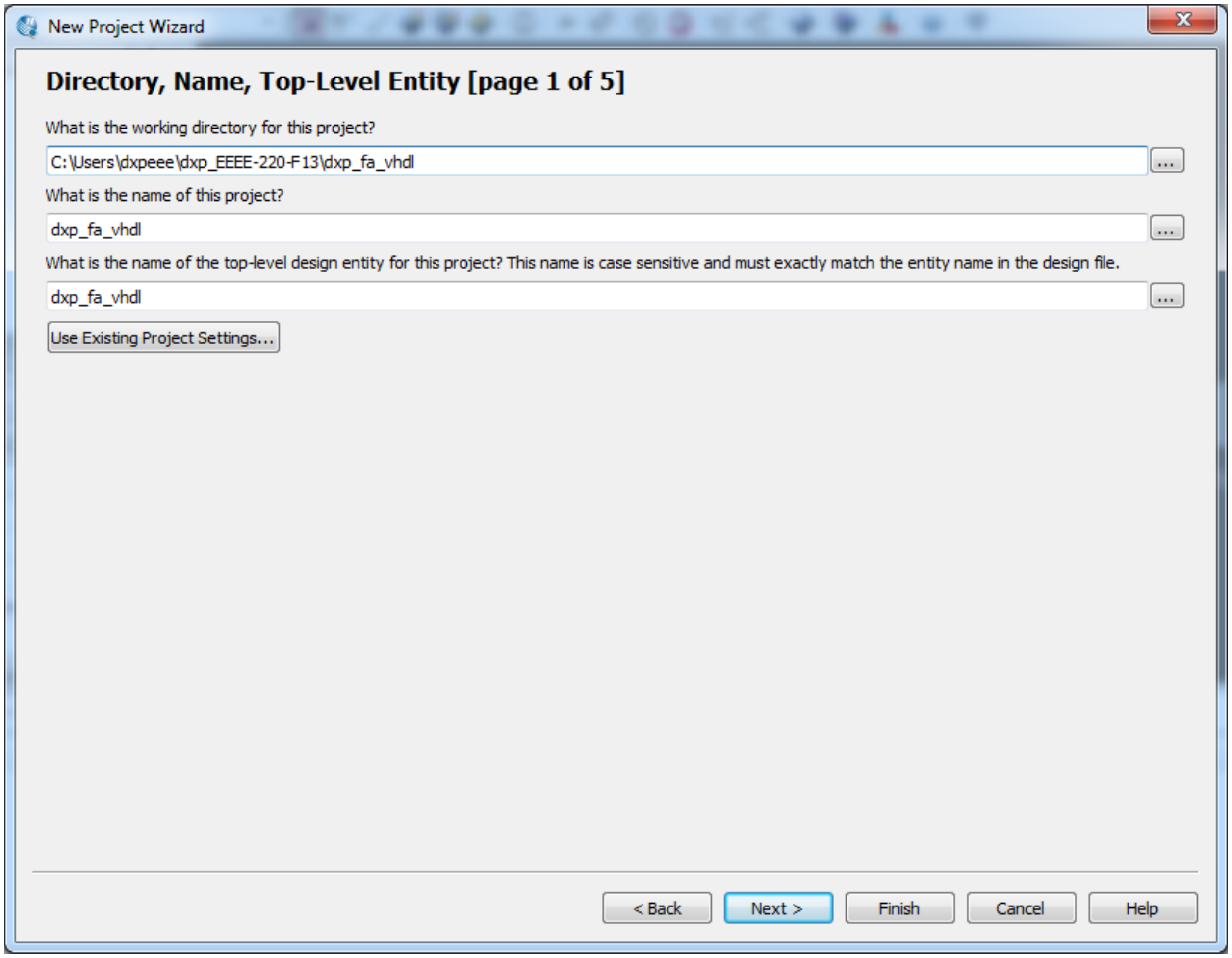
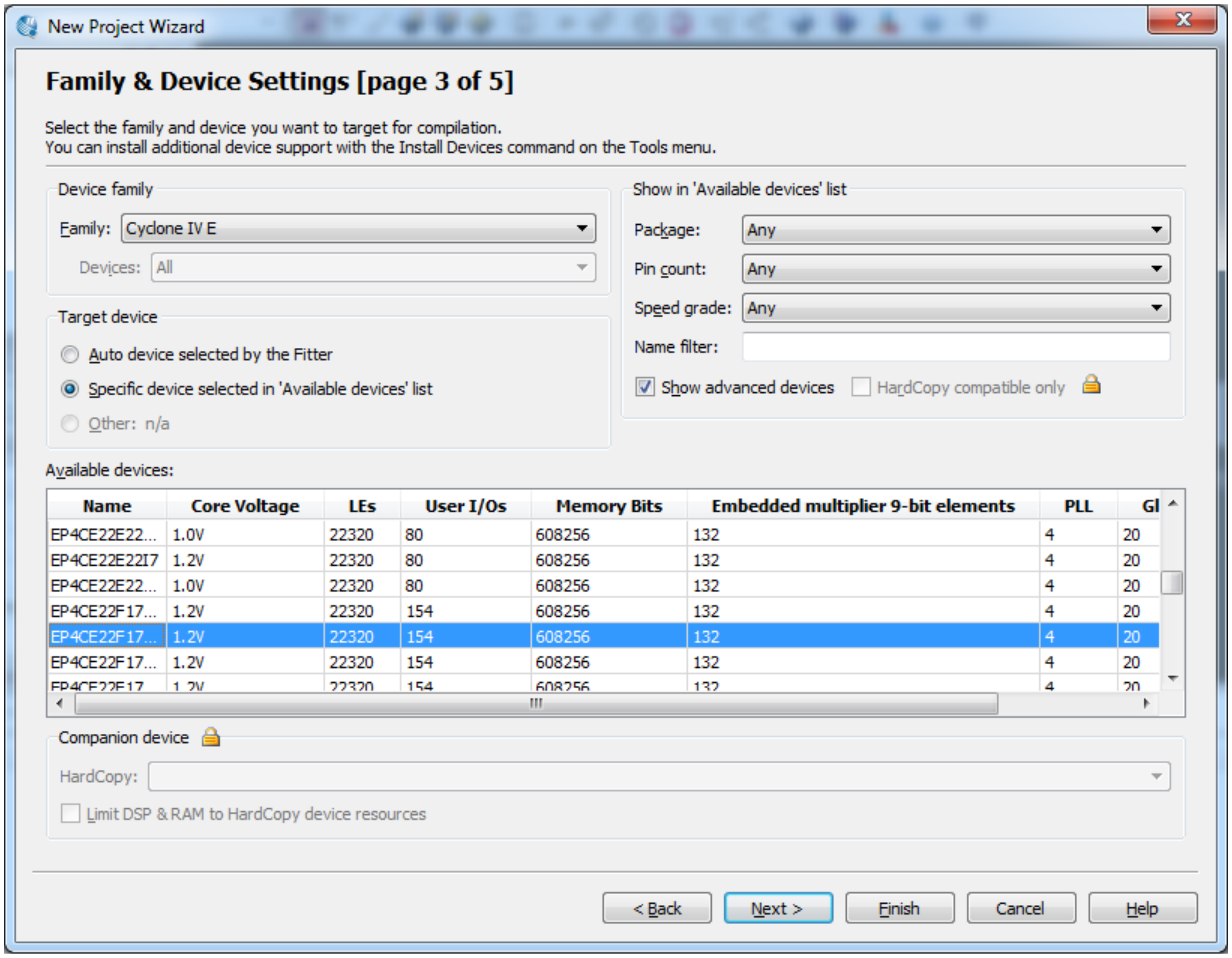
EEEE-220-Lab3

1) From mycourses download on your desktop this pdf file. This lab consists of four parts:
    a. A full-adder implementation in VHDL,
    b. A full-adder VHDL testbench and simulation,
    c. A 4-bit ripple carry adder/subtractor implementation in VHDL,
    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_vhdl*. These should be created in the directory: *fml_EEEE-220-F13,* as shown below.


New Project Wizard
Directory, Name, Top-Level Entity [page 1 of 5]
Use Existing Project Settings...
< Back
Next >
Finish
Cancel
Help

What is the working directory for this project?
C:\Users\dxpeee\dxp_EEEE-220-F13\dxp_fa_vhdl
What is the name of this project?
dxp_fa_vhdl
What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.
dxp_fa_vhdl

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:

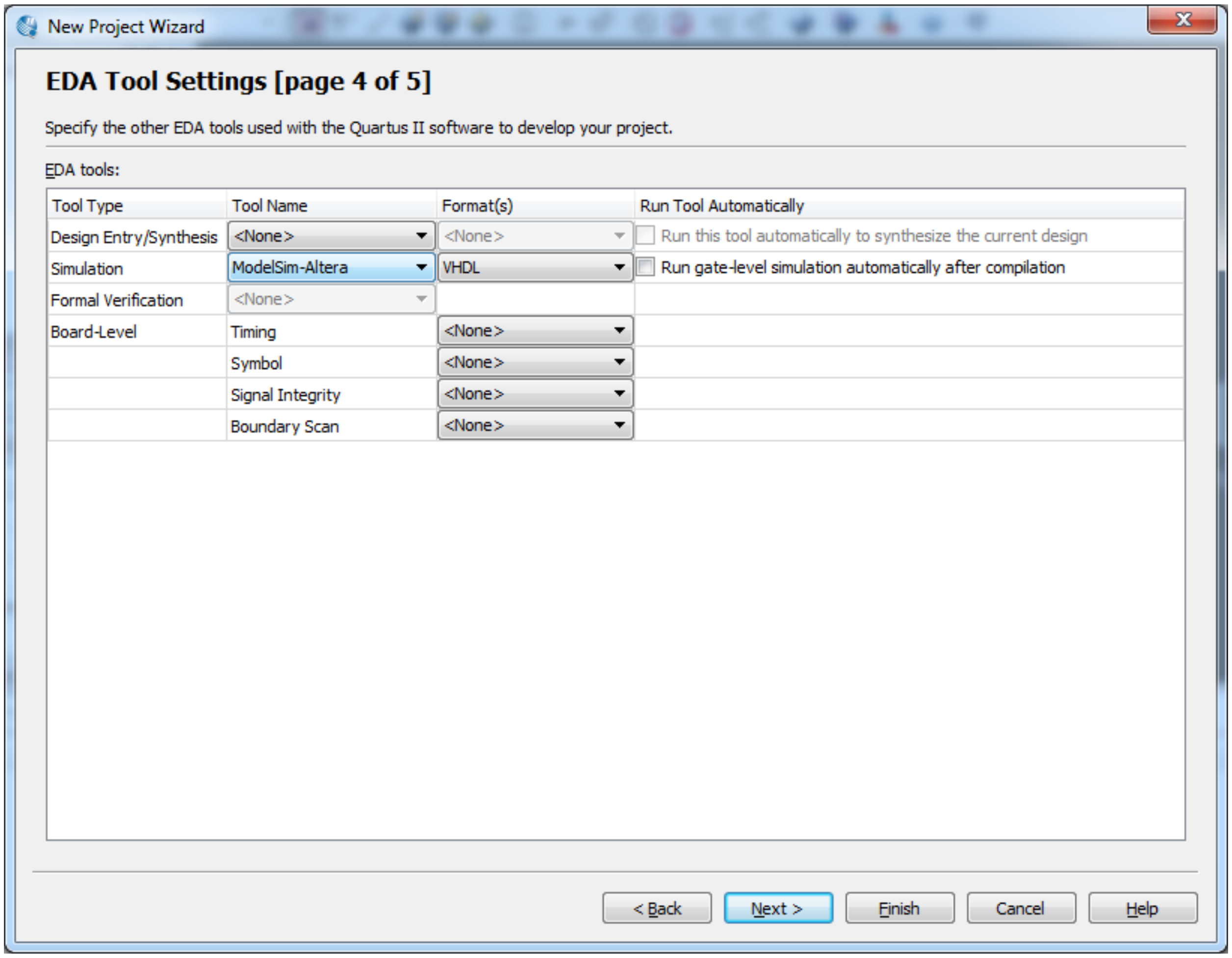
Name	Core Voltage	LEs	User I/Os	Memory Bits	Embedded multiplier 9-bit elements	PLL	Gl
EP4CE22E22...	1.0V	22320	80	608256	132	4	20
EP4CE22E22I7	1.2V	22320	80	608256	132	4	20
EP4CE22E22...	1.0V	22320	80	608256	132	4	20
EP4CE22F17...	1.2V	22320	154	608256	132	4	20
EP4CE22F17...	1.2V	22320	154	608256	132	4	20
EP4CE22F17...	1.2V	22320	154	608256	132	4	20
EP4CE22F17	1.2V	22320	154	608256	132	4	20

