# VHDL Assignment #1: Getting Started with Quartus CAD Software

## 1 Instructions

- TA in charge: Arish Yaseen (arish.yaseen@mail.mcgill.ca) - please utilize discussion boards on myCourses for questions as much as possible.
- Due date: Friday, September 9, 2022 by 11:59 pm EDT.
- Submission is in teams using myCourses. **Only one team member submits**. In the report, provide the names and McGill IDs of the team members.
- Late submissions will incur penalties as described in the course syllabus.

## 2 Introduction

In this assignment, you will learn how to use Intel Quartus II FPGA design software, how to set up a project, and the basics of writing VHDL code by following a step-by-step tutorial.

## 3 Learning Outcomes

After completing this lab you should know how to:

- Run the Intel Quartus software
- Create the framework for a new project
- Create a new VHDL file

## 4 Run Intel Quartus

In this course you will be using commercial FPGA design software: the Intel Quartus Prime program and the Mentor Graphics ModelSim simulation program. Quartus Prime and ModelSim are installed on the computers in the lab. You can also obtain a slightly restricted version, the Quartus Lite edition, from the Intel web site[1]. The program restrictions will not affect any designs you will be doing in this course. You can (and you should) install the applications on your personal computer to work on your project outside of the lab. *You should use version 18.0 of the program*, as this is the latest version that supports the prototyping board (the Altera DE1-SoC board) that you will be using.

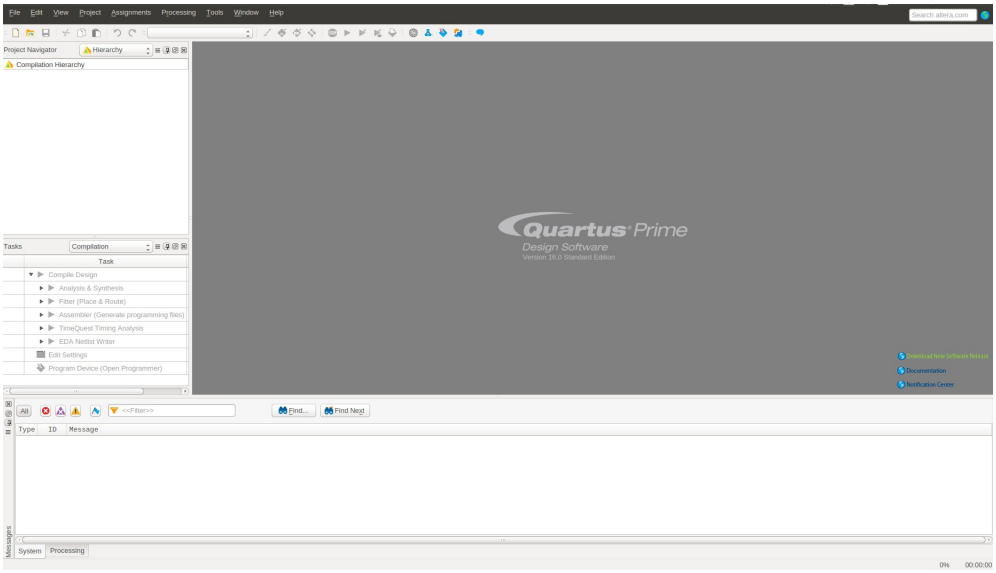
For Mac users, Intel Quartus is not available for MacOS. As such, you can connect to the lab computers via remote access. Follow the these instructions:

1. Connect to the McGill VPN as shown here: https://mcgill.service-now.com/itportal?id=kb_article&sysparm_article=KB0010687
2. Use Microsoft Remote Desktop (https://mcgill.service-now.com/itportal?id=kb_article&sysparm_article=KB0010725) to connect to machines 156TR4060-01.campus.mcgill.ca - 156TR4060-17.campus.mcgill.ca. Note that the number at the end of the name of the machine (before ".campus.mcgill.ca") is the machine number in the lab (a total of 17 machines).

To begin, start Quartus Prime by selecting it in the Windows Start menu:

