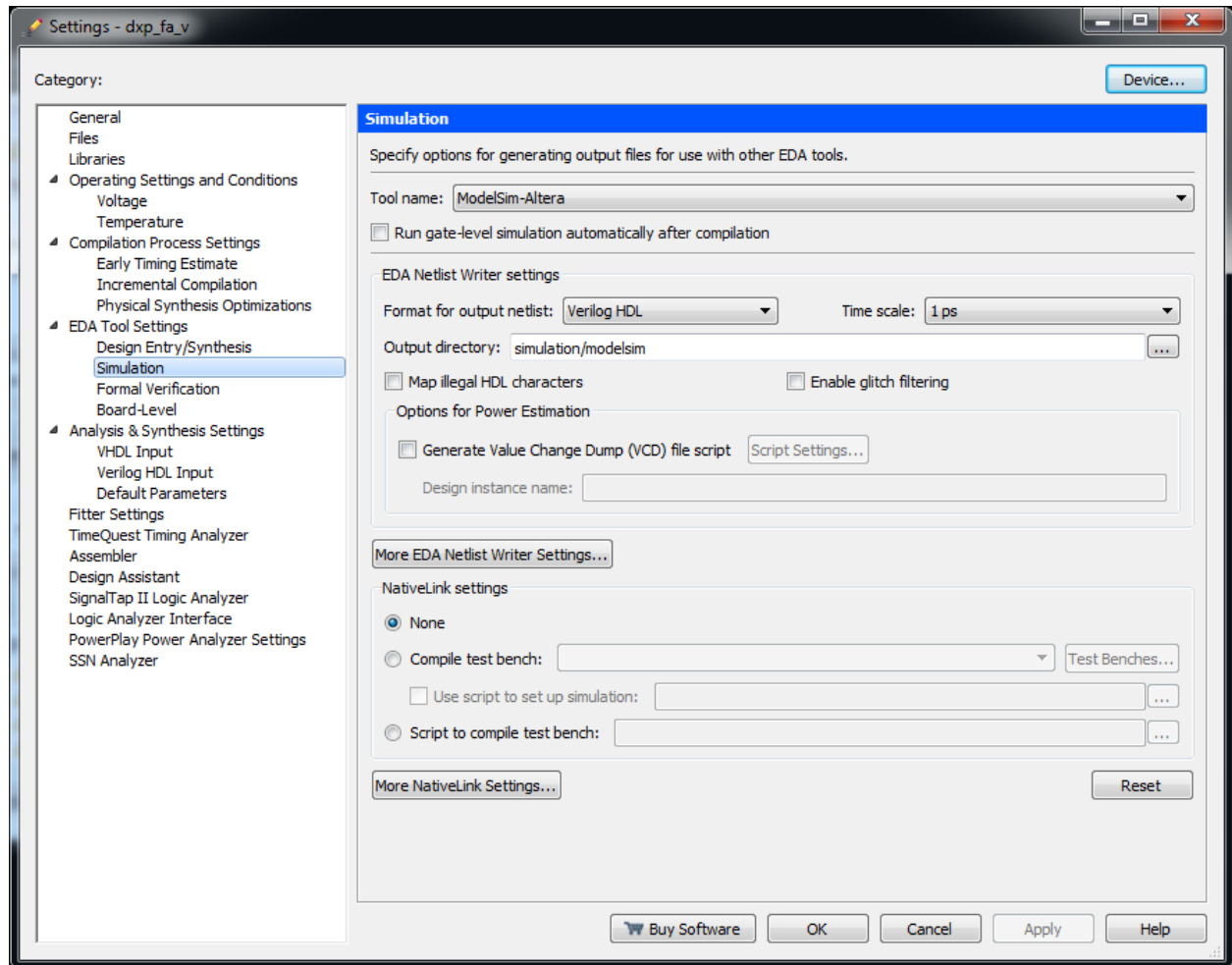
- 13) This completes the first part of this lab. Don't close the project.
- 14) In the second part we want to verify the correct functionality of this full adder using a dedicated testbench. You can find details about the structure of a VERILOG testbench in the lecture presentation: [dxd_chapter3b.ppt.pdf](#).
- 15) Go to File > New > Verilog HDL file.
- 16) Copy/paste or even better type the code of the full adder testbench captured in the file `fulladd_tb.vhd`. Save as: `fml_fa_v_tb.vhd`.
- 17) Change the entity name inside the code to `fml_fa_v_tb`. Save again.