Companion device

HardCopy:

☐ Limit DSP & RAM to HardCopy device resources

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


New Project Wizard

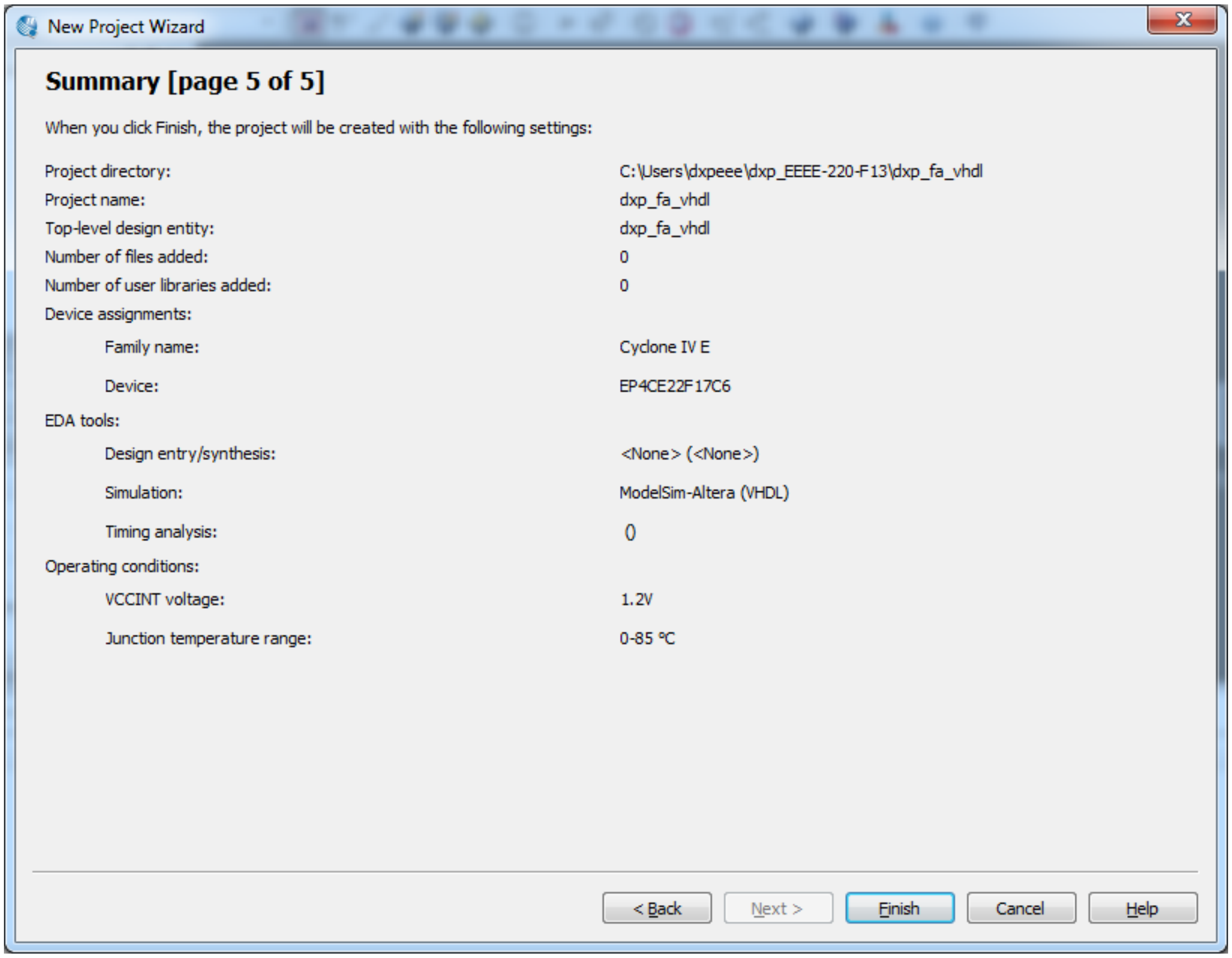
EDA Tool Settings [page 4 of 5]

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>	☐ Run this tool automatically to synthesize the current design
Simulation	ModelSim-Altera	VHDL	☐ Run gate-level simulation automatically after compilation
Formal Verification	<None>		
Board-Level	Timing	<None>	
	Symbol	<None>	
	Signal Integrity	<None>	
	Boundary Scan	<None>	

6) Before you click Finish review the summary report.

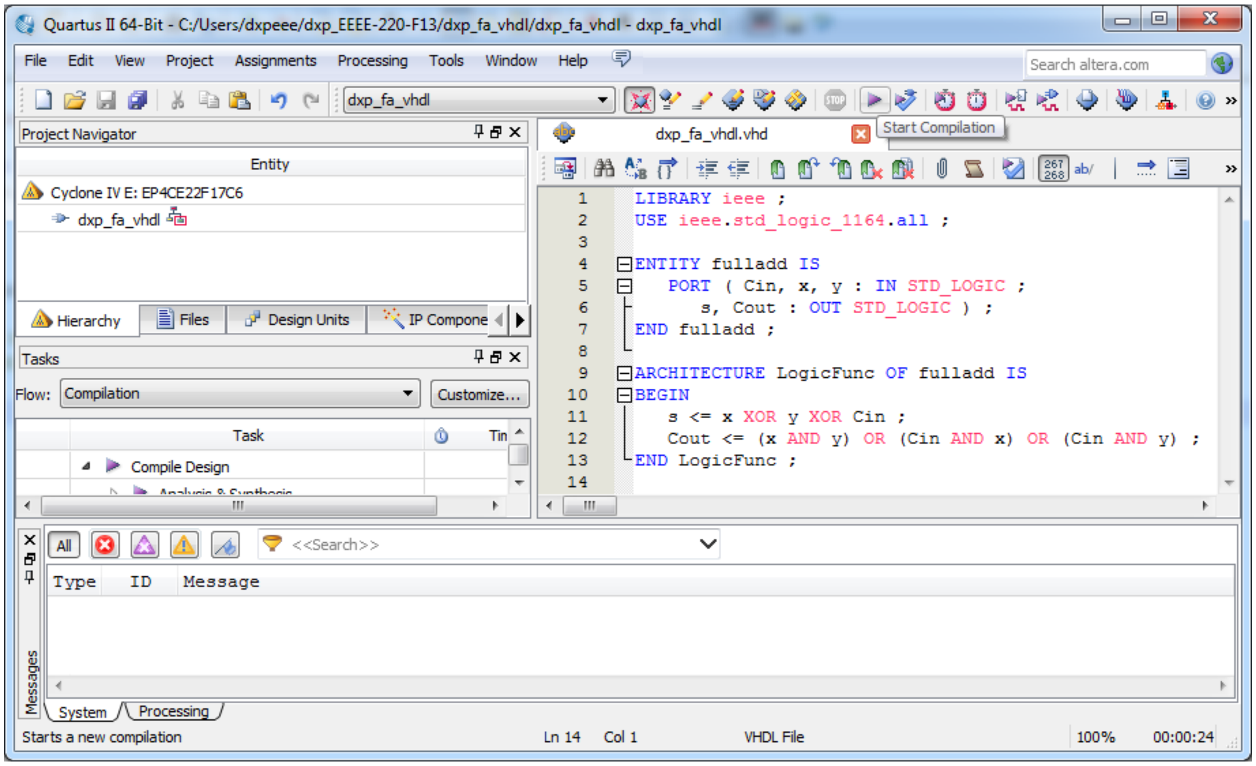


Summary [page 5 of 5]

When you click Finish, the project will be created with the following settings:

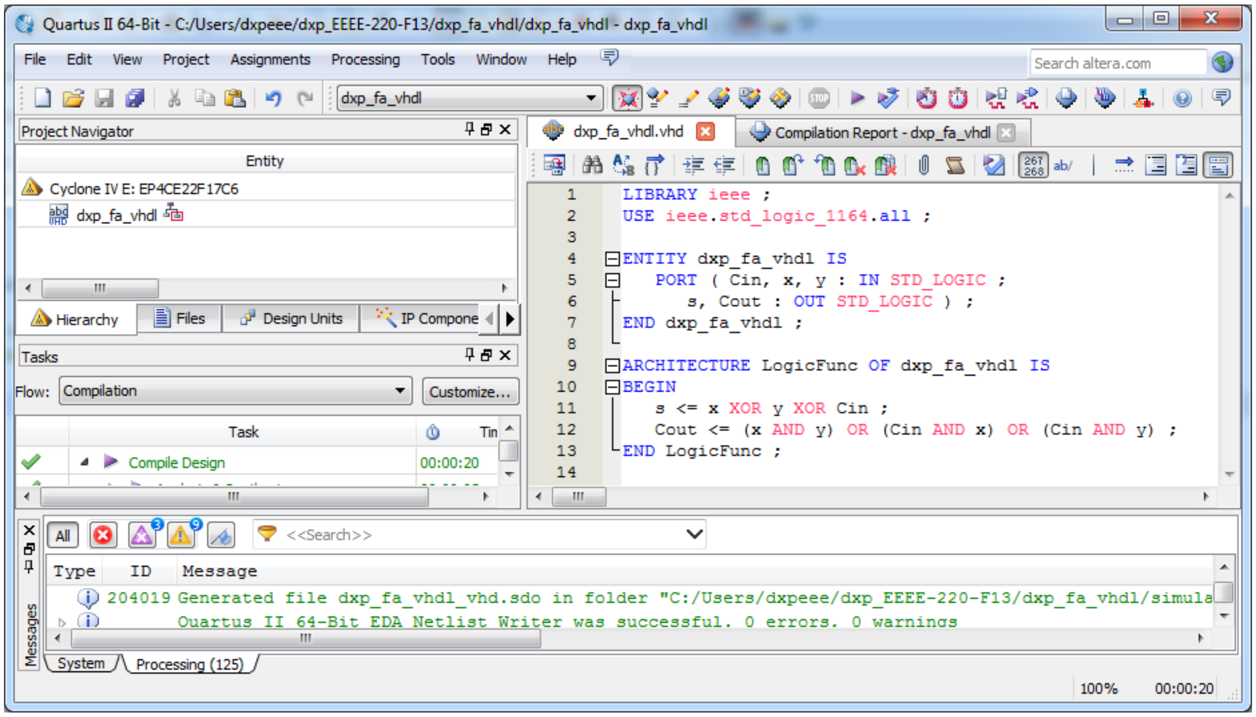
Project directory:	C:\Users\dxpeee\dxp_EEEE-220-F13\dxp_fa_vhdl
Project name:	dxp_fa_vhdl
Top-level design entity:	dxp_fa_vhdl
Number of files added:	0
Number of user libraries added:	0
Device assignments:	
Family name:	Cyclone IV E
Device:	EP4CE22F17C6
EDA tools:	
Design entry/synthesis:	<None> (<None>)
Simulation:	ModelSim-Altera (VHDL)
Timing analysis:	0
Operating conditions:	
VCCINT voltage:	1.2V
Junction temperature range:	0-85 °C

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


Quartus II 64-Bit - C:/Users/dxpeee/dxp_EEEE-220-F13/dxp_fa_vhdl/dxp_fa_vhdl - dxp_fa_vhdl
File Edit View Project Assignments Processing Tools Window Help
Project Navigator
Cyclone IV E: EP4CE22F17C6
dxp_fa_vhdl
Tasks
Flow: Compilation
Compile Design
dxp_fa_vhdl.vhd
Start Compilation
Starts a new compilation

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY fulladd IS
    PORT ( Cin, x, y : IN STD_LOGIC ;
        s, Cout : OUT STD_LOGIC ) ;
END fulladd ;

ARCHITECTURE LogicFunc OF fulladd IS
BEGIN
    s <= x XOR y XOR Cin ;
    Cout <= (x AND y) OR (Cin AND x) OR (Cin AND y) ;
END LogicFunc ;
```

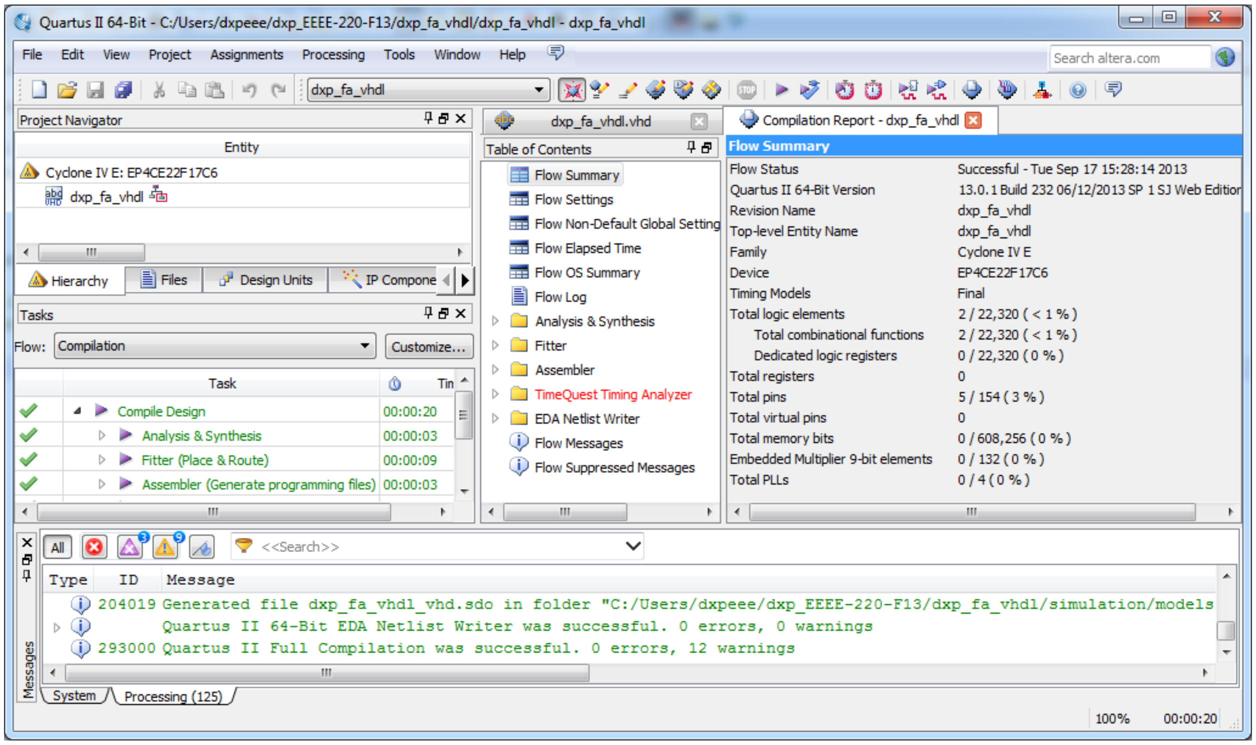
11) **<u>Now Remember</u>: the top-level file and entity names have to be identical!** Therefore, change the entity name inside the code to fml_fa_vhdl. Otherwise, the compiler will give you an error saying that it cannot find the top-level entity.


Quartus II 64-Bit - C:/Users/dxpeee/dxp_EEEE-220-F13/dxp_fa_vhdl/dxp_fa_vhdl - dxp_fa_vhdl
File Edit View Project Assignments Processing Tools Window Help
Project Navigator
Cyclone IV E: EP4CE22F17C6
dxp_fa_vhdl
Tasks
Flow: Compilation
Compile Design 00:00:20
dxp_fa_vhdl.vhd
Compilation Report - dxp_fa_vhdl

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY dxp_fa_vhdl IS
    PORT ( Cin, x, y : IN STD_LOGIC ;
        s, Cout : OUT STD_LOGIC ) ;
END dxp_fa_vhdl ;

ARCHITECTURE LogicFunc OF dxp_fa_vhdl IS
BEGIN
    s <= x XOR y XOR Cin ;
    Cout <= (x AND y) OR (Cin AND x) OR (Cin AND y) ;
END LogicFunc ;
```

```
Type  ID     Message
      204019 Generated file dxp_fa_vhdl_vhd.sdo in folder "C:/Users/dxpeee/dxp_EEEE-220-F13/dxp_fa_vhdl/simula
             Quartus II 64-Bit EDA Netlist Writer was successful. 0 errors, 0 warnings
```

12) Click on the magenta play button to start compilation. If all goes well you should only get some 12 warnings.



Task	Time
✓ Compile Design	00:00:20
✓ Analysis & Synthesis	00:00:03
✓ Fitter (Place & Route)	00:00:09
✓ Assembler (Generate programming files)	00:00:03

Flow Summary

Flow Status	Successful - Tue Sep 17 15:28:14 2013
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Editior
Revision Name	dxp_fa_vhdl
Top-level Entity Name	dxp_fa_vhdl
Family	Cyclone IV E
Device	EP4CE22F17C6
Timing Models	Final
Total logic elements	2 / 22,320 (< 1 %)
Total combinational functions	2 / 22,320 (< 1 %)
Dedicated logic registers	0 / 22,320 (0 %)
Total registers	0
Total pins	5 / 154 (3 %)
Total virtual pins	0
Total memory bits	0 / 608,256 (0 %)
Embedded Multiplier 9-bit elements	0 / 132 (0 %)
Total PLLs	0 / 4 (0 %)

```
Type  ID     Message
      204019 Generated file dxp_fa_vhdl_vhd.sdo in folder "C:/Users/dxpeee/dxp_EEEE-220-F13/dxp_fa_vhdl/simulation/models
               Quartus II 64-Bit EDA Netlist Writer was successful. 0 errors, 0 warnings
      293000 Quartus II Full Compilation was successful. 0 errors, 12 warnings
```

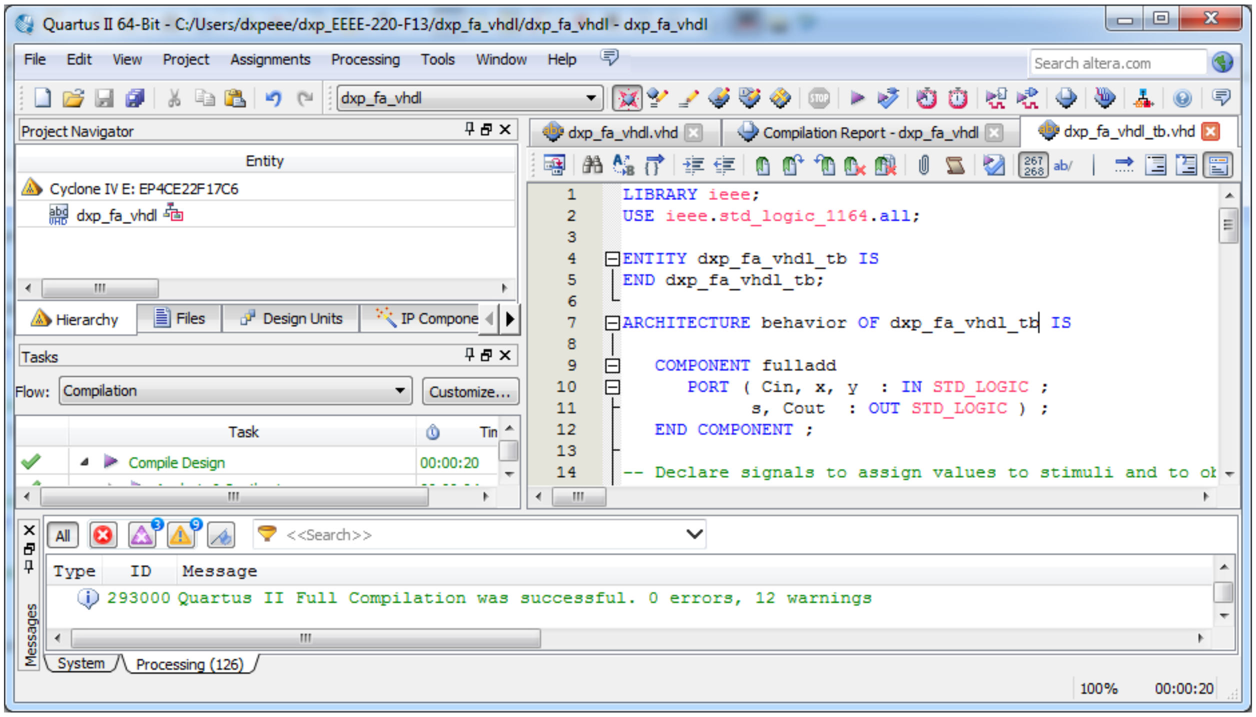
13) This completes the first part of this lab. Don’t close the project.

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 VHDL testbench in the lecture presentation: dxp_chapter3b.ppt.pdf.

15) Go to File > New > VHDL 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_vhdl_tb.vhd.

17) Change the entity name inside the code to fml_fa_vhdl_tb. Save again.

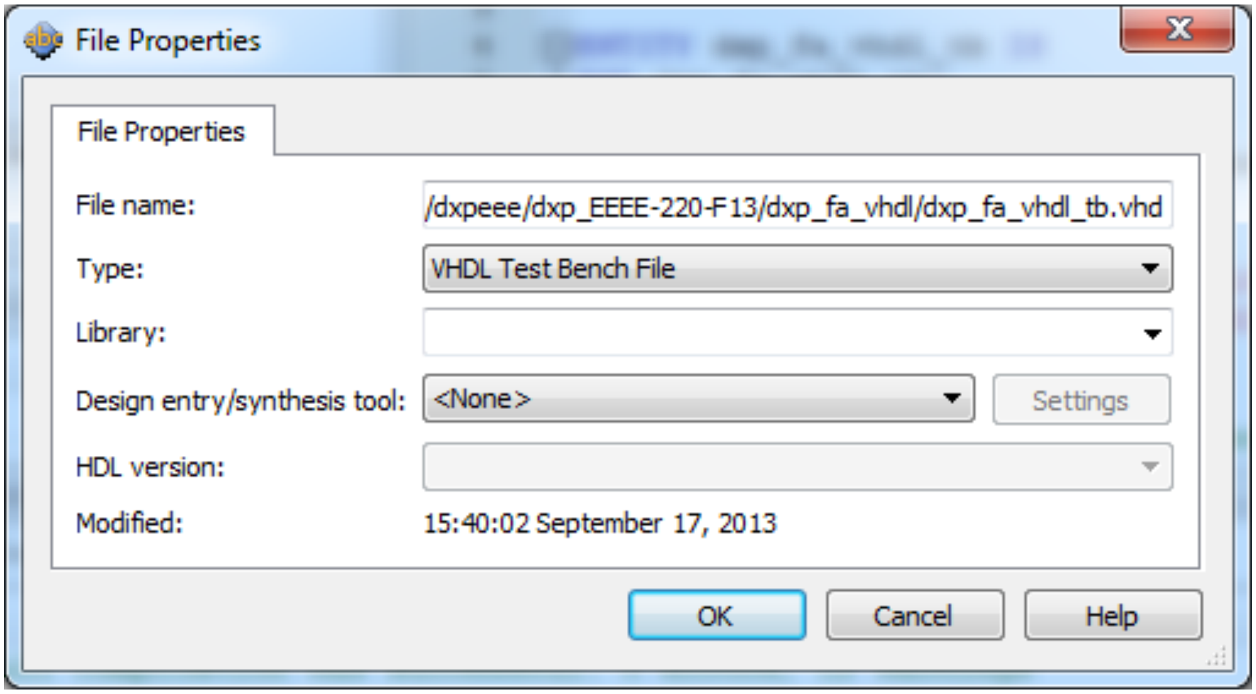


Quartus II 64-Bit - C:/Users/dxpeee/dxp_EEEE-220-F13/dxp_fa_vhdl/dxp_fa_vhdl - dxp_fa_vhdl
File Edit View Project Assignments Processing Tools Window Help
Project Navigator
Entity
Cyclone IV E: EP4CE22F17C6
dxp_fa_vhdl
Hierarchy
Files
Design Units
IP Compone
Tasks
Flow: Compilation
Customize...
Task
Compile Design
00:00:20
dxp_fa_vhdl.vhd
Compilation Report - dxp_fa_vhdl
dxp_fa_vhdl_tb.vhd
System
Processing (126)
100%
00:00:20

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY dxp_fa_vhdl_tb IS
END dxp_fa_vhdl_tb;

ARCHITECTURE behavior OF dxp_fa_vhdl_tb IS

   COMPONENT fulladd
      PORT ( Cin, x, y  : IN STD_LOGIC ;
               s, Cout  : OUT STD_LOGIC ) ;
   END COMPONENT ;

