Digital Systems

Prof. James Clark

ECSE 325

Lab #1: Introduction to Synchronous Circuit Design Flow with Quartus II FPGA Software

Winter 2018

Introduction

In this lab, you will learn the basics of Quartus-II FPGA software, how to compile synchronous circuit VHDL declarations to a target FPGA. The goal of this lab is to see how the fitter of the Quartus compiler maps the design onto the FPGA hardware, through a step-by-step tutorial.

Learning Outcomes

After completing this lab you should know how to:

- Startup the Altera Quartus II software
- Create the framework for a new project
- Write a VHDL description of a logic circuit
- Understand logic utilization within the FPGA board
- Understand the floorplan and RTL viewer of a circuit

Outline of the Lab Series

The lab portion of the course consists of 5 parts. The bulk of the work will be done using the Altera Quartus II software and ModelSim software running on the computers in the lab. Throughout the 5 lab experiments, you will develop all of the building blocks for the system, and integrate them into a complete user-friendly system, using an FPGA development board.

Each lab will have a number of items that the lab groups must demonstrate to a TA. The TA will then sign the appropriate entry on the lab grade sheet (the grade sheet can be found at the end of the lab description, and should be printed out). Items that needed to be demonstrated to the TAs are indicated in the lab description with the pencil-paper-checkmark icon. Once you have obtained all the necessary signatures on the grade-sheet, give it to one of the TAs, who will then pass it on to the course instructor for recording the grade.

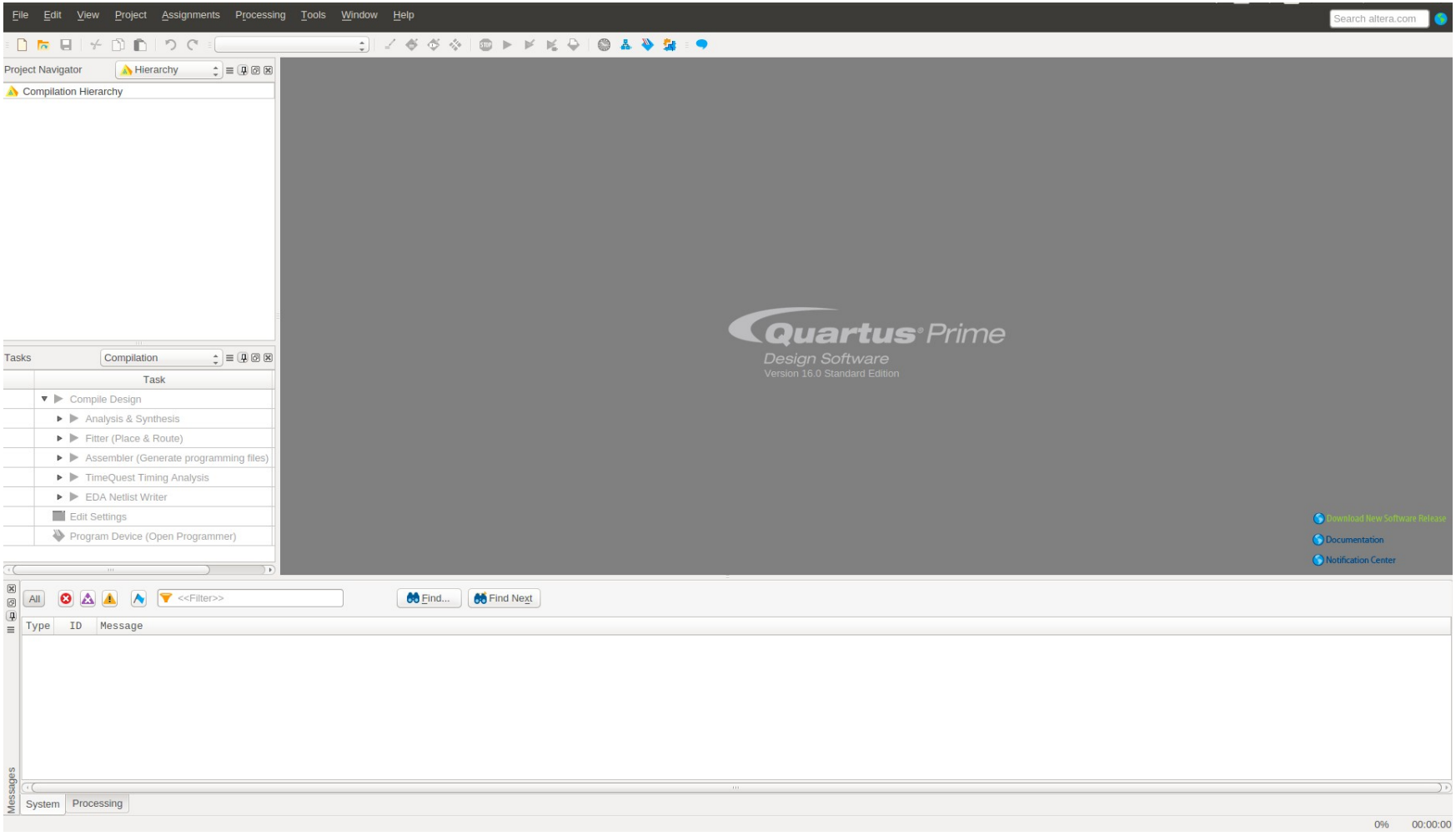
Startup and Creation of Project Framework

In this course you will be using commercial FPGA design software, the **Altera Quartus II** program and the **Mentor Graphics ModelSim** simulation program.

The Quartus II and ModelSim programs are installed on the computers in the lab. You can also obtain a slightly restricted version from the Altera web site. The program restrictions will not affect any designs you will be doing in the course, so you can install the program on your personal computer, so that you can work on your project outside of the lab. You should use version 16.1 of the program, as this is the latest version that supports the prototyping board (the DE1-SoC) that you will be using in the lab.