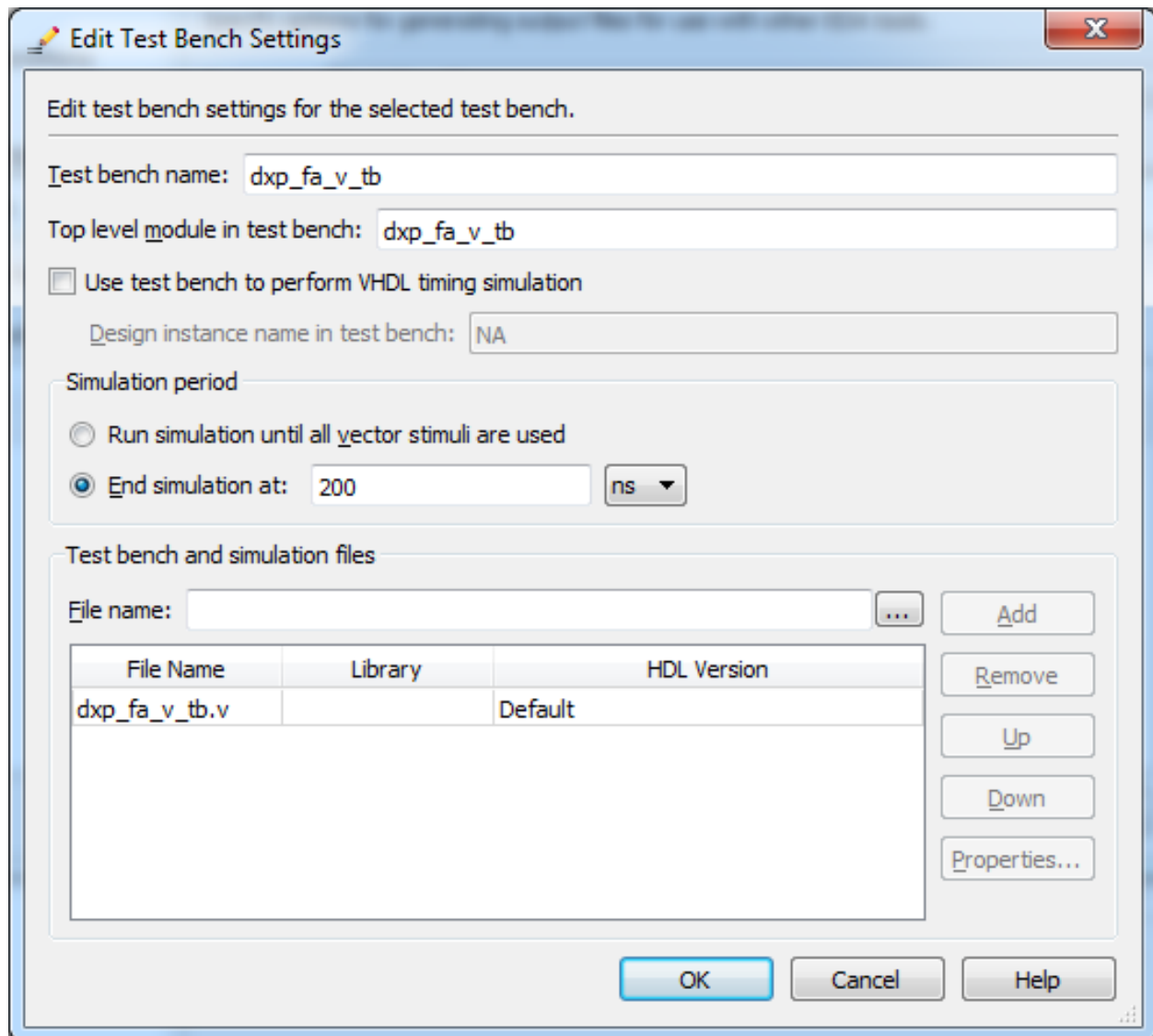
- 18) With the testbench file selected, go to File > File Properties and select the file to be a VERILOG Testbench as shown below:



- 19) Once per project you have to configure the “NativeLink” settings. Every CAD-IDE may have its own way of doing this.
- 20) Go to Assignments > Settings > EDA Tools Settings > Simulation.



- 21) To start with make sure you have all above settings.
- 22) Now, select to compile test bench. Click on Test Benches > New. Enter your testbench name without extension.
- 23) Further, choose to end simulation at 200 ns. This may need to change depending on the design and simulation circumstances.
- 24) Finally, browse to your testbench file and add it. Before you click OK your window should look as below:



25) Click three times OK.

26) One other thing that in principle has to be done once per project, unless the lab computers are re-imaged, is to tell Quartus where ModelSim resides.

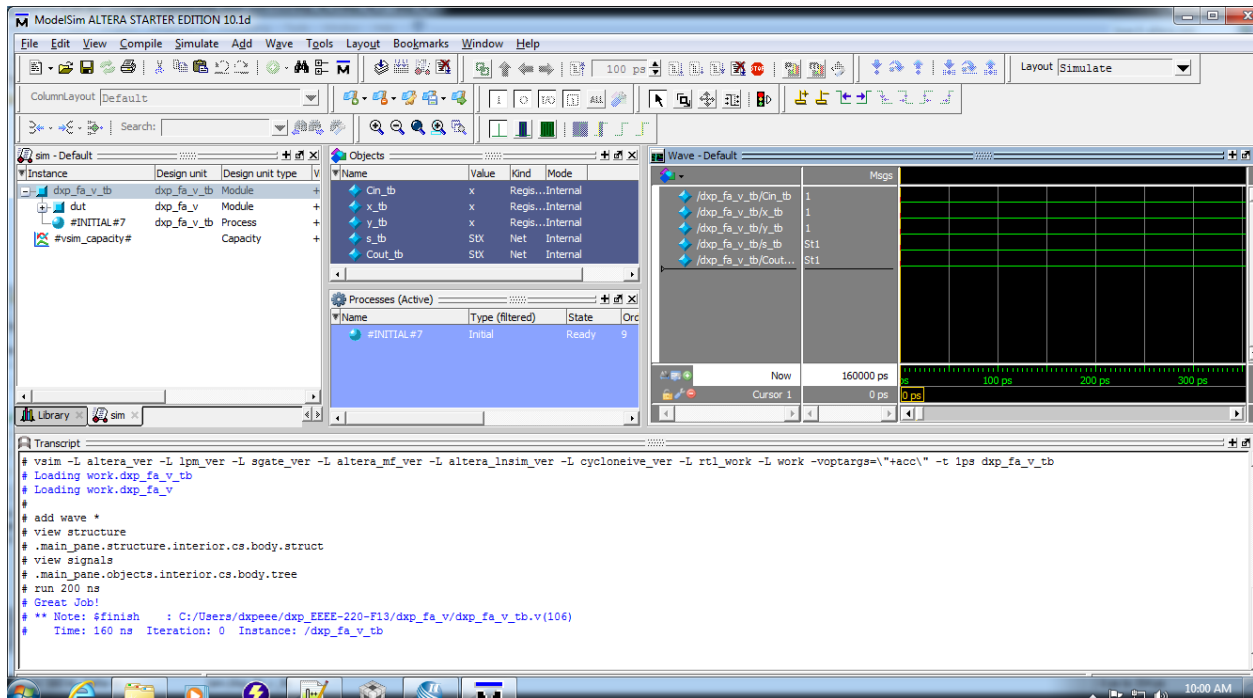
27) Go to Tools > Options > EDA Tools Options. The path to the ModelSim executable should be:

C:\altera\13.0\modelsim_ase\win32aloem

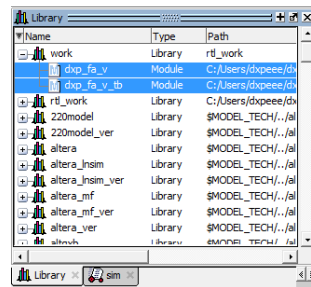
28) Click OK.