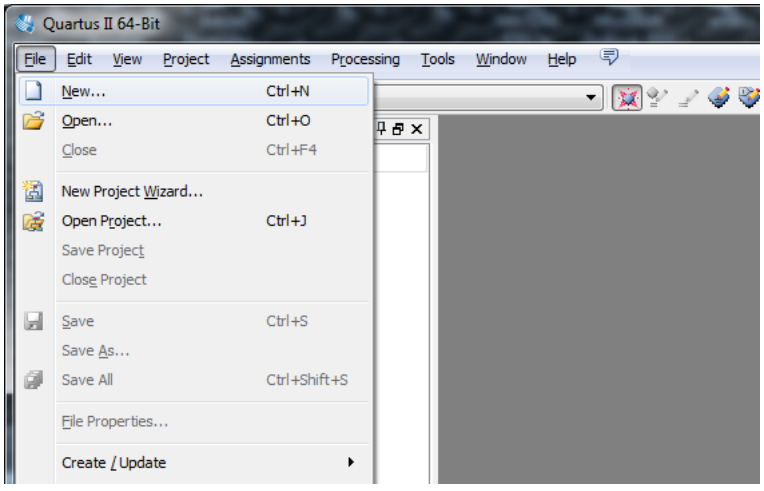

[1] Go to https://www.intel.com/content/www/us/en/programmable/downloads/download-center.html Then select Version 18.0 and finally choose the Lite edition. You may have to register for an Intel account. Also, before you press the Download button make sure that the correct version and edition are selected through the dropdown menus.

The following window will appear on startup (this shows version 18.0 downloaded from Intel's web site; the versions on the lab computers may look slightly different).



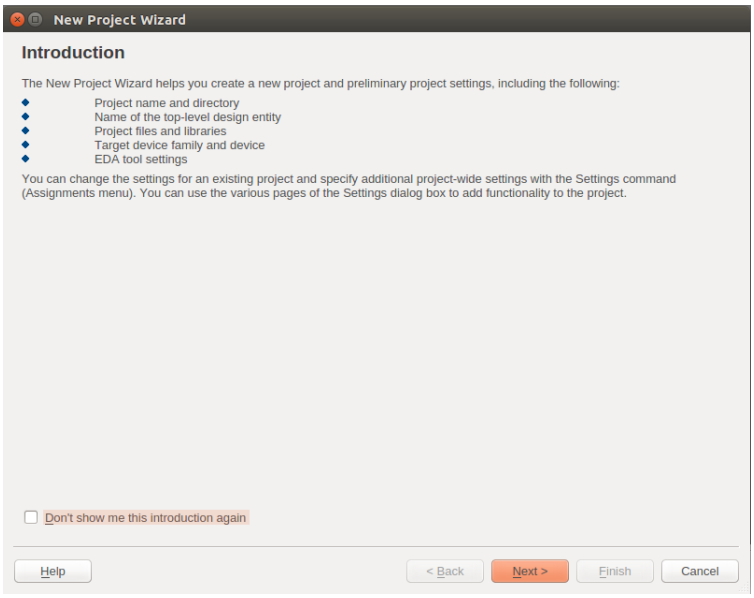

Intel Quartus Prime employs a project-based approach. The goal of a Quartus project is to develop a hardware implementation of a specific function, targeted to an FPGA (Field Programmable Gate Array) device. Typically, the project will involve a (large) number of different circuits, each designed individually, or taken from circuit libraries. Project management is therefore important. The Quartus Prime 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 Prime 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.