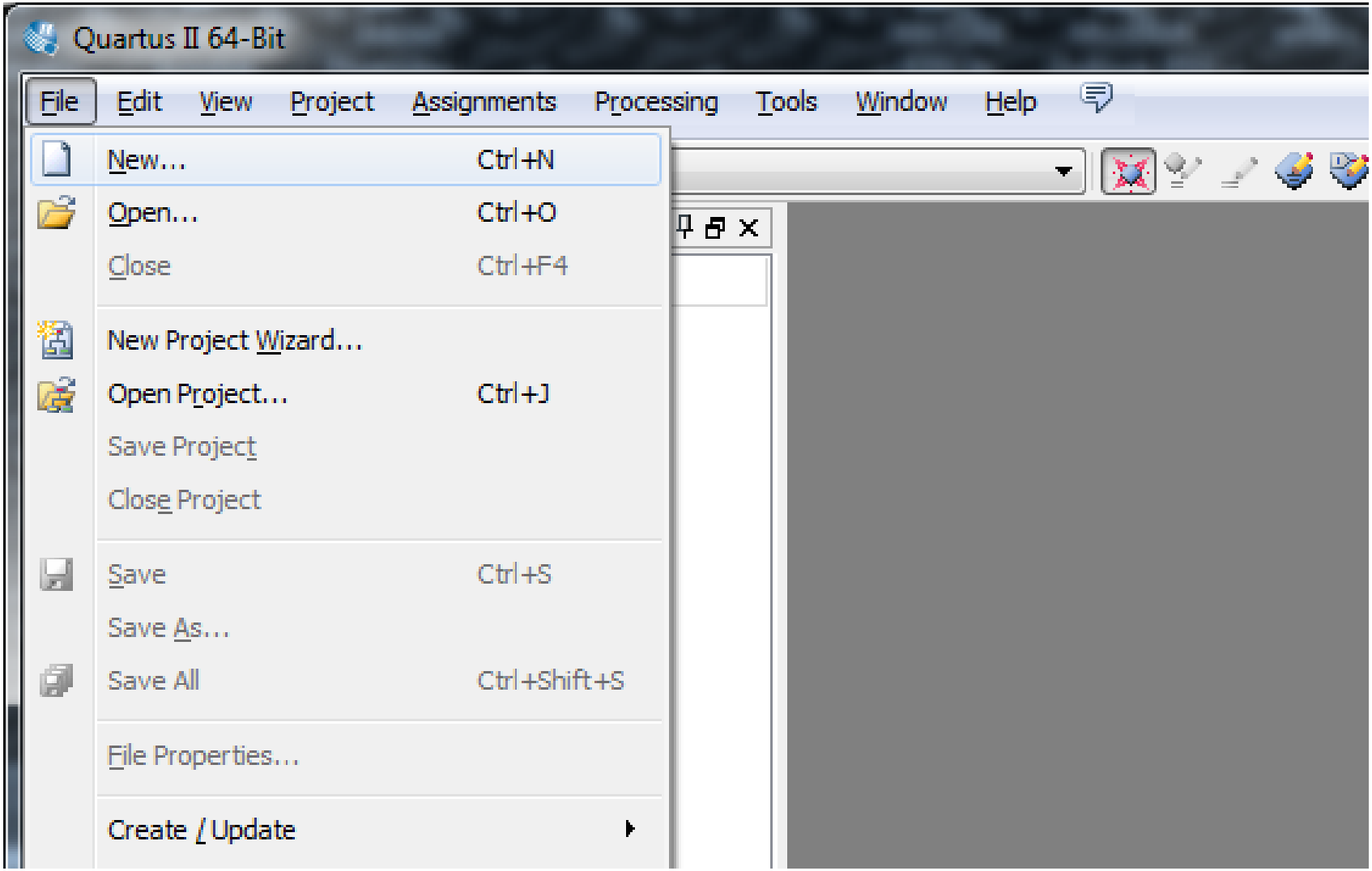
To begin, start the Quartus II program by selecting the program in the Windows Start menu:

This is the window that appears on startup (this shows version 16.1 downloaded from Altera's web site the versions on the lab computers may look slightly different)

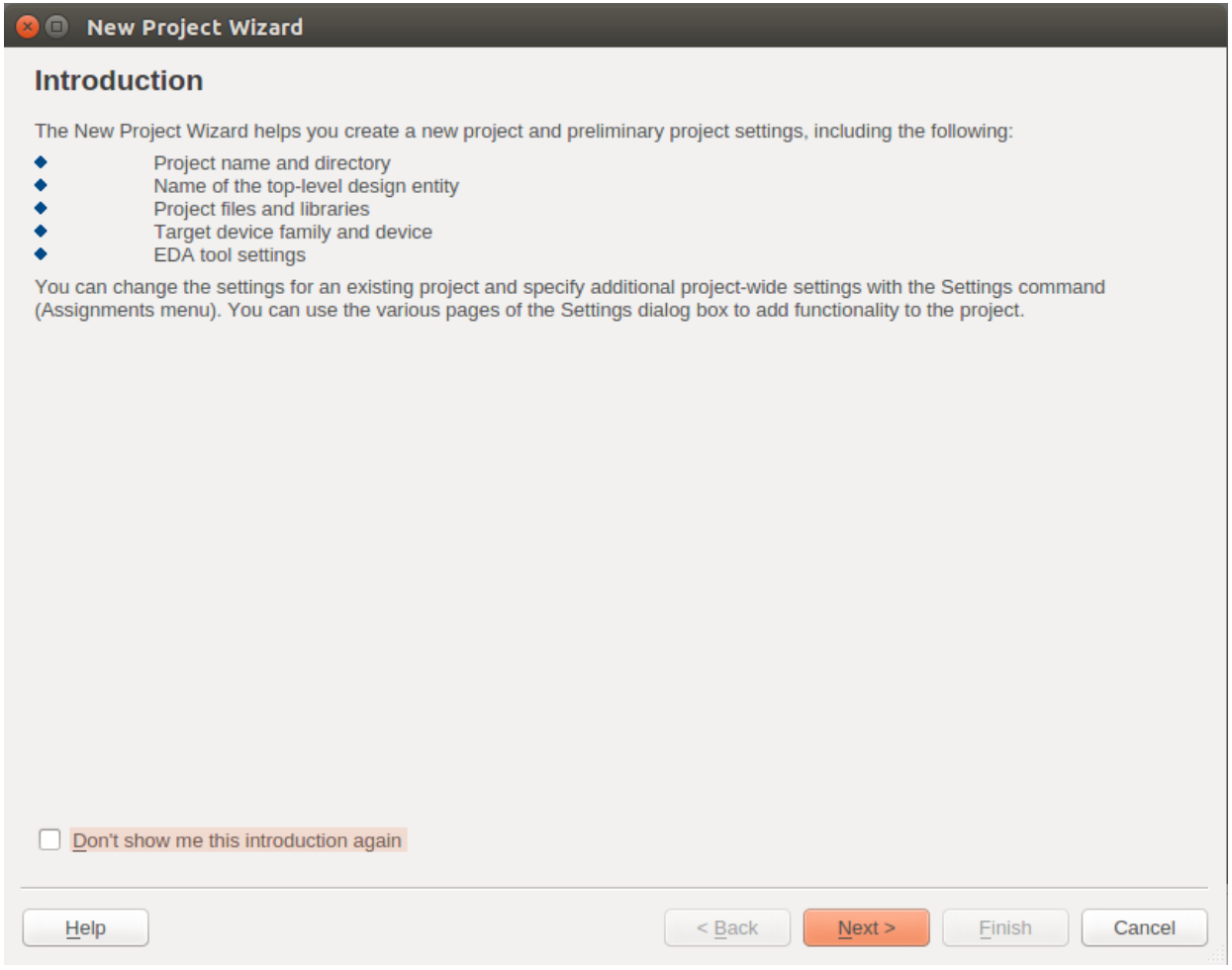


The Altera Quartus II program employs a **project-based approach**. The goal of a Quartus project is to develop a hardware implementation of a specific function, targeted to an FPGA (Field Programmable Gate Array) device.
Typically, the project will involve a (large) number of different circuits, each designed individually, or taken from circuit libraries. **Project management is therefore important.** The Quartus II program aids in the project management by providing a project framework, that keeps track of the various components of the project, including design files (such as schematic block diagrams or VHDL descriptions), simulation files, compilation reports, FPGA configuration or programming files, project specific program settings and assignments, and many others.
The first step in designing a system using the Quartus II approach is therefore to create the project framework. The program simplifies this by providing a "Wizard" which guides you through a step-by-step setting of the most important options.