29) Analyze and synthesize again.

30) Now open the ModelSim simulator: Tools > Run Simulation Tools > RTL Simulation. Depending on your settings, ModelSim may actually compile and simulate when it is started. If the simulation is successful, it will arrive at the last "\$finish" in the testbench file. When it asks you if you want to finish select NO. Otherwise it will close. After a few moments the ModelSim main window will look like this:

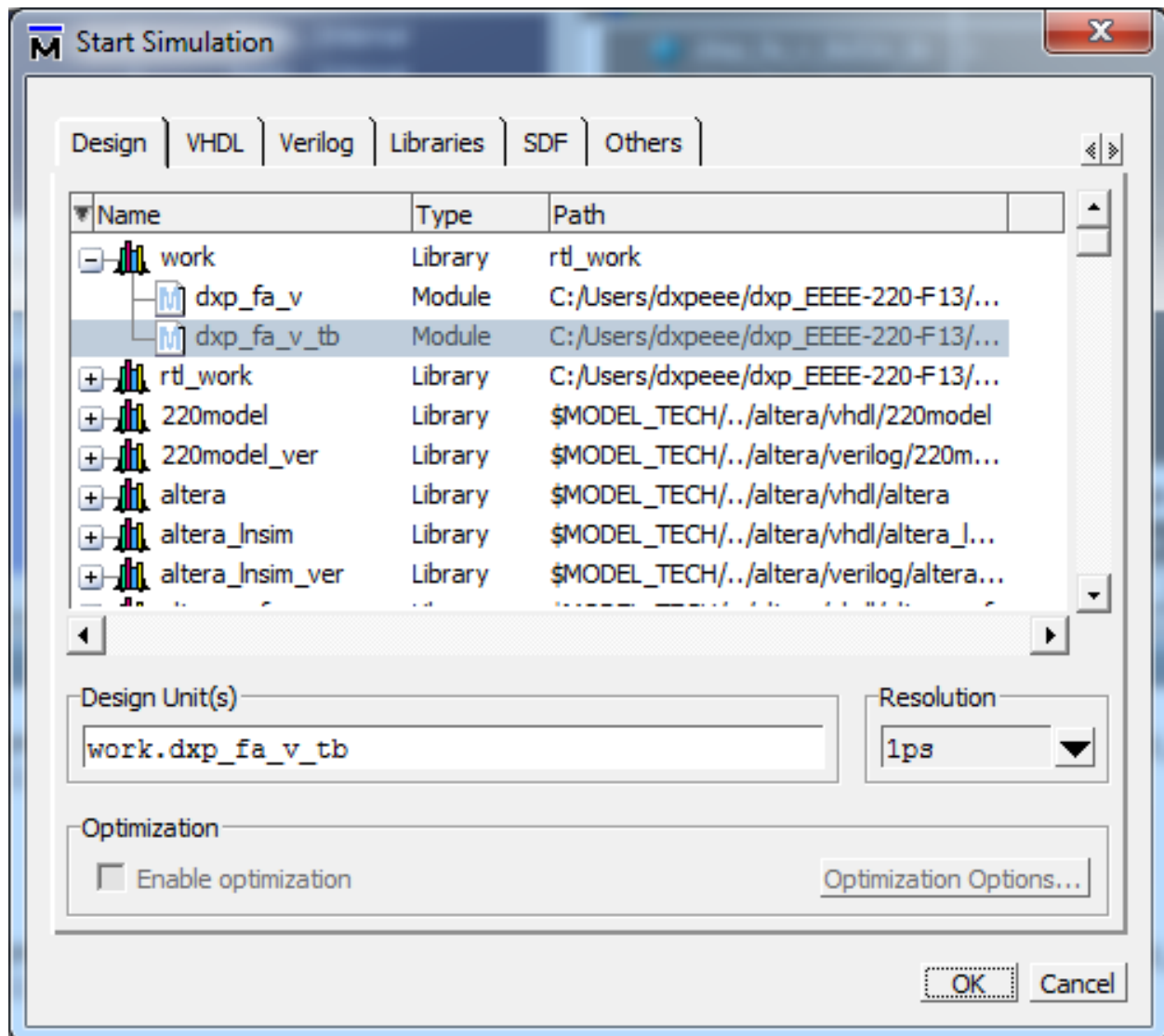


31) Click on the Library tab on the LHS and select the two files in your working directory.

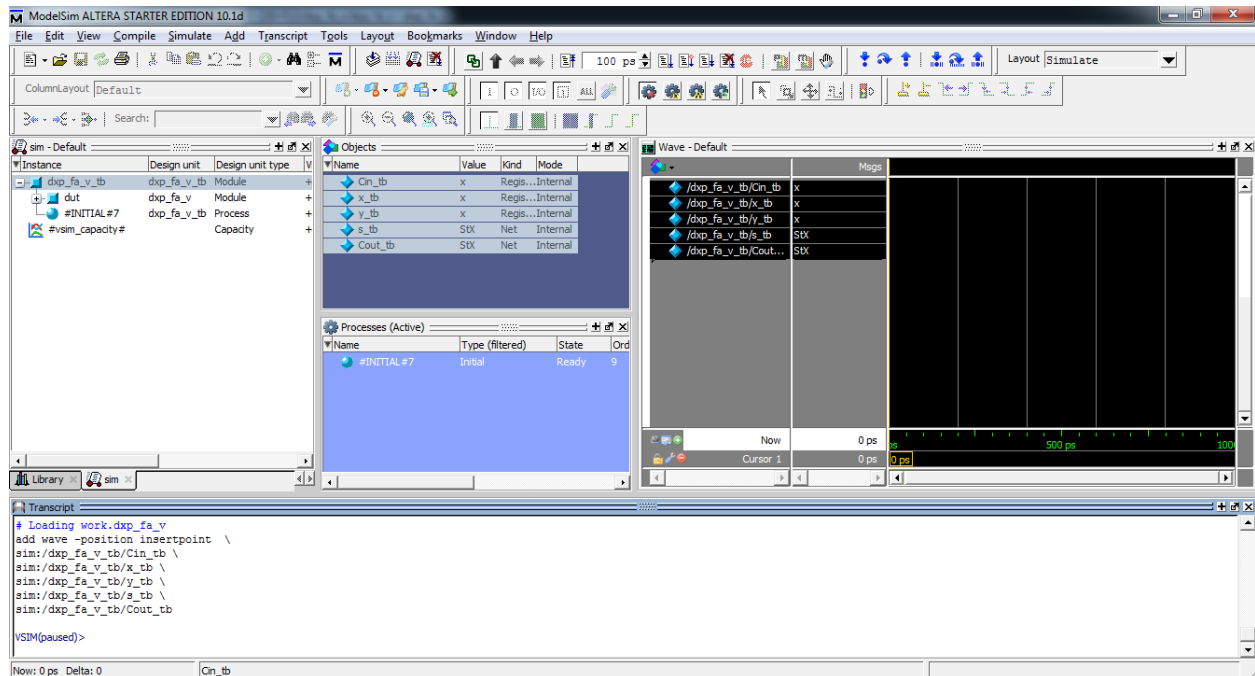


32) Right-click and select to Recompile.