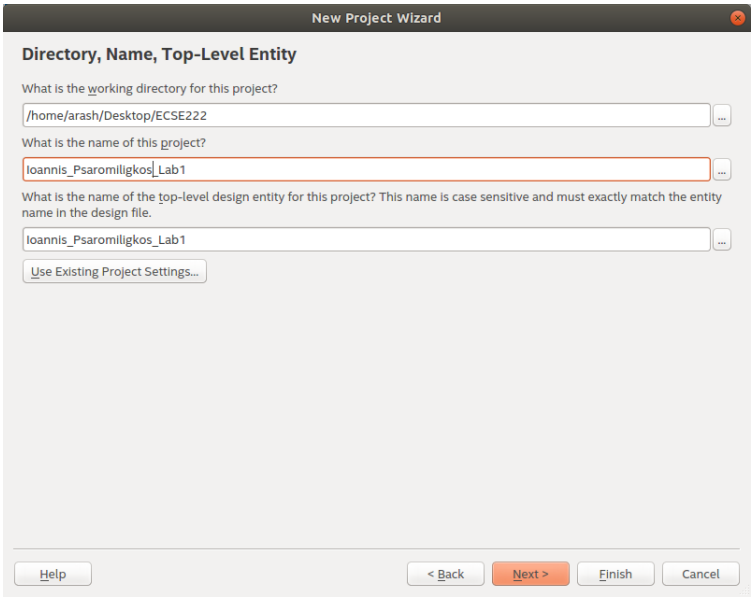



## 5 Creating a New Project

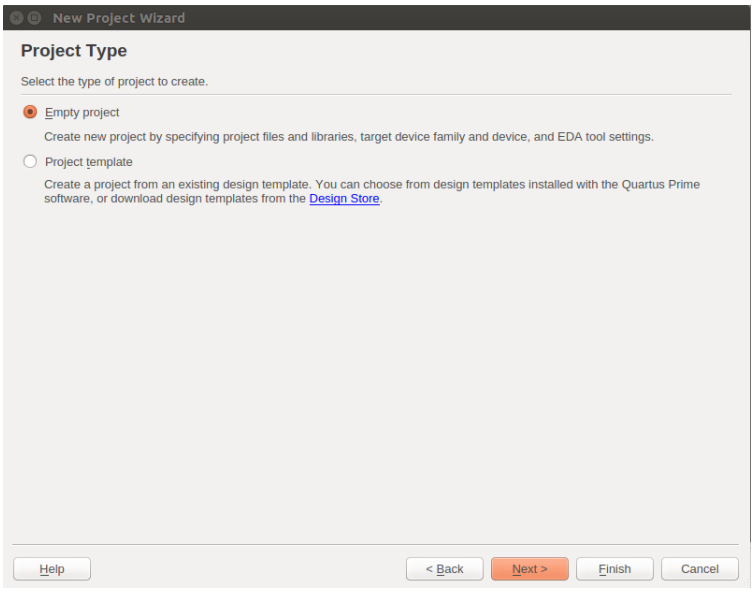
The New Project Wizard involves going through a series of windows. The first window is an introduction, listing the settings that can be applied. After reading the text on this window, click on "Next" to proceed.



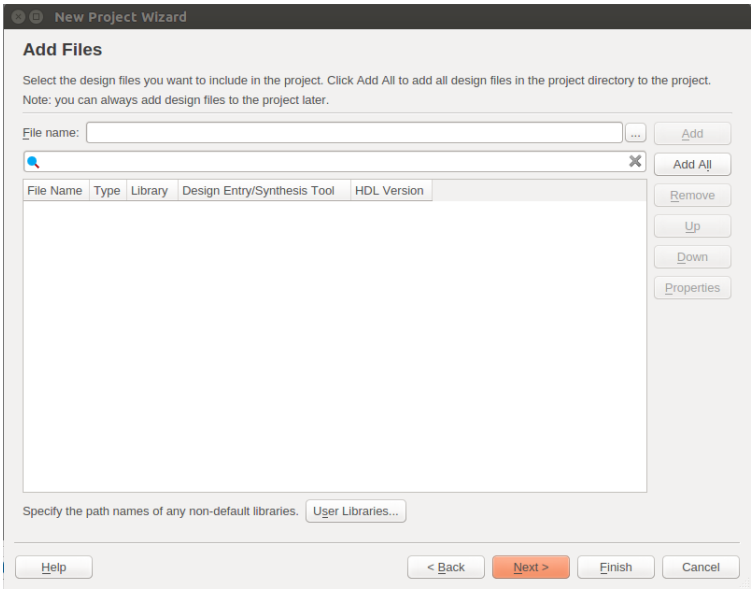

In the second window, you should give the project the following name: `firstname_lastname_vhdl1`. Make sure to replace `firstname_lastname` with your full name. The working directory for your project will be different than that shown in the screenshot below. Make sure that you change the working directory to the directory of your choice. If you use a lab computer, use your network drive for your project files.



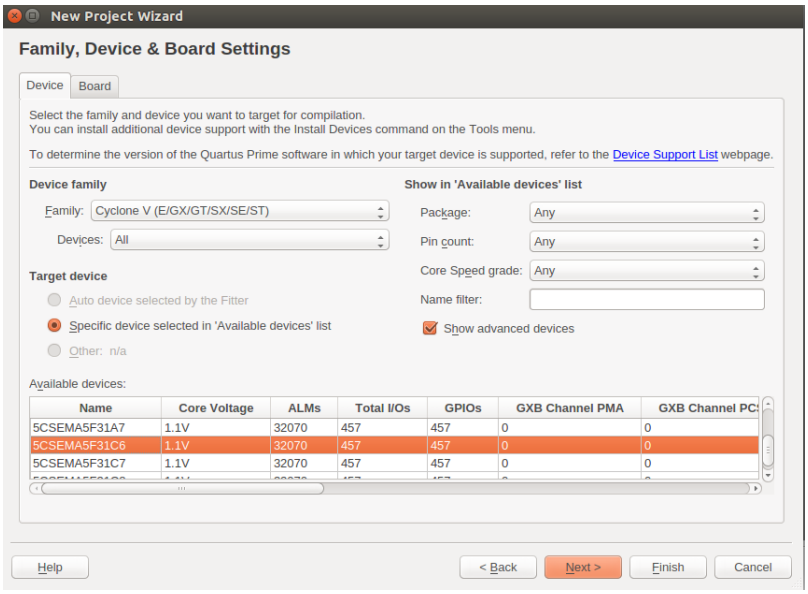

We don't have a project template at this point, so select **Empty project** and proceed.