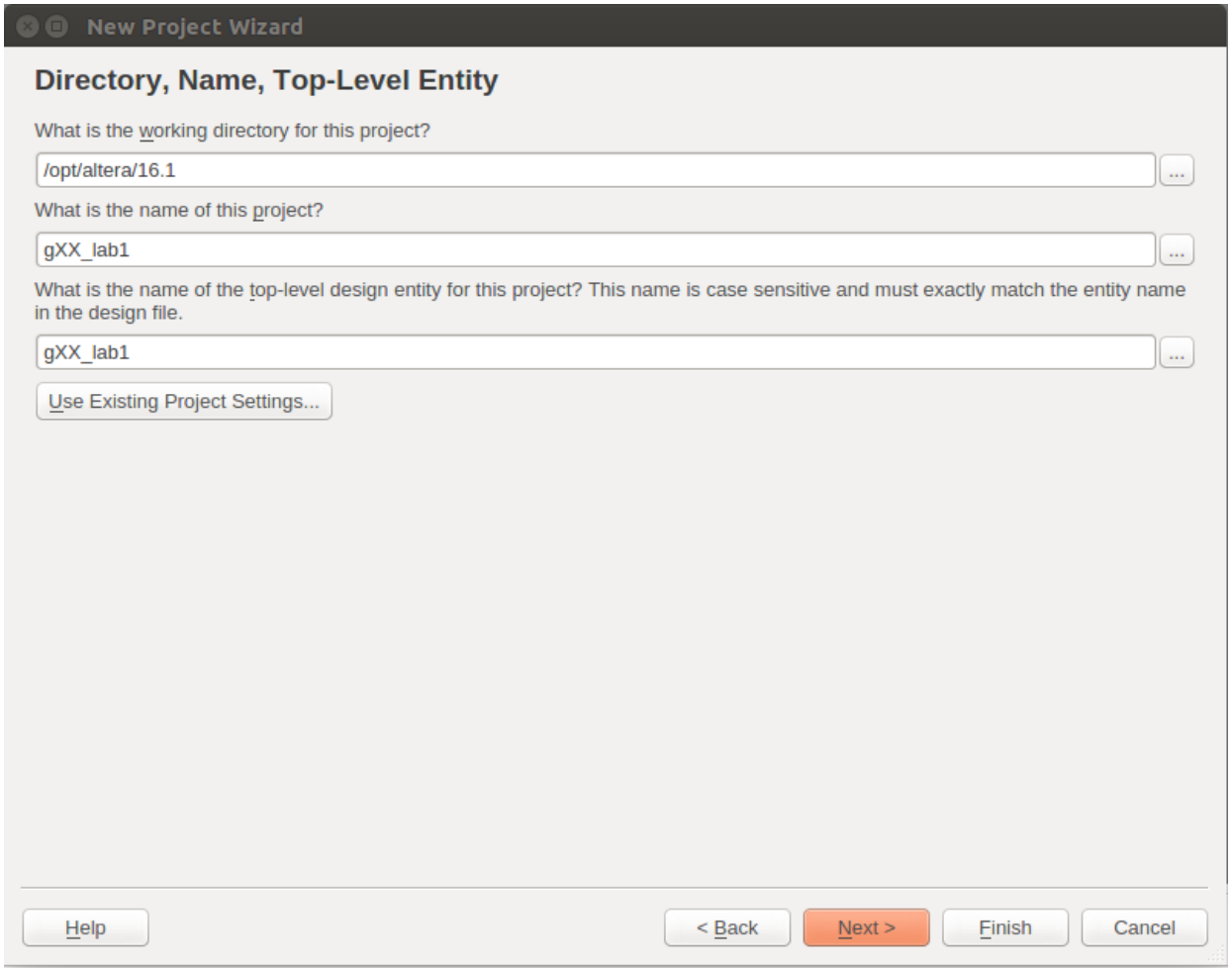
To run the Project Wizard, click on the **File** menu and select the **New Project Wizard** entry.



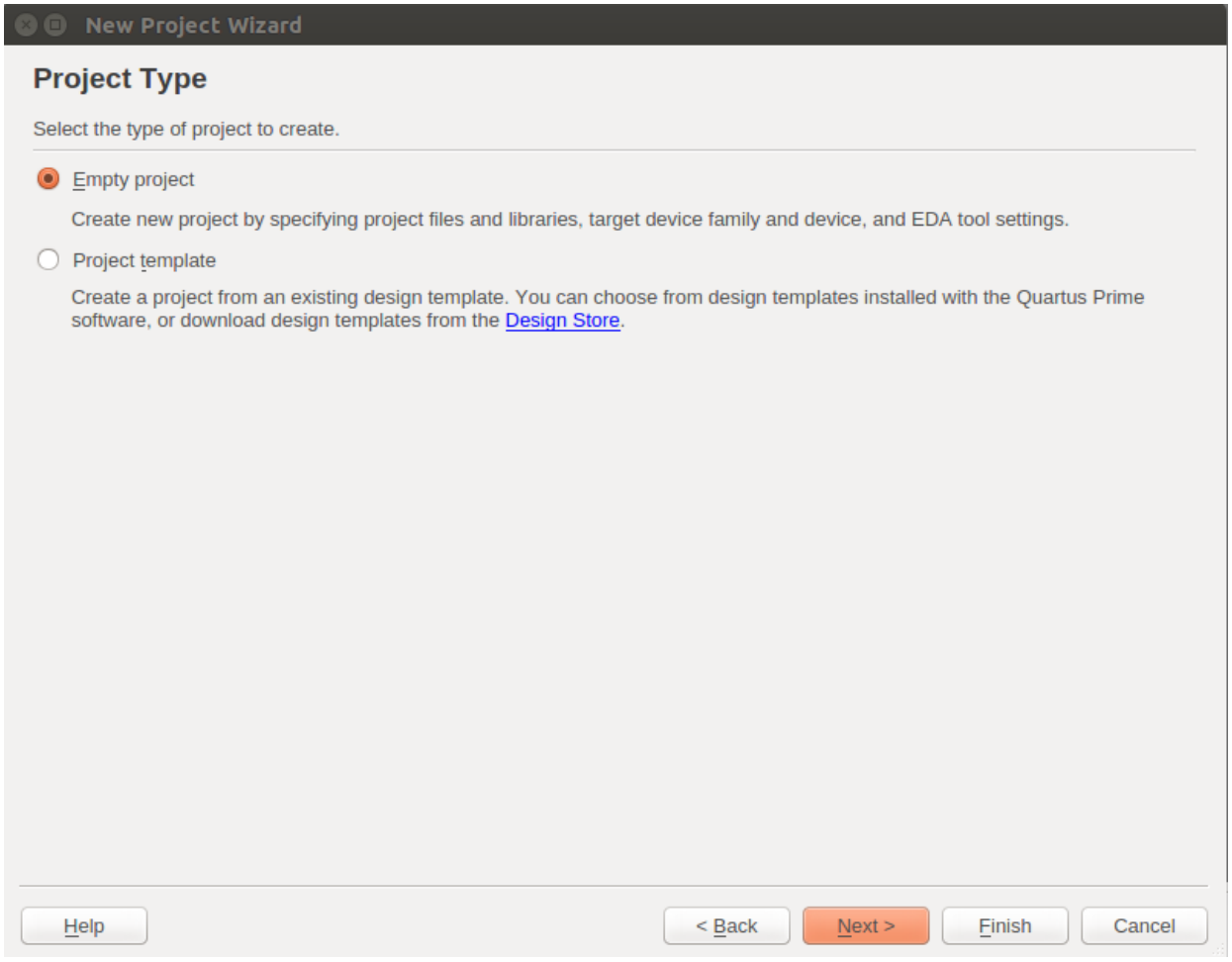
The New Project Wizard involves going through a series of windows. The first window, shown at right, is an introduction, listing the settings that can be applied. After reading the text on this window, click on "Next" to proceed.