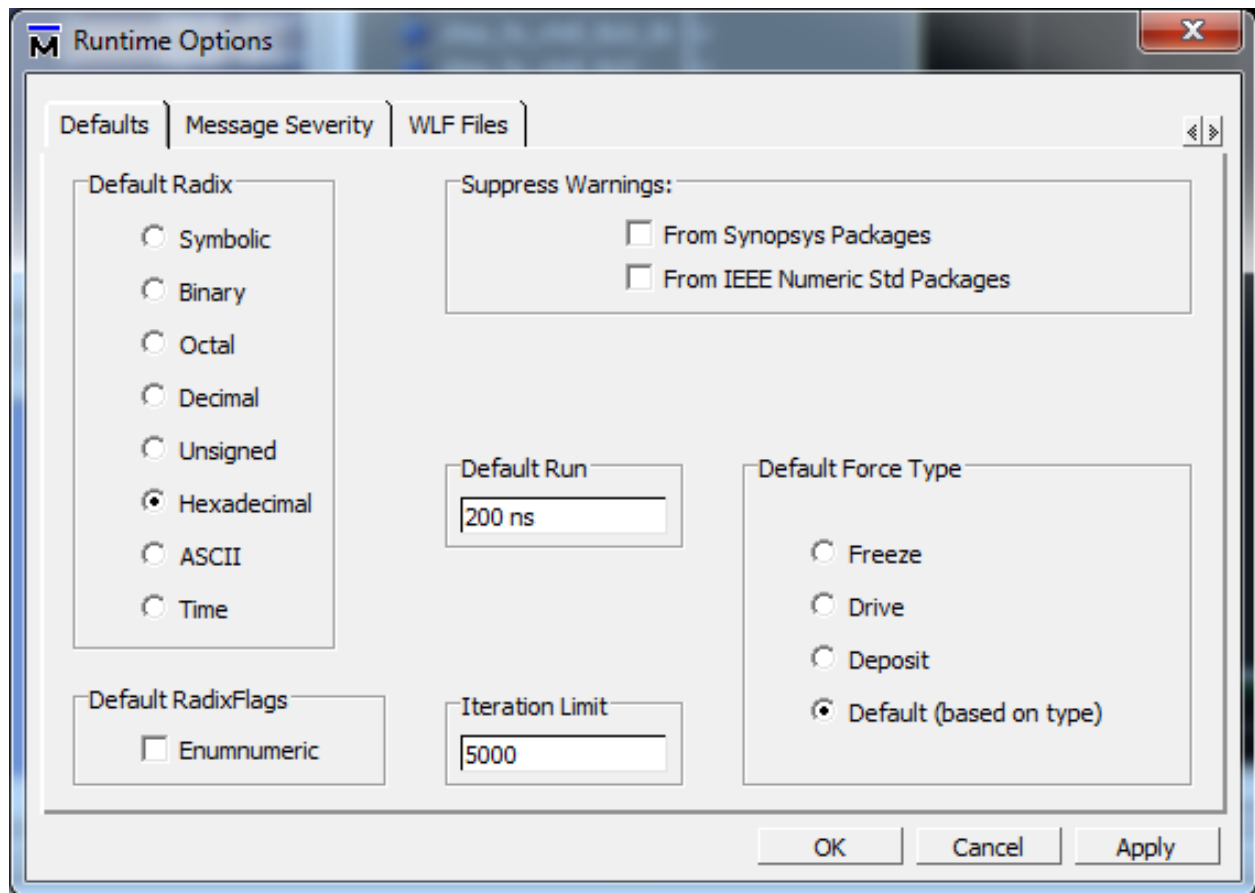
33) To switch ModelSim in simulator mode go to: Simulate > Start Simulation. A new window pops up. Collapse the work library, select your testbench and click OK.