-- Declare signals to assign values to stimuli and to ob
```

```
Type  ID    Message
      293000 Quartus II Full Compilation was successful. 0 errors, 12 warnings
```

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

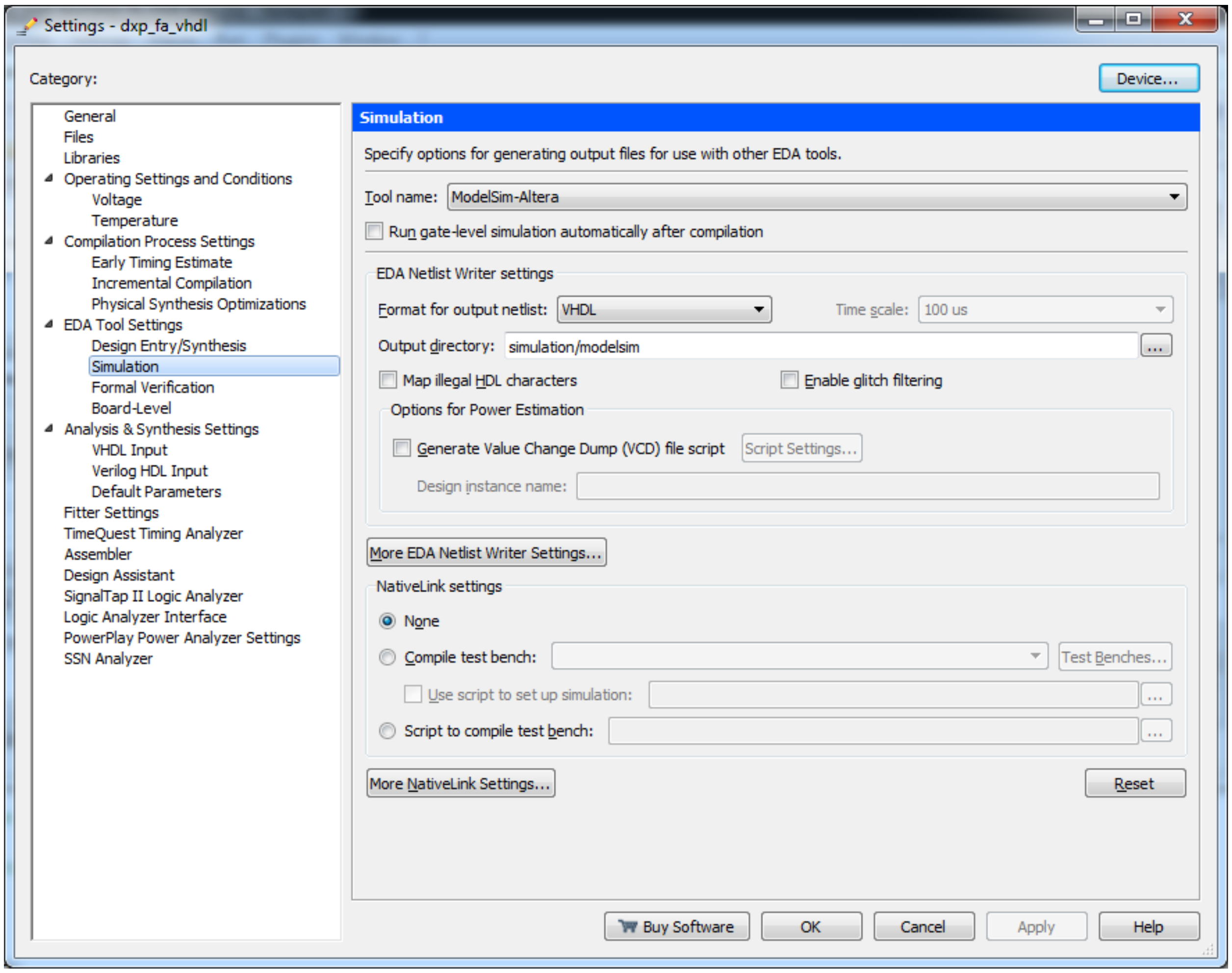


File Properties

File name:	/dxpeee/dxp_EEEE-220-F13/dxp_fa_vhdl/dxp_fa_vhdl_tb.vhd
Type:	VHDL Test Bench File
Library:	
Design entry/synthesis tool:	<None>
HDL version:	
Modified:	15:40:02 September 17, 2013

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.



Settings - dxp_fa_vhdl

Category:

- General
- Files
- Libraries
- Operating Settings and Conditions
  - Voltage
  - Temperature
- Compilation Process Settings
  - Early Timing Estimate
  - Incremental Compilation
  - Physical Synthesis Optimizations
- EDA Tool Settings
  - Design Entry/Synthesis
  - Simulation
  - Formal Verification
  - Board-Level
- Analysis & Synthesis Settings
  - VHDL Input
  - Verilog HDL Input
  - Default Parameters
- Fitter Settings
- TimeQuest Timing Analyzer
- Assembler
- Design Assistant
- SignalTap II Logic Analyzer
- Logic Analyzer Interface
- PowerPlay Power Analyzer Settings
- SSN Analyzer

Simulation

Specify options for generating output files for use with other EDA tools.

Tool name: ModelSim-Altera

☐ Run gate-level simulation automatically after compilation

EDA Netlist Writer settings

Format for output netlist: VHDL Time scale: 100 us

Output directory: simulation/modelsim

☐ Map illegal HDL characters ☐ Enable glitch filtering

Options for Power Estimation

☐ Generate Value Change Dump (VCD) file script

Design instance name:

NativeLink settings

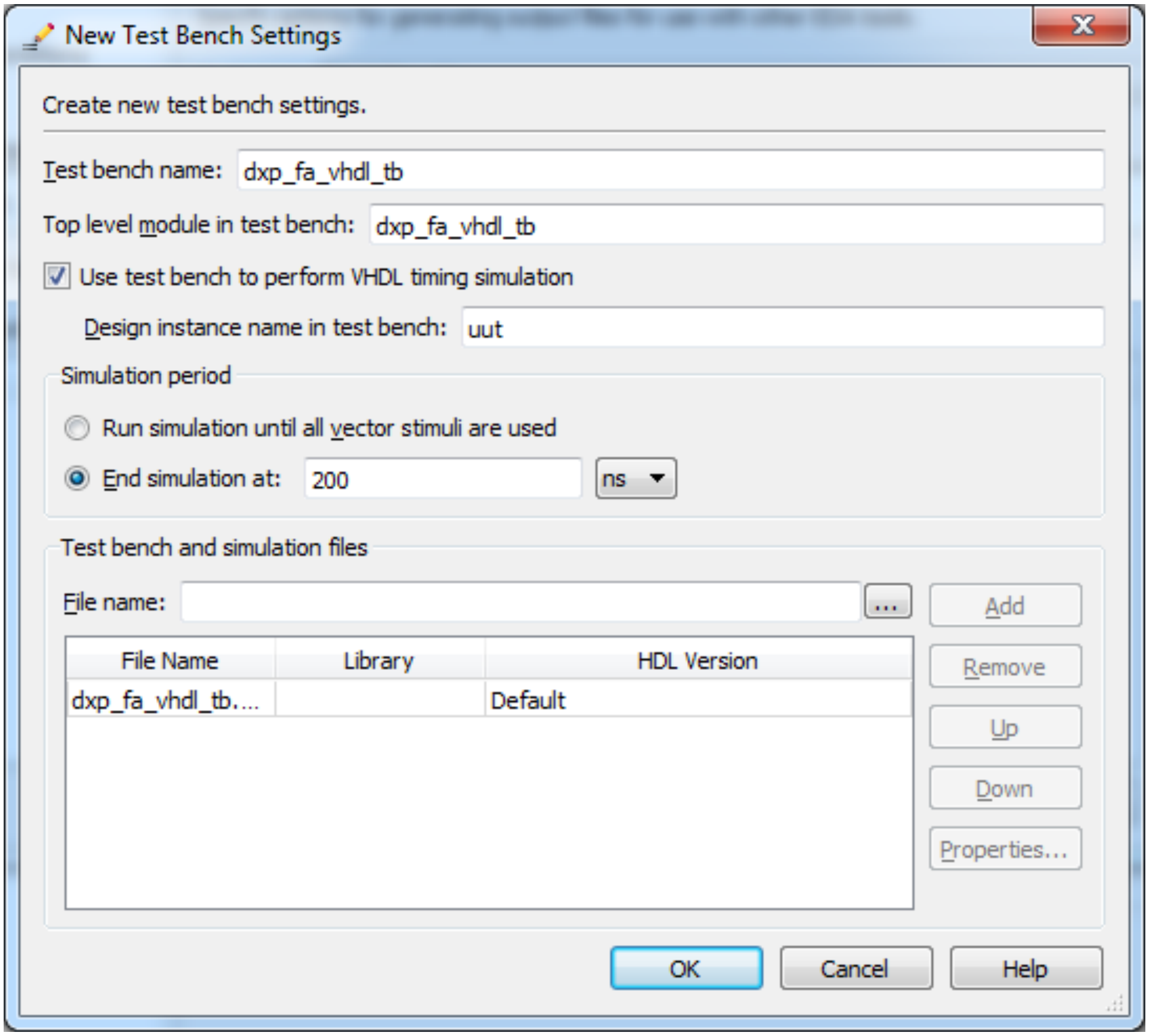
● None

○ Compile test bench:

☐ Use script to set up simulation:

○ Script to compile test bench:

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. Next, select to use the testbench to perform simulation and enter the device instance name in the testbench, which is **uut**.
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:


New Test Bench Settings

Create new test bench settings.

Test bench name: dxp_fa_vhdl_tb

Top level module in test bench: dxp_fa_vhdl_tb

[✓] Use test bench to perform VHDL timing simulation

Design instance name in test bench: uut

Simulation period

( ) Run simulation until all vector stimuli are used

(●) End simulation at: 200 ns

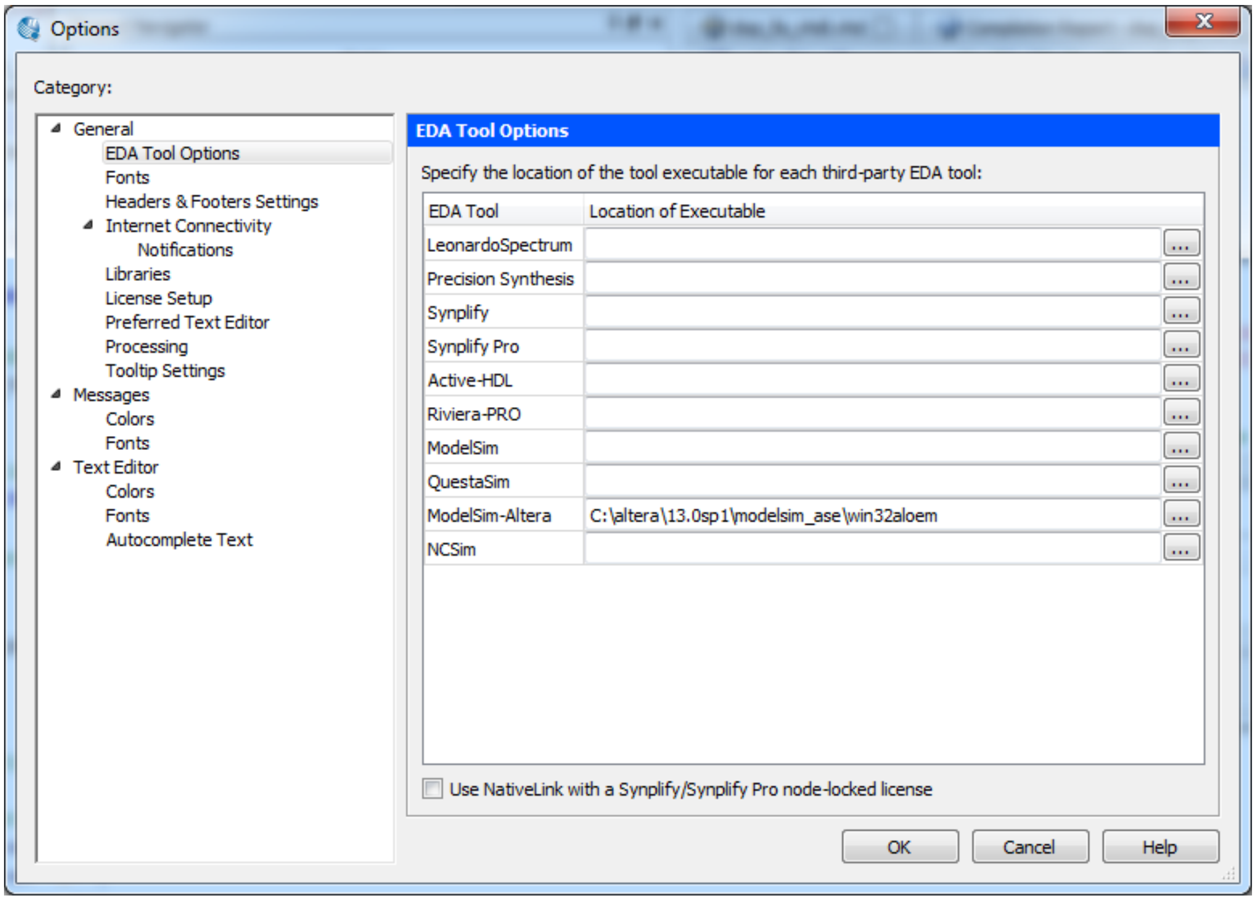
Test bench and simulation files

File name:

File Name	Library	HDL Version
dxp_fa_vhdl_tb....		Default

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


Options

Category:

- General
  - EDA Tool Options
  - Fonts
  - Headers & Footers Settings
  - Internet Connectivity
    - Notifications
  - Libraries
  - License Setup
  - Preferred Text Editor
  - Processing
  - Tooltip Settings
- Messages
  - Colors
  - Fonts
- Text Editor
  - Colors
  - Fonts
  - Autocomplete Text

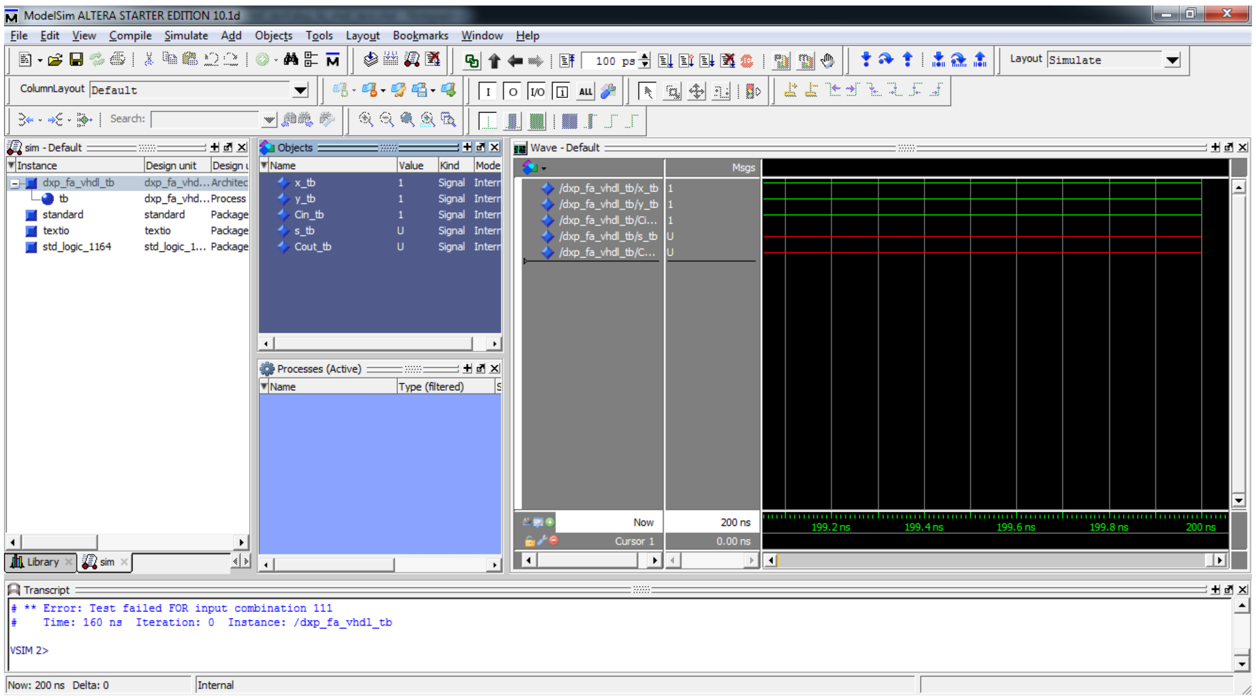
EDA Tool Options

Specify the location of the tool executable for each third-party EDA tool:

EDA Tool	Location of Executable
LeonardoSpectrum	
Precision Synthesis	
Synplify	
Synplify Pro	
Active-HDL	
Riviera-PRO	
ModelSim	
QuestaSim	
ModelSim-Altera	C:\altera\13.0sp1\modelsim_ase\win32aloem
NCSim	

☐ Use NativeLink with a Synplify/Synplify Pro node-locked license

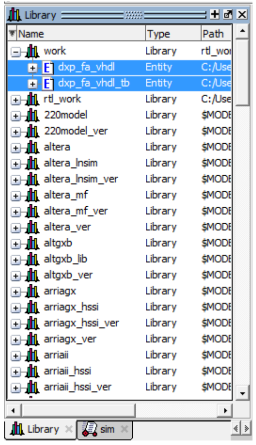
28) Click OK.
29) Compile again.
30) Now open the ModelSIm simulator: Tools > Run Simulation Tools > RTL Simulation. After a few moments ModelSim will start and the main window will look like:



Name	Value	Kind	Mode
x_tb	1	Signal	Intern
y_tb	1	Signal	Intern
Cin_tb	1	Signal	Intern
s_tb	U	Signal	Intern
Cout_tb	U	Signal	Intern

```
# ** Error: Test failed FOR input combination 111
#    Time: 160 ns  Iteration: 0  Instance: /dxp_fa_vhdl_tb

VSIM 2>
```

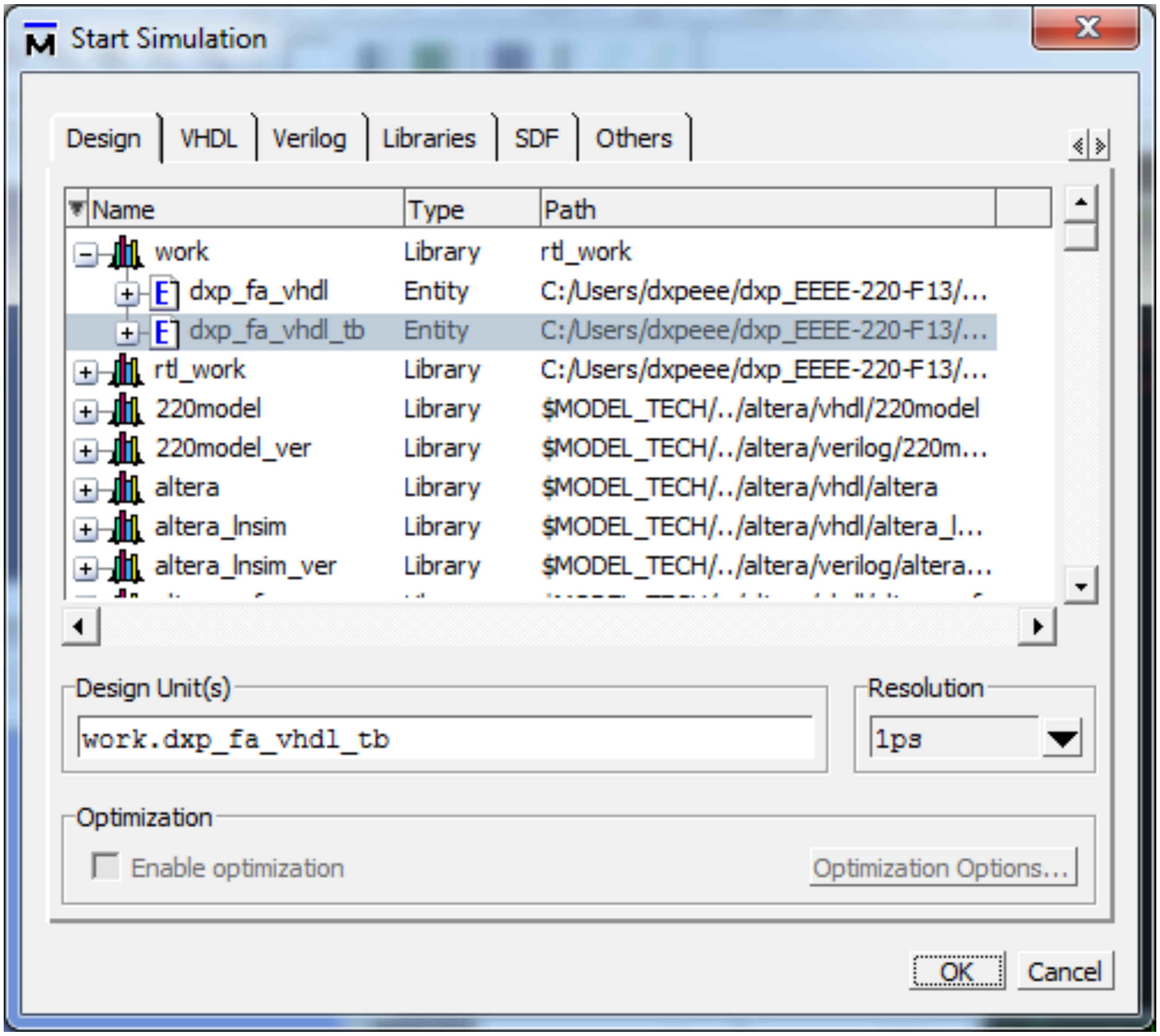
31) Click on the Library tab on the LHS and select the two files in your working directory.



Name	Type	Path
work	Library	rtl_wor
dxp_fa_vhdl	Entity	C:/Use
dxp_fa_vhdl_tb	Entity	C:/Use
rtl_work	Library	C:/Use
220model	Library	$MODE
220model_ver	Library	$MODE
altera	Library	$MODE
altera_lnsim	Library	$MODE
altera_lnsim_ver	Library	$MODE
altera_mf	Library	$MODE
altera_mf_ver	Library	$MODE
altera_ver	Library	$MODE
altgxb	Library	$MODE
altgxb_lib	Library	$MODE
altgxb_ver	Library	$MODE
arriagx	Library	$MODE
arriagx_hssi	Library	$MODE
arriagx_hssi_ver	Library	$MODE
arriagx_ver	Library	$MODE
arriaii	Library	$MODE
arriaii_hssi	Library	$MODE
arriaii_hssi_ver	Library	$MODE

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.



Start Simulation

Name	Type	Path
work	Library	rtl_work
dxp_fa_vhdl	Entity	C:/Users/dxpeee/dxp_EEEE-220-F13/...
dxp_fa_vhdl_tb	Entity	C:/Users/dxpeee/dxp_EEEE-220-F13/...
rtl_work	Library	C:/Users/dxpeee/dxp_EEEE-220-F13/...
220model	Library	$MODEL_TECH/../altera/vhdl/220model
220model_ver	Library	$MODEL_TECH/../altera/verilog/220m...
altera	Library	$MODEL_TECH/../altera/vhdl/altera
altera_lnsim	Library	$MODEL_TECH/../altera/vhdl/altera_l...
altera_lnsim_ver	Library	$MODEL_TECH/../altera/verilog/altera...

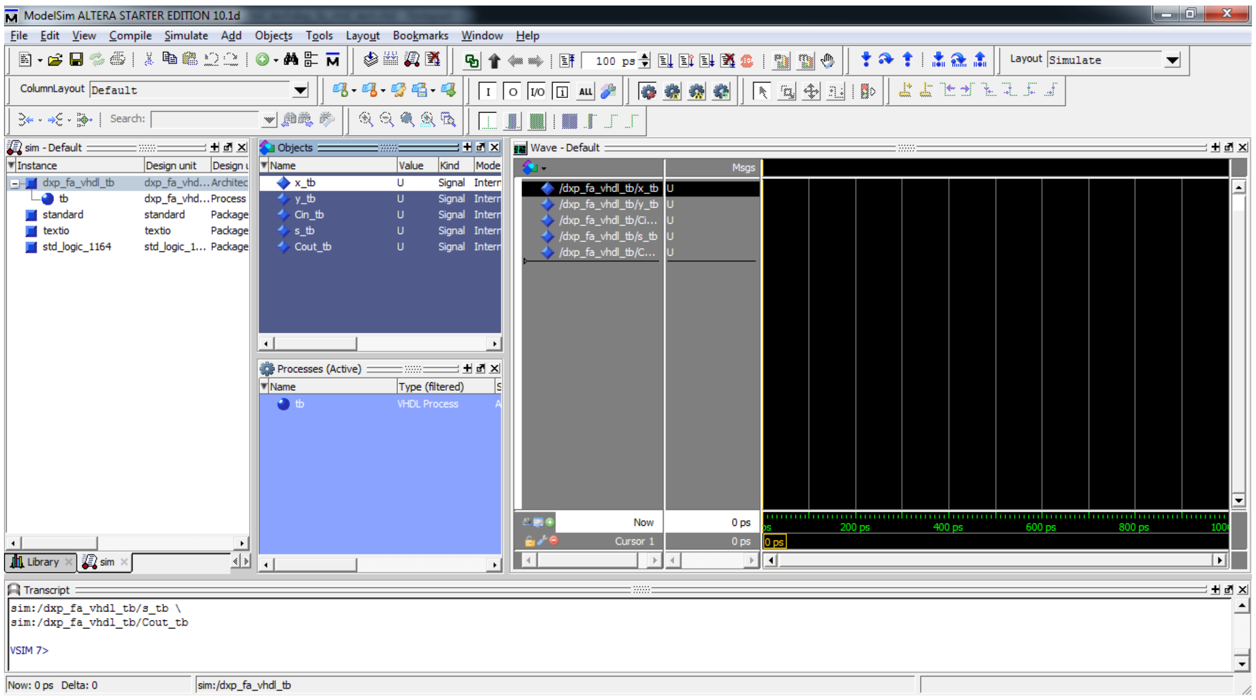
Design Unit(s): work.dxp_fa_vhdl_tb

Resolution: 1ps

Optimization

Enable optimization

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:



Instance	Design unit	Design unit type
dxp_fa_vhdl_tb	dxp_fa_vhd…	Architec
tb	dxp_fa_vhd…	Process
standard	standard	Package
textio	textio	Package
std_logic_1164	std_logic_1…	Package

Name	Value	Kind	Mode
x_tb	U	Signal	Intern
y_tb	U	Signal	Intern
Cin_tb	U	Signal	Intern
s_tb	U	Signal	Intern
Cout_tb	U	Signal	Intern

Name	Type (filtered)
tb	VHDL Process

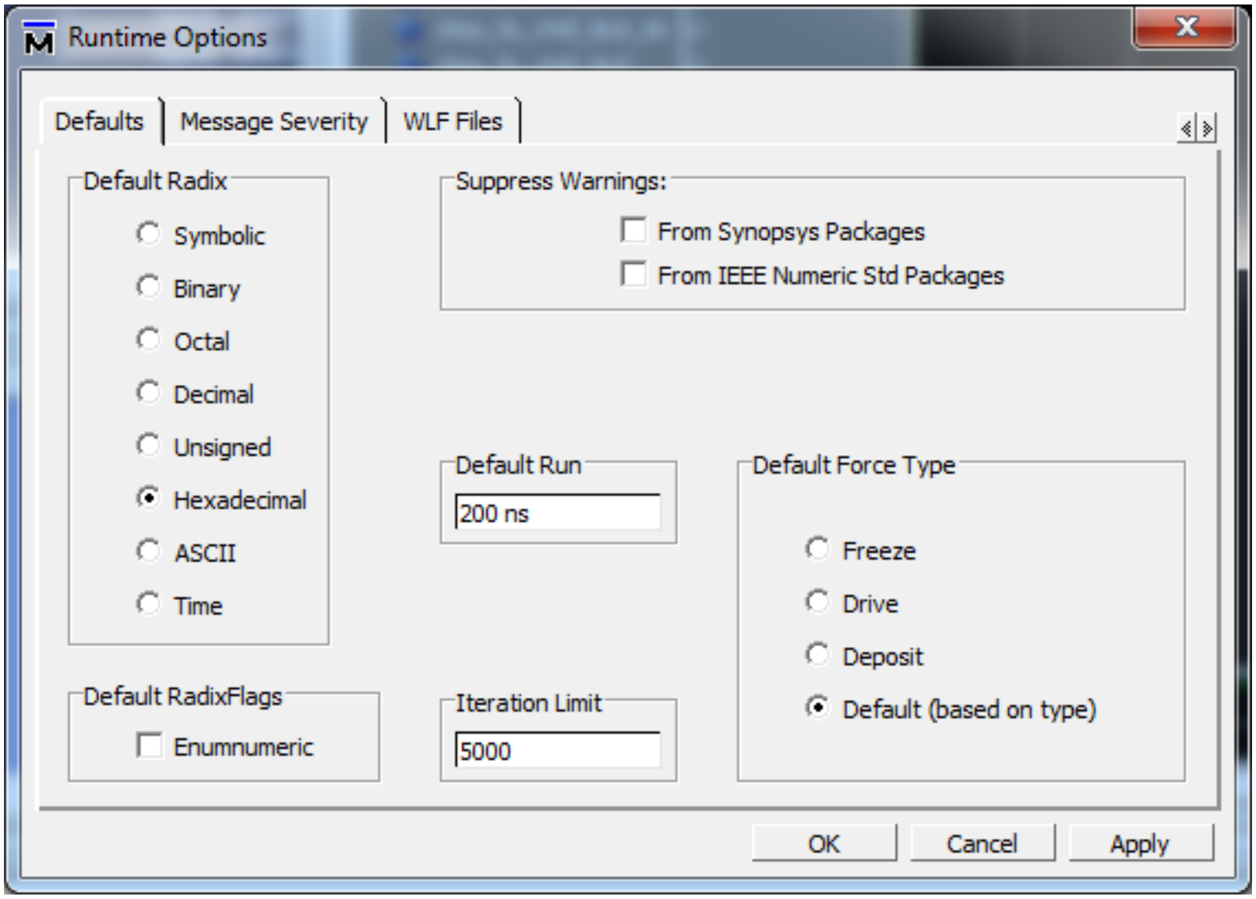
Signal	Msgs
/dxp_fa_vhdl_tb/x_tb	U
/dxp_fa_vhdl_tb/y_tb	U
/dxp_fa_vhdl_tb/Ci…	U
/dxp_fa_vhdl_tb/s_tb	U
/dxp_fa_vhdl_tb/C…	U

```
sim:/dxp_fa_vhdl_tb/s_tb \
sim:/dxp_fa_vhdl_tb/Cout_tb

VSIM 7>
```

```
Now: 0 ps  Delta: 0        sim:/dxp_fa_vhdl_tb
```

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:


Runtime Options

Defaults | Message Severity | WLF Files

Default Radix: Symbolic, Binary, Octal, Decimal, Unsigned, [✓ Hexadecimal], ASCII, Time

Suppress Warnings:
- [ ] From Synopsys Packages
- [ ] From IEEE Numeric Std Packages

Default Run: 200 ns

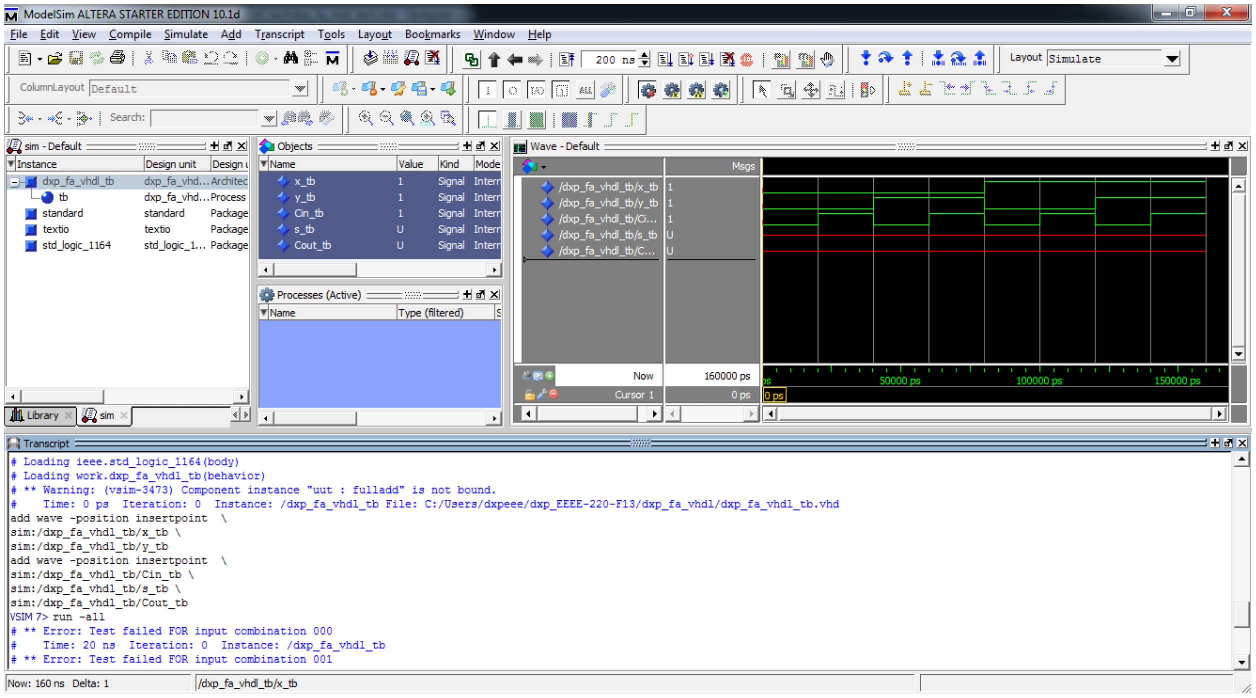
Default Force Type: Freeze, Drive, Deposit, [✓ Default (based on type)]

Default RadixFlags:
- [ ] Enumnumeric

Iteration Limit: 5000

OK | Cancel | Apply

37) Now run the simulation by clicking Simulate > Run > Run-All. Once the simulation finishes, Zoom Out in the waveform window if necessary.

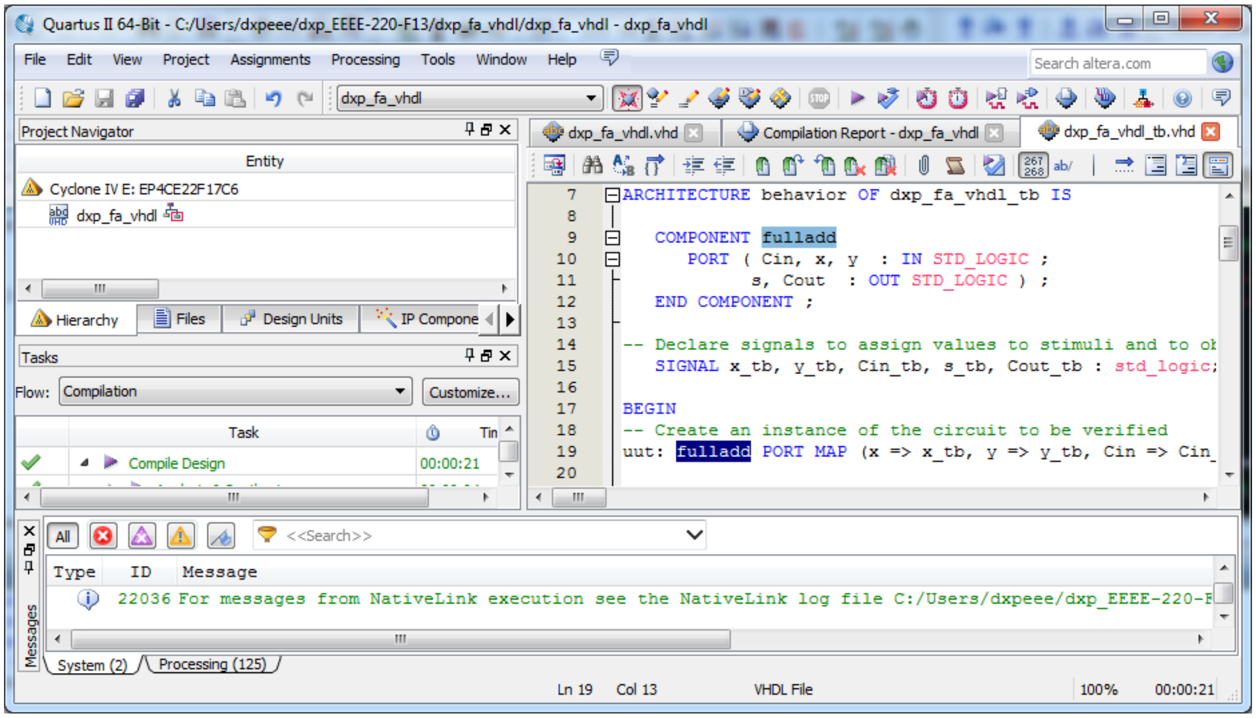

ModelSim ALTERA STARTER EDITION 10.1d

Name	Value	Kind	Mode
x_tb	1	Signal	Intern
y_tb	1	Signal	Intern
Cin_tb	1	Signal	Intern
s_tb	U	Signal	Intern
Cout_tb	U	Signal	Intern