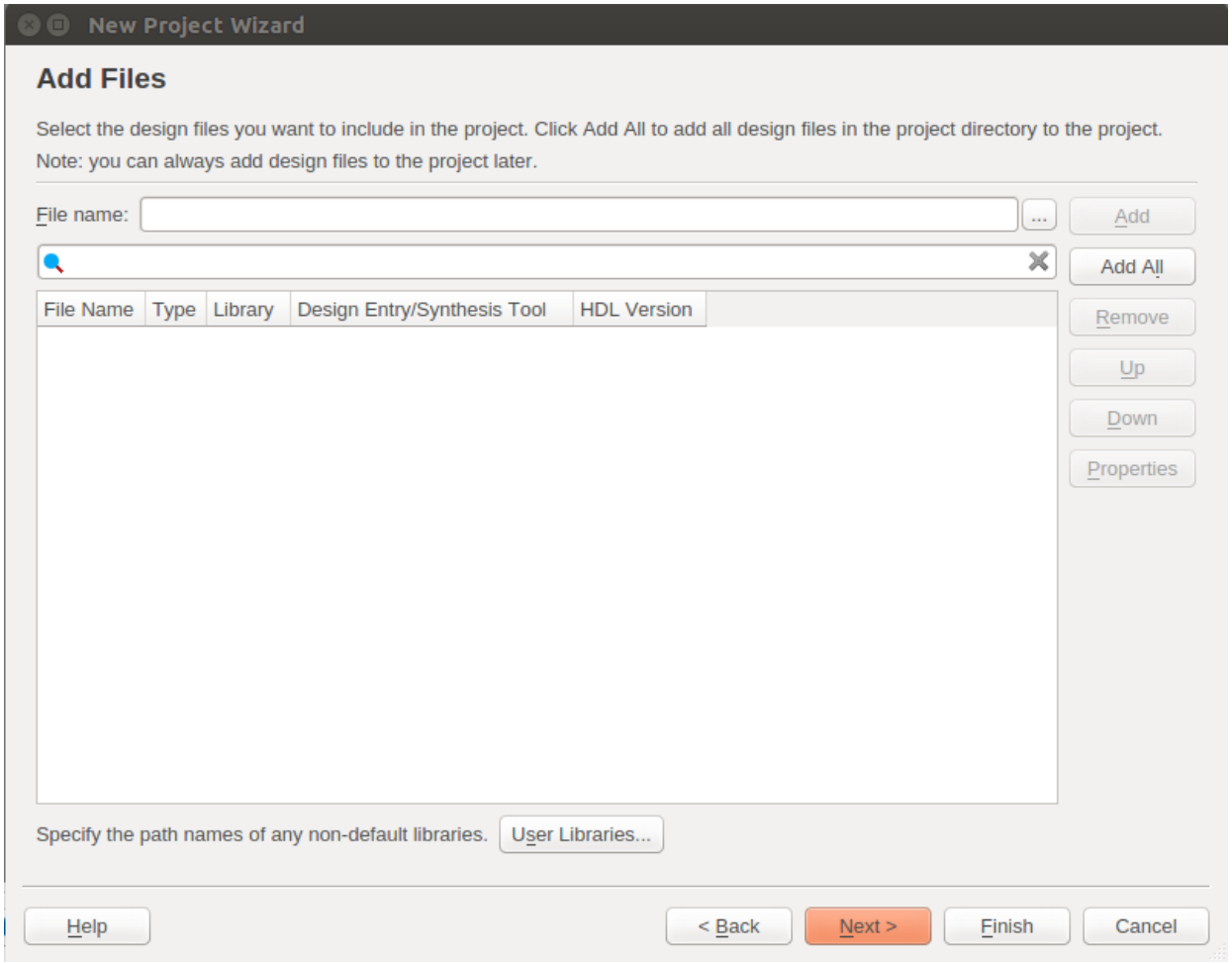
In the second window you should give the project the following name: *gNN_lab*1 where **NN** is replaced with your 2-digit group number. The working directory for your project will be different than that shown here. Use your network drive for your project files.



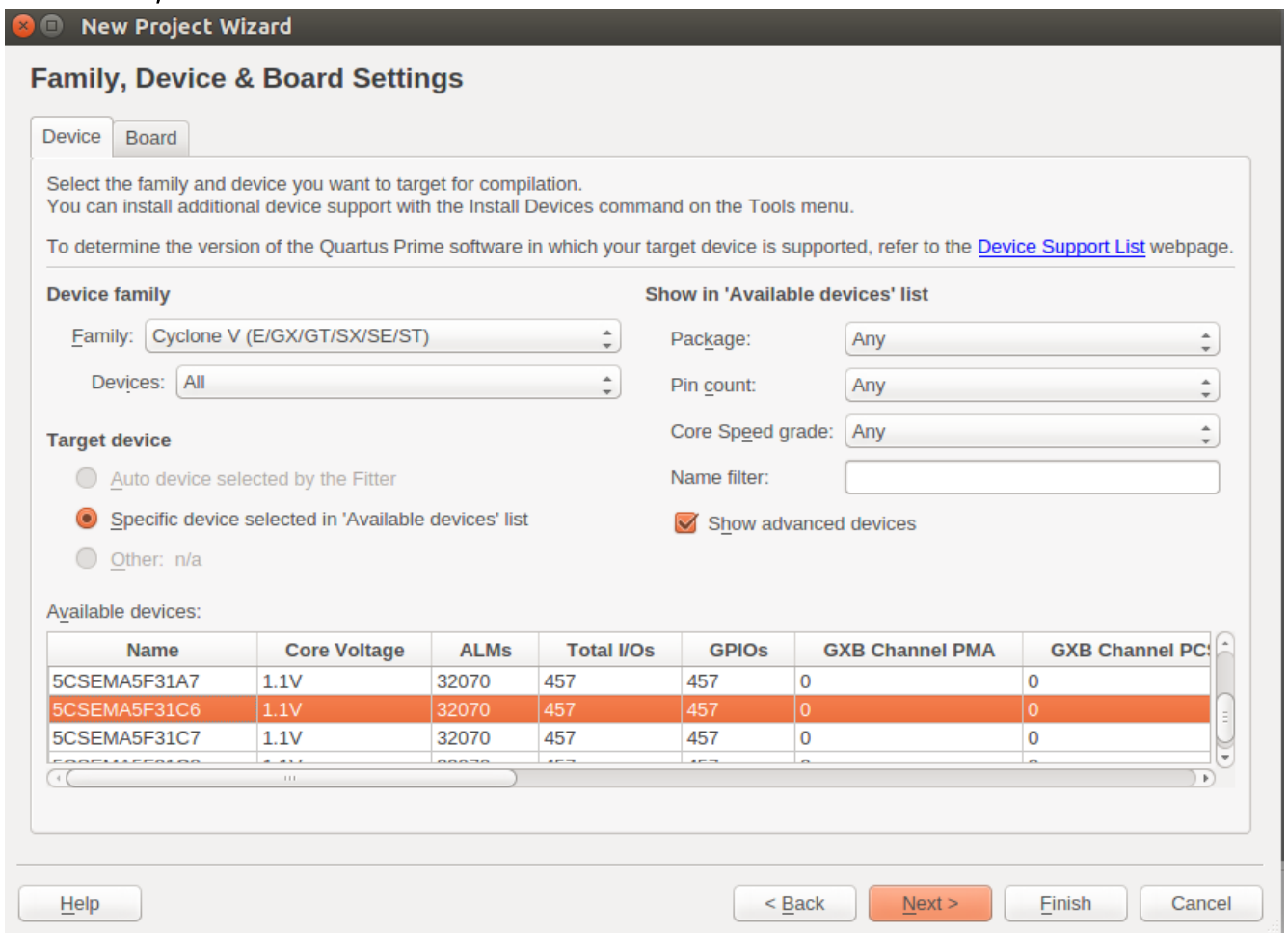
We don't have a project template at this point, so select **Empty project** and proceed.