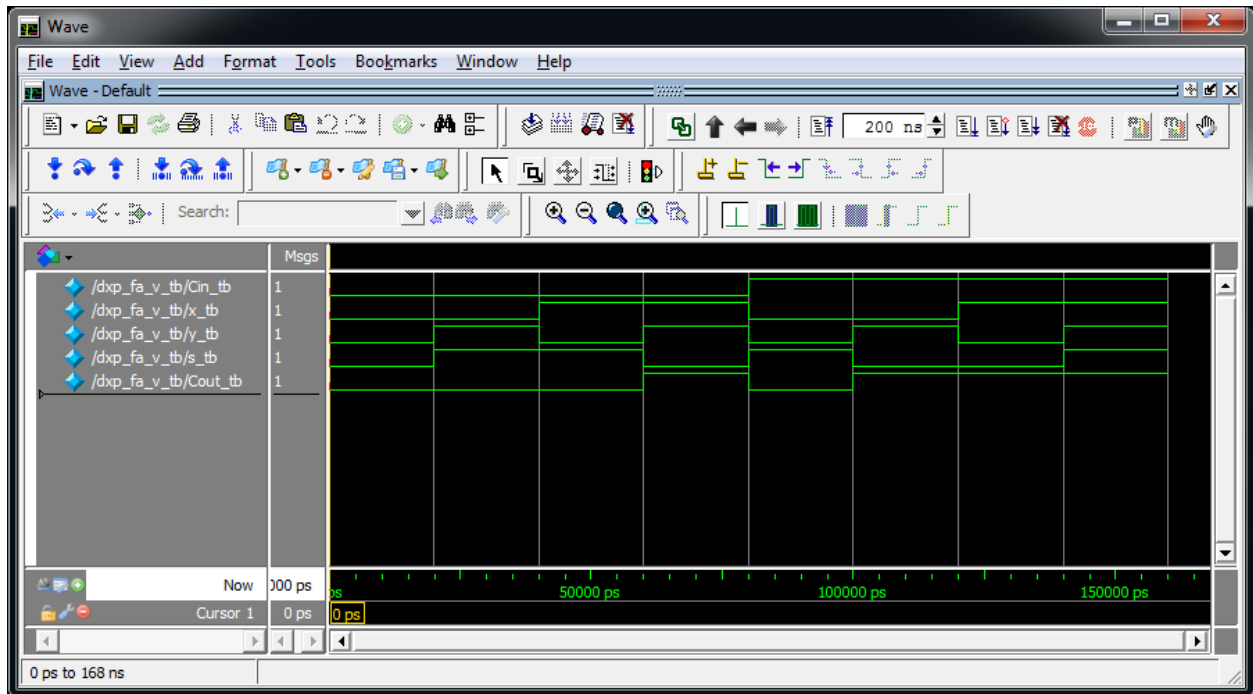
34) New panes open. The sim pane lists all design files, Testbenches, and libraries. The objects pane lists all available objects, i.e. signals. The process pane confirms that your testbench has an active process. Finally, the wave pane will list all signals selected for viewing. Select all signals in the objects pane, right-click and Add Wave. Your window by now should look as shown below:



- 35) At this point you can undock the wave pane. Click on the middle button in the upper RHS corner. Clicking on the same button will dock it again.
- 36) Next click on Simulate > Runtime Options. Choose the Default Run to be 200 ns (may change for different projects), and the default radix to Hexadecimal. Before you click OK, your window should look like the one below:

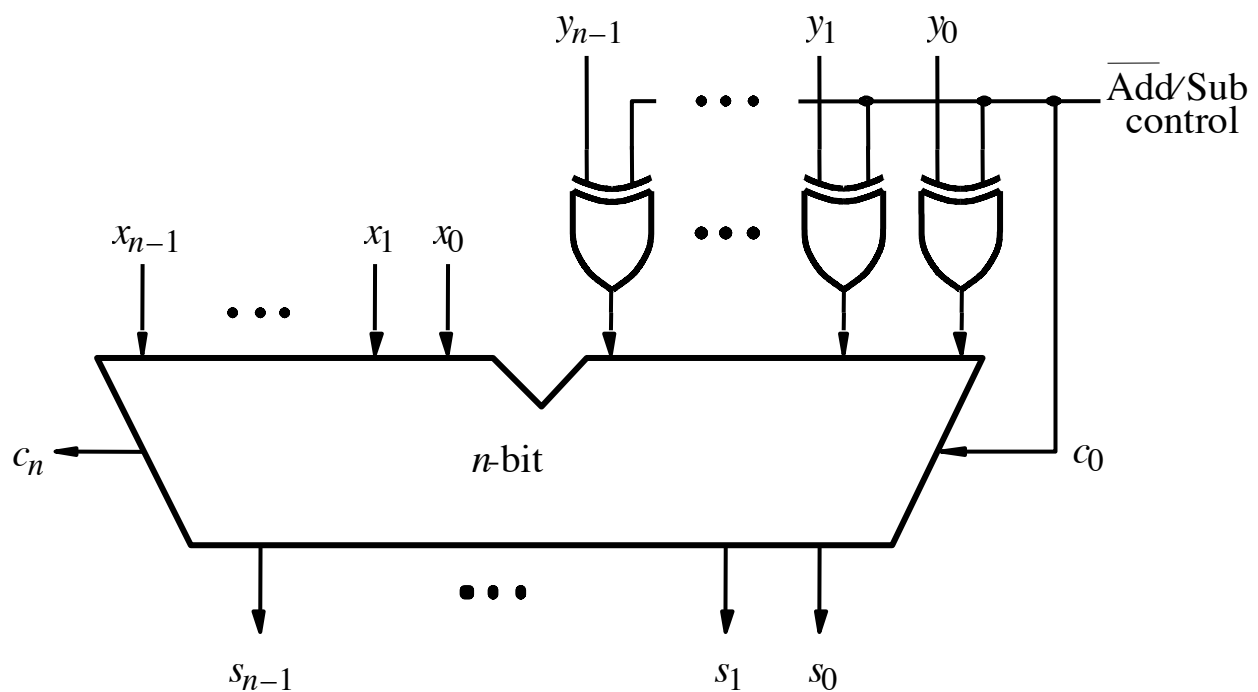


37) Now run the simulation by clicking Simulate > Run > Run-All. If all goes well you'll see the results of the simulation in the wave pane. To view it properly you may need to right-click and select Zoom Full. If there are now simulation errors, all signals are green. If there are simulation errors, those waveforms will be highlighted in red. If any of the assert statements fail, there will be a red marker at that point in time.



38) At this point you can close ModelSim. This concludes the second part of this week's lab.

39) In the third part of this week's lab you'll have to create a project and create a VERILOG description of a **4-bit ripple carry adder/subtractor**. You can use any of the template examples in the lecture presentation: dxd_chapter3a.ppt.pdf. Below is the schematic of this circuit. You can use hierarchical, data flow, or behavioral code. This part is considered done when it compiles without errors.



- 40) In the fourth part you'll have to verify the correctness of your design. You'll have to design a testbench for your circuit following the template in part 2.
- 41) Now, your circuit has 9 inputs. To test your circuit exhaustively, i.e. run every possible combination of the nine inputs, you'll need to run through $2^9 = 512$ different combinations. While this wouldn't be a problem for the simulator, it would be extremely time consuming for you to type in the statements to accomplish this. Later on, we will introduce some constructs that will help in the creation of Testbenches for exhaustive testing.
- 42) Therefore, you will only test a few relevant input combinations. These are:

x	y	Cin=c0=asc	s	Cout=c4
0000	0000	0	0000	0
0000	1111	0	1111	0
1111	1111	0	1110	1
1111	0000	0	1111	0
0101	0101	0	1010	0
0101	1010	0	1111	0
1010	1010	0	0100	1
1010	0101	0	1111	0
0000	0000	1	0000	1
0000	1111	1	0001	0
1111	1111	1	0000	1
1111	0000	1	1111	1
0101	0101	1	0000	1
0101	1010	1	1011	0
1010	1010	1	0000	1
1010	0101	1	0101	1

- 43) Show your working, i.e. compiled and simulated, designs to the TA. Write your report and upload it along with your project archives in the dropbox on mycourses, as described in the lab policy.
- 44) This concludes this week's lab.
- 45) Grading:
- 10 points for full adder VERILOG compilation
 - 10 points for full adder VERILOG testbench simulation
 - 10 points for 4-bit ripple carry adder/subtractor compilation
 - 10 points for 4-bit ripple carry adder/subtractor simulation.