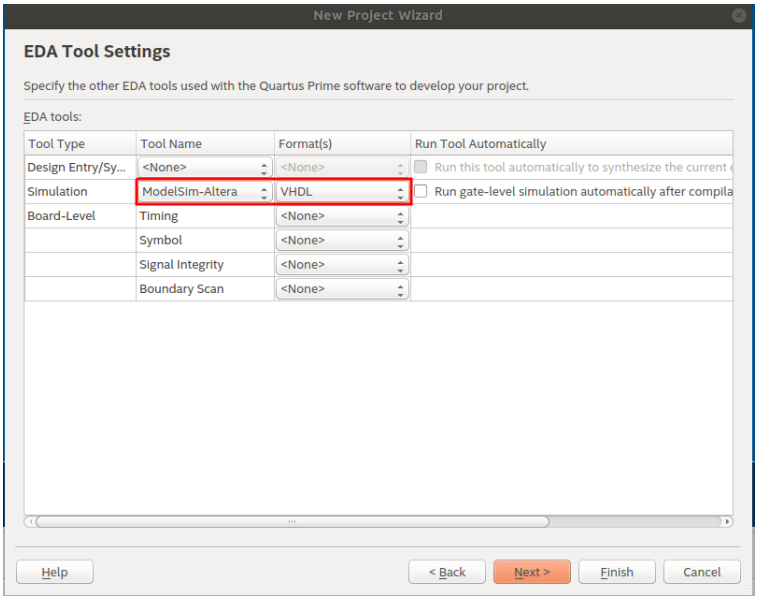

You will add files later, so for now, just click on "Next".



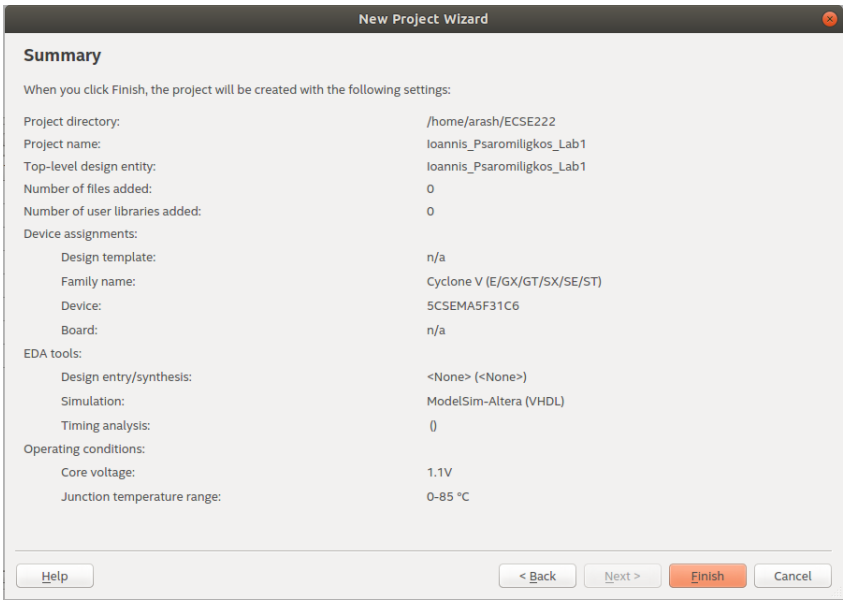

Later in the course, you will be downloading a design to an FPGA device on the DE1-SoC board. These devices belong to the **Cyclone V** family of FPGAs, with the following part number: **5CSEMA5F31C6**. To ensure proper configuration of the FPGAs, select this device as shown below.




The dialog box in the next window 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 okay (*e.g.*, the project name, directory, and device assignment) then click **Finish**.

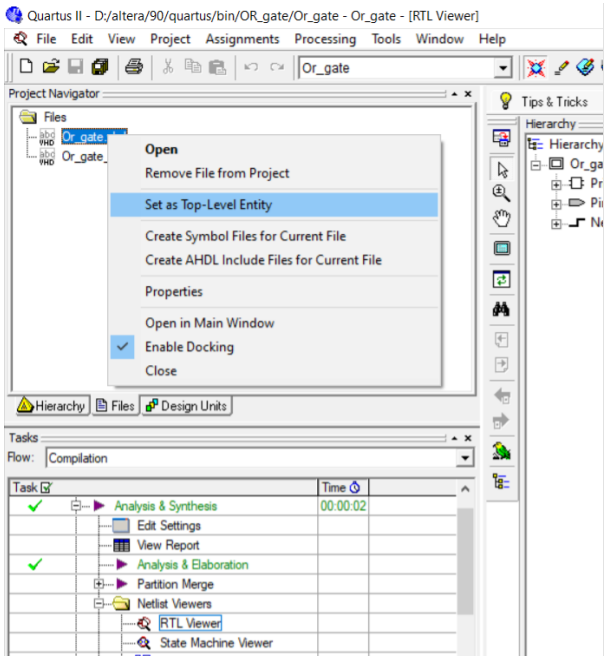



Your project framework is now ready.