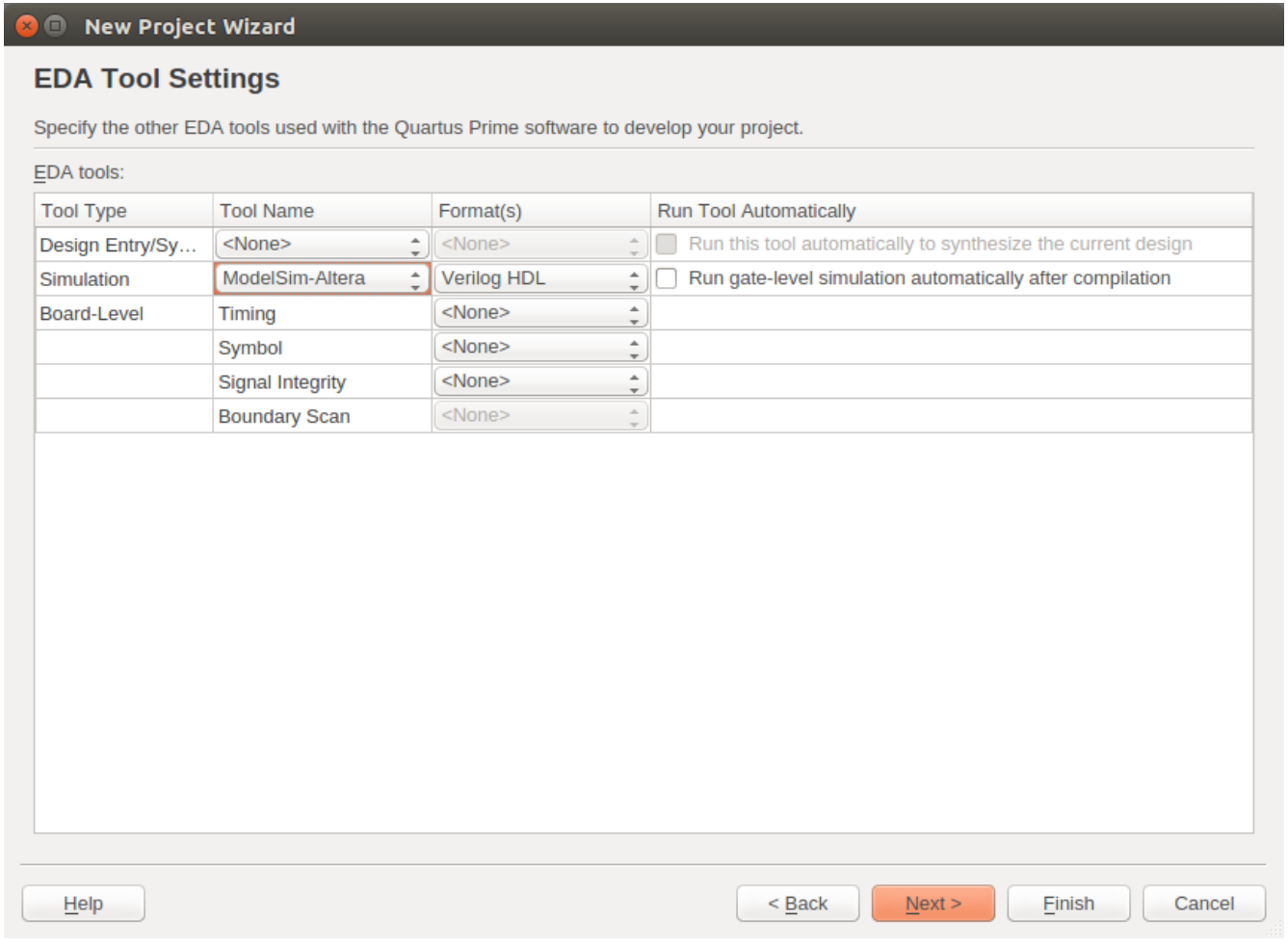
You will add files later, so for now, just click on "Next".



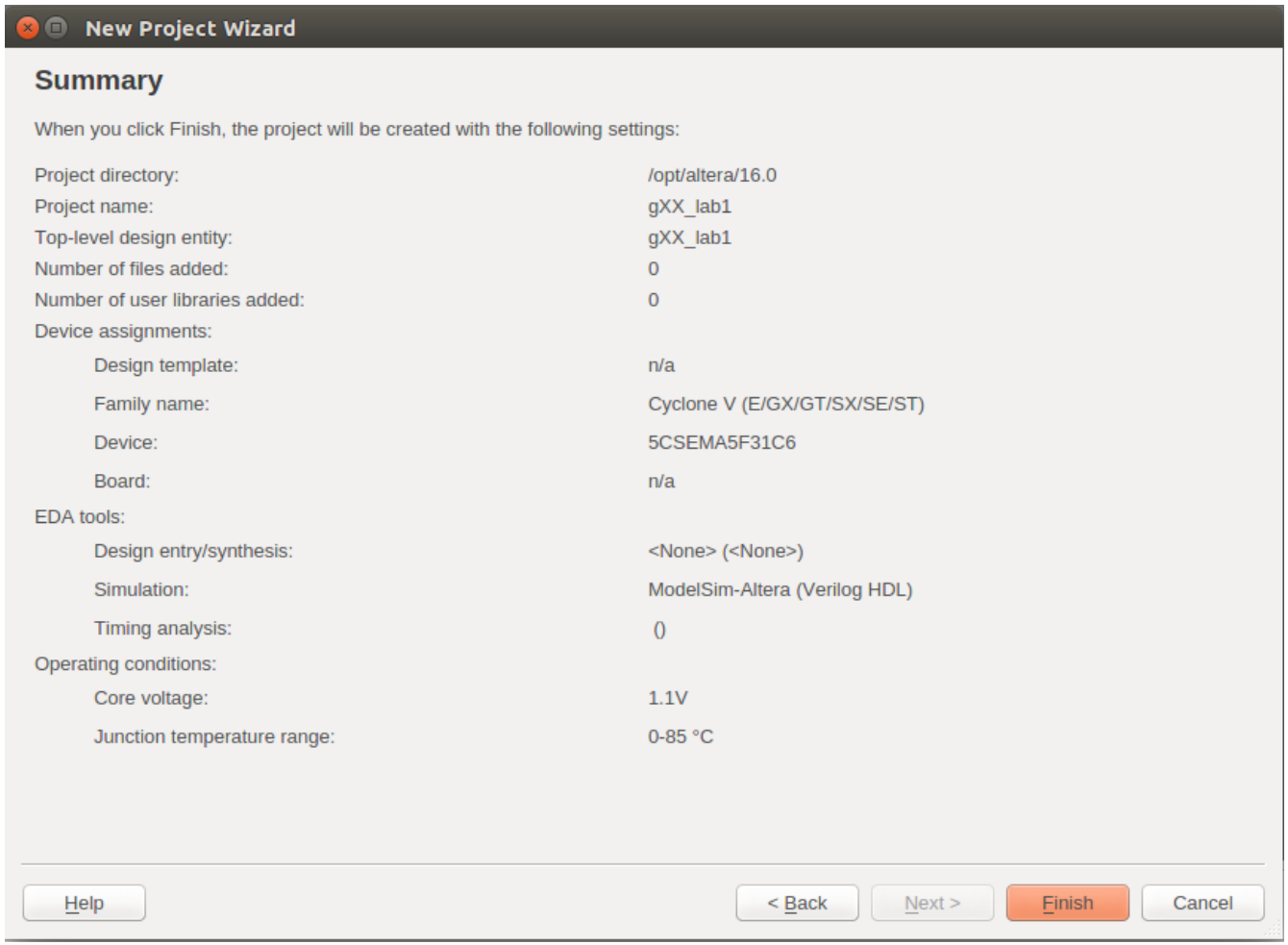
In later labs you will be downloading the designs to an FPGA device on the Altera development board. These devices belong to the **Cyclone V** family of FPGAs, with the following part number: **5CSEMA5F31C6** So, to ensure proper configuration of the FPGAs in future labs, select this device.