```
# Loading ieee.std_logic_1164(body)
# Loading work.dxp_fa_vhdl_tb(behavior)
# ** Warning: (vsim-3473) Component instance "uut : fulladd" is not bound.
#    Time: 0 ps  Iteration: 0  Instance: /dxp_fa_vhdl_tb File: C:/Users/dxpeee/dxp_EEEE-220-F13/dxp_fa_vhdl/dxp_fa_vhdl_tb.vhd
add wave -position insertpoint  \
sim:/dxp_fa_vhdl_tb/x_tb \
sim:/dxp_fa_vhdl_tb/y_tb
add wave -position insertpoint  \
sim:/dxp_fa_vhdl_tb/Cin_tb \
sim:/dxp_fa_vhdl_tb/s_tb \
sim:/dxp_fa_vhdl_tb/Cout_tb
VSIM 7> run -all
# ** Error: Test failed FOR input combination 000
#    Time: 20 ns  Iteration: 0  Instance: /dxp_fa_vhdl_tb
# ** Error: Test failed FOR input combination 001
```

Now: 160 ns Delta: 1   /dxp_fa_vhdl_tb/x_tb

38) If you ignored the Warning before "fulladd not bound" you will get an error like I did above. The issue is that I didn't change/update the name of the uut in the testbench. Below is the problem highlighted.


Quartus II 64-Bit - C:/Users/dxpeee/dxp_EEEE-220-F13/dxp_fa_vhdl/dxp_fa_vhdl - dxp_fa_vhdl
File Edit View Project Assignments Processing Tools Window Help
Search altera.com
dxp_fa_vhdl
Project Navigator
Entity
Cyclone IV E: EP4CE22F17C6
dxp_fa_vhdl
Hierarchy
Files
Design Units
IP Compone
Tasks
Flow: Compilation
Customize...
Task
Compile Design
00:00:21
dxp_fa_vhdl.vhd
Compilation Report - dxp_fa_vhdl
dxp_fa_vhdl_tb.vhd
All
<<Search>>
System (2)
Processing (125)
Messages
Ln 19 Col 13 VHDL File 100% 00:00:21

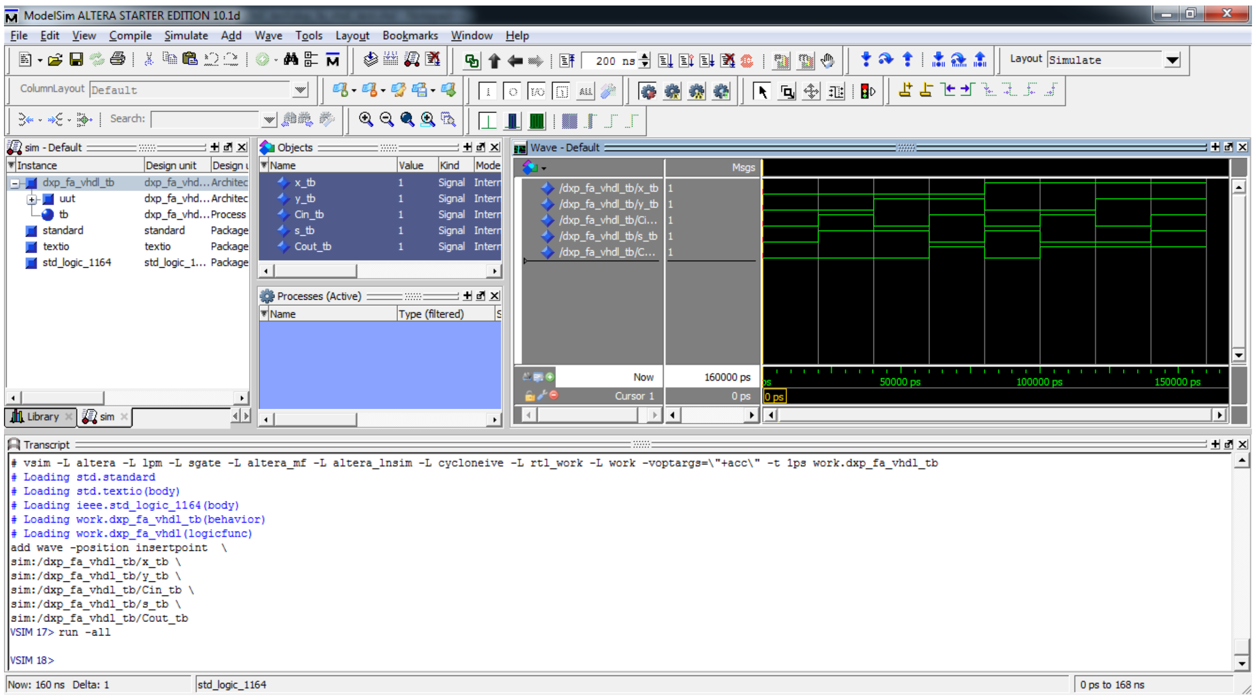
```
ARCHITECTURE behavior OF dxp_fa_vhdl_tb IS

   COMPONENT fulladd
      PORT ( Cin, x, y   : IN STD_LOGIC ;
             s, Cout   : OUT STD_LOGIC ) ;
   END COMPONENT ;

-- Declare signals to assign values to stimuli and to ob
   SIGNAL x_tb, y_tb, Cin_tb, s_tb, Cout_tb : std_logic;

BEGIN
-- Create an instance of the circuit to be verified
uut: fulladd PORT MAP (x => x_tb, y => y_tb, Cin => Cin_
```

```
Type  ID    Message
      22036 For messages from NativeLink execution see the NativeLink log file C:/Users/dxpeee/dxp_EEEE-220-F
```

39) If you didn't either, go back to the main Quartus window and fix it. Change the *fulladd* name to *fml_fa_vhdl*. You'll need to recompile the two files in ModelSim. Click on the Library tab, select both working files, right-click and Recompile.

40) After you recompile you need to place ModelSim in simulator mode again as described from step 33 on above.

41) Now simulate again. 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.

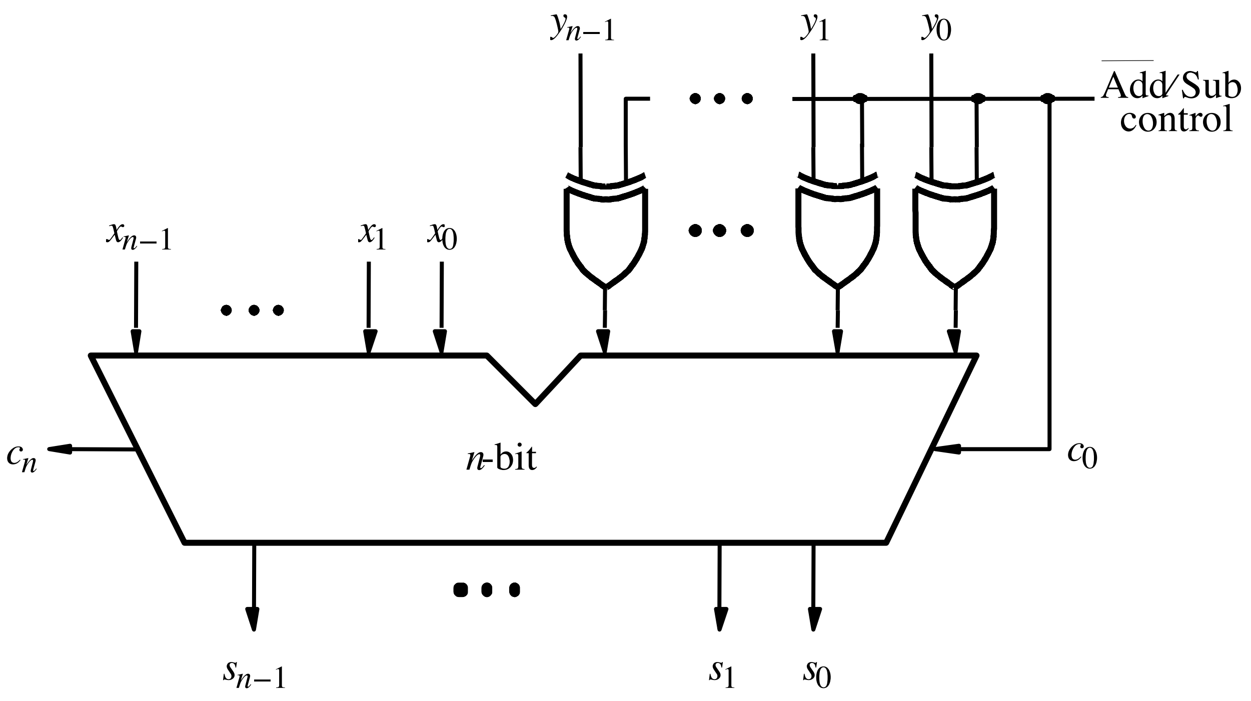


```
# vsim -L altera -L lpm -L sgate -L altera_mf -L altera_lnsim -L cycloneive -L rtl_work -L work -voptargs=\"+acc\" -t 1ps work.dxp_fa_vhdl_tb
# Loading std.standard
# Loading std.textio(body)
# Loading ieee.std_logic_1164(body)
# Loading work.dxp_fa_vhdl_tb(behavior)
# Loading work.dxp_fa_vhdl(logicfunc)
add wave -position insertpoint  \
sim:/dxp_fa_vhdl_tb/x_tb \
sim:/dxp_fa_vhdl_tb/y_tb \
sim:/dxp_fa_vhdl_tb/Cin_tb \
sim:/dxp_fa_vhdl_tb/s_tb \
sim:/dxp_fa_vhdl_tb/Cout_tb
VSIM 17> run -all

VSIM 18>
```

Now: 160 ns Delta: 1 | std_logic_1164 | 0 ps to 168 ns

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

43) In the third part of this week's lab you'll have to create a project and create a VHDL description of a **4-bit ripple carry adder/subtractor**. You can use any of the template examples in the lecture presentation: dxp_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.



44) 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.

45) 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.

46) 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

47) 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.

48) This concludes this week's lab.

49) Grading:
   a. 10 points for full adder VHDL compilation
   b. 10 points for full adder VHDL testbench simulation
   c. 10 points for 4-bit ripple carry adder/subtractor compilation
   d. 10 points for 4-bit ripple carry adder/subtractor simulation.