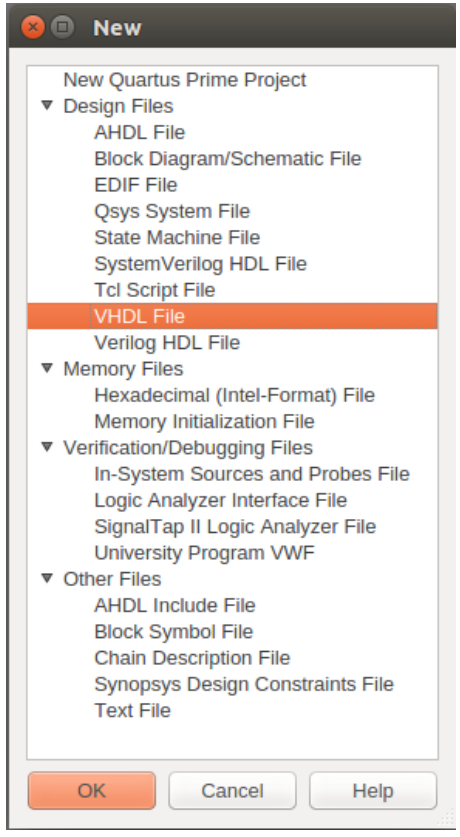
This dialog box permits the designer to specify 3rd-party tools to use for various parts of the design process. We will be using a 3rd-party Simulation tool called **Modelsim-Altera**, so select this item from the Simulation drop-down menu.



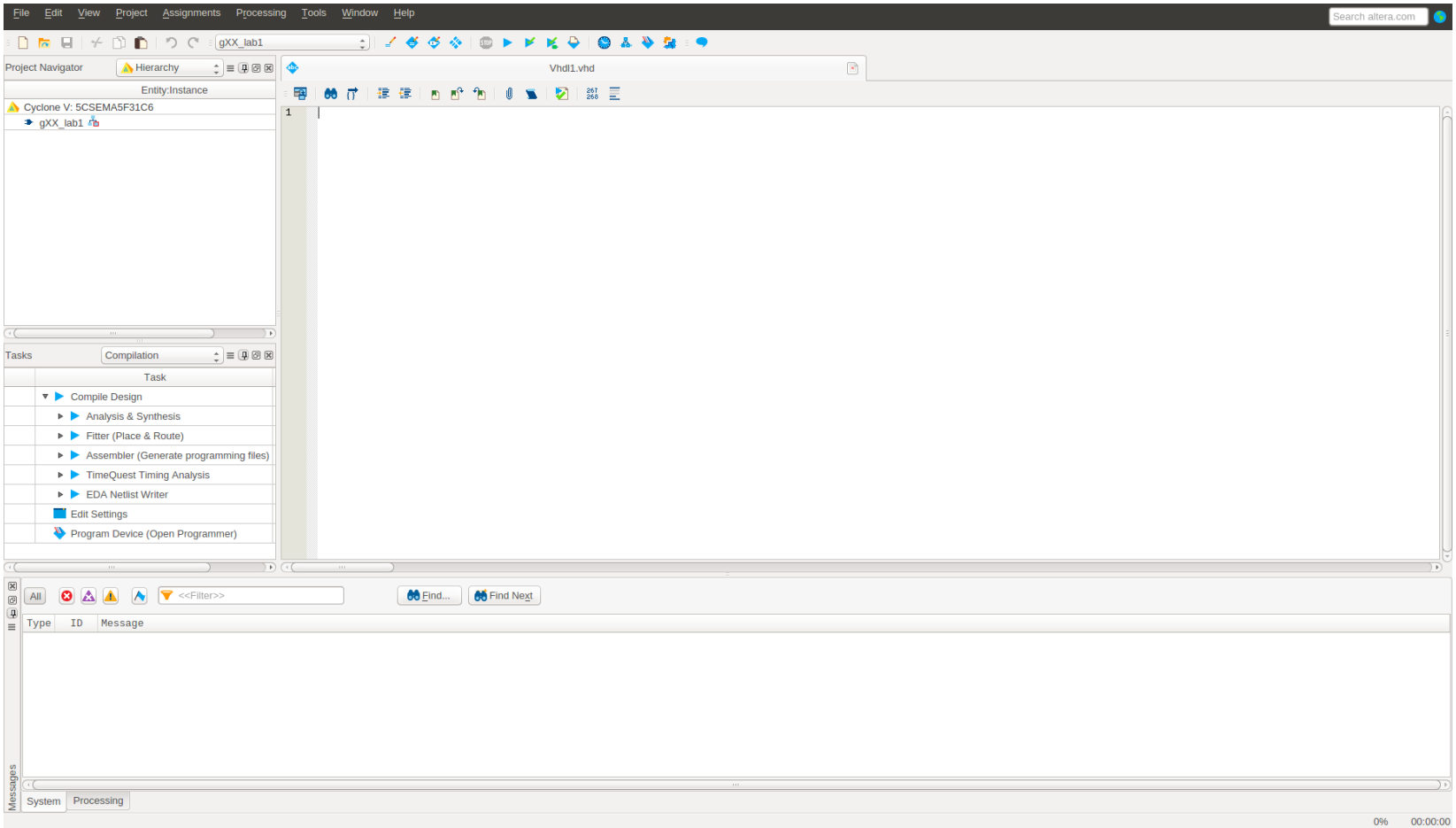
The final page in the New Project Wizard is a summary. Check it over to make sure everything is OK (e.g. the project name, directory, and device assignment), then click on the **Finish** button.



Your project framework is now ready. Now, from **File**, select **New**, and select **VHDL file** from the list as shown.



You should have a VHDL editor opened in your framework. We will write and edit our code from this editor.



Synchronous Counter in VHDL

In this Lab, you will implement a synchronous 8-bit counter with the following specifications:

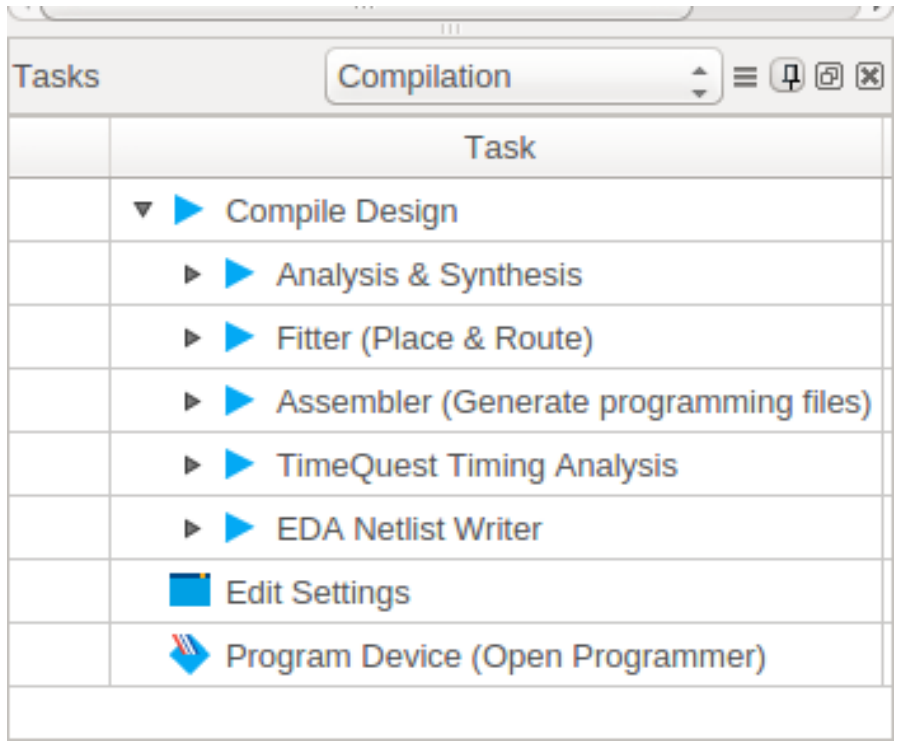
- Synchronous reset
- Up/down counting features
- Enable counting option

The library and entity declarations are provided below.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
entity gXX_lab1 is
    Port ( clk          : in std_logic;
           direction    : in std_logic;
           rst          : in std_logic;
           enable       : in std_logic;
           output       : out std_logic_vector(7 downto 0));
end gXX_lab1;
```

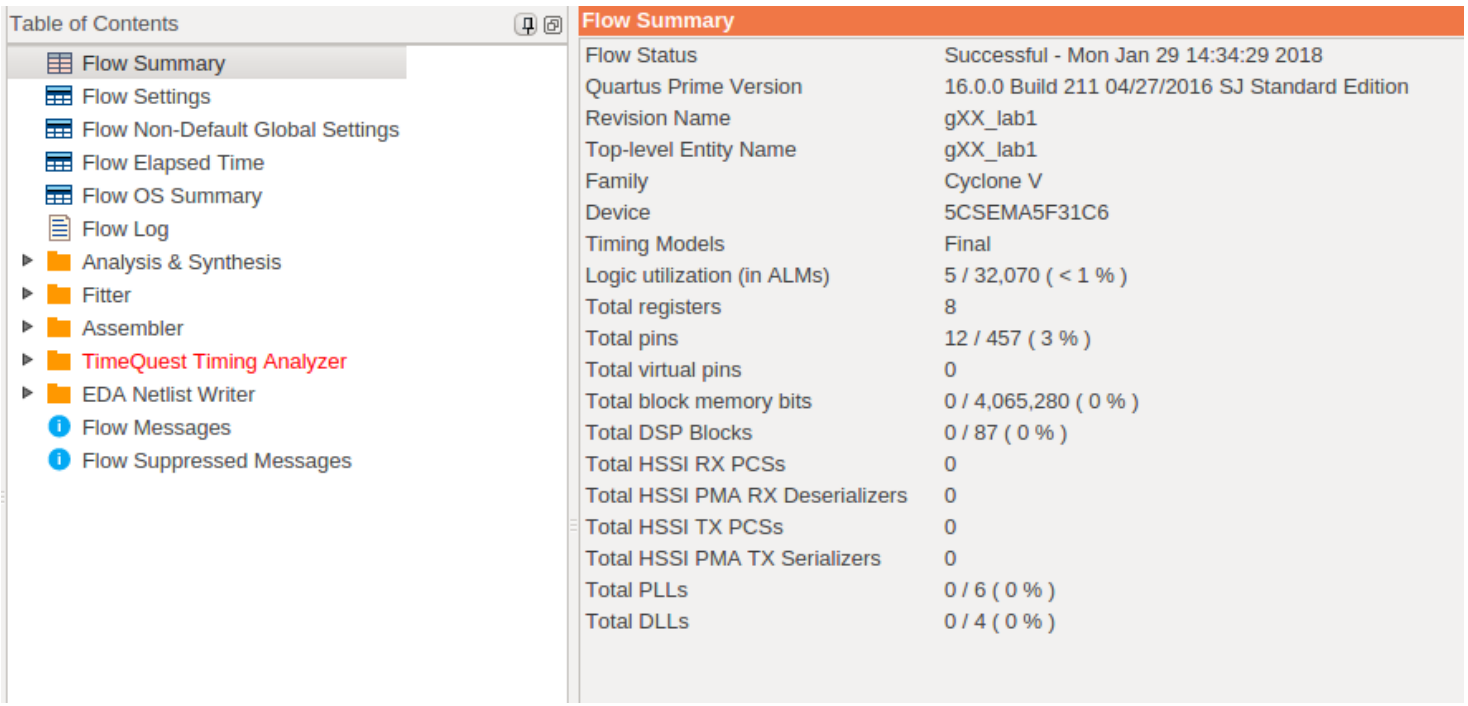
Compilation

Once you have finished writing your VHDL code, compile your design from **Processing** → **Start Compilation**, or from the Tasks navigator. If your code is free of syntax errors, the entire compilation process should turn green.



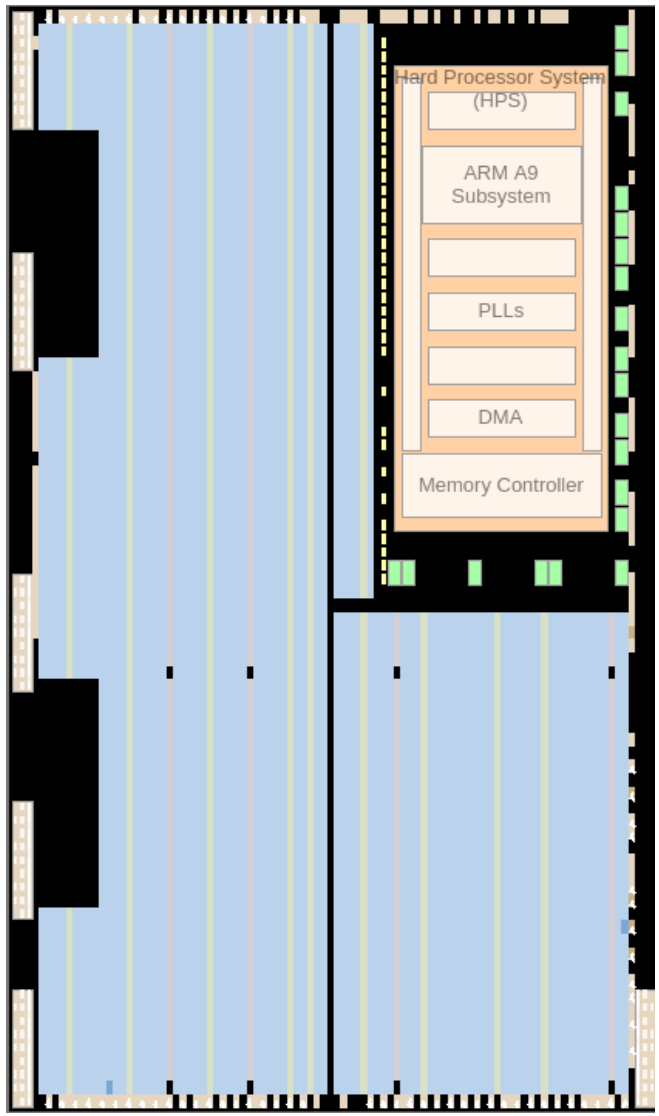
Compilation Summary

At the end of compilation process, a new tab in the main window will appear with a list of flow summary; describing the resource utilization you use to implement your function in the DE1-SoC board. Take a minute to observe this report to have an understanding of hardware utilization.



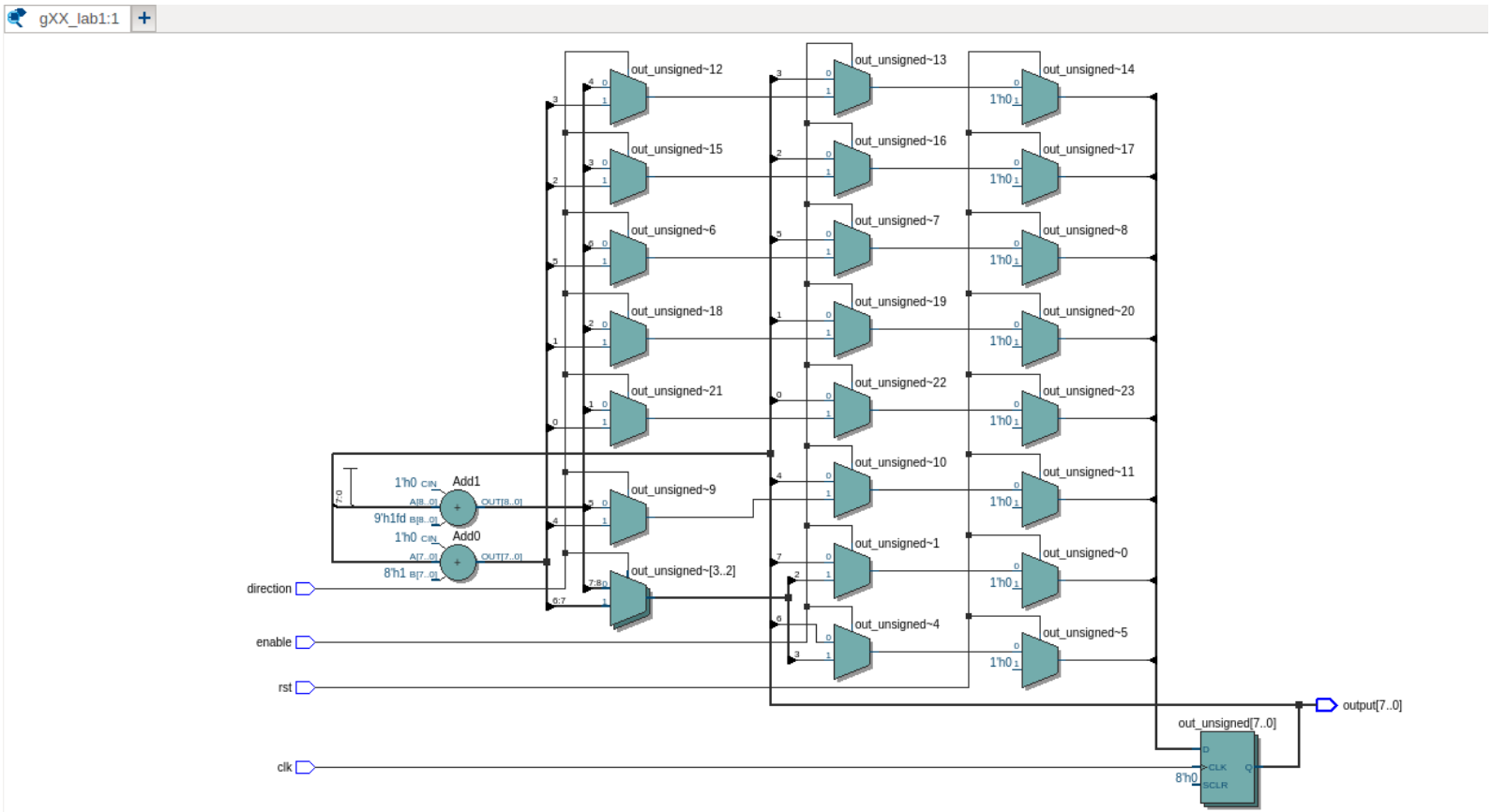
Chip Planner

The hardware implementation you made utilizes hardware units such as logic, memory, DSP, registers, etc. that correspond to a physical unit in the FPGA board. Click **Tools** $\rightarrow$ **Chip Planner** to open the FPGA layout and observe the physical utilization of your design. Areas highlighted in darker blue are the units that your design utilize.



RTL Viewer

RTL viewer is a powerful tool to visualize the hardware resources your code utilize in detail. Click **Tools** → **Netlist Viewers** → **RTL Viewer** to see the logic schematic that corresponds to your counter implementation.



Writeup of the Lab Report

At this point, you should know how to compile a VHDL code, fix syntax errors, and have an understanding of resource utilization, chip planning and RTL viewer. You are required to submit a written report and your code to myCourses on the first Friday after the Lab1 period at midnight.

- The report in the electronic file should be in PDF format.
- The report should be written in the standard technical report format.
- Document every design choice clearly.
- Everything should be organized for the grader to easily reproduce your results by running your code through the tools.
- The code should be well-documented and easy to read.
- The grader should not have to struggle to understand your design.

Writeup of the Lab Report - cont'd

Please refer to the example lab report from myCourses content for an example lab report template.

Your report must include:

- The VHDL code you wrote for the complex counter
- Compilation report (Flow Summary) of your design
- Discussion on the resource utilization of your design (i.e. how many registers are used and why? What changes would you respect in the resource utilization if you increased the bit size of the counter?)
- Chip planner screenshot, with used resources highlighted
- RTL view of the circuit, with description of resources

Grading Sheet

Group Number:
Name 1:
Name 2:

Task	Grade	/Total	Comments
Creating Project		/10	
VHDL for counter		/50	
Resource Utilization		/20	
Floorplan & RTL		/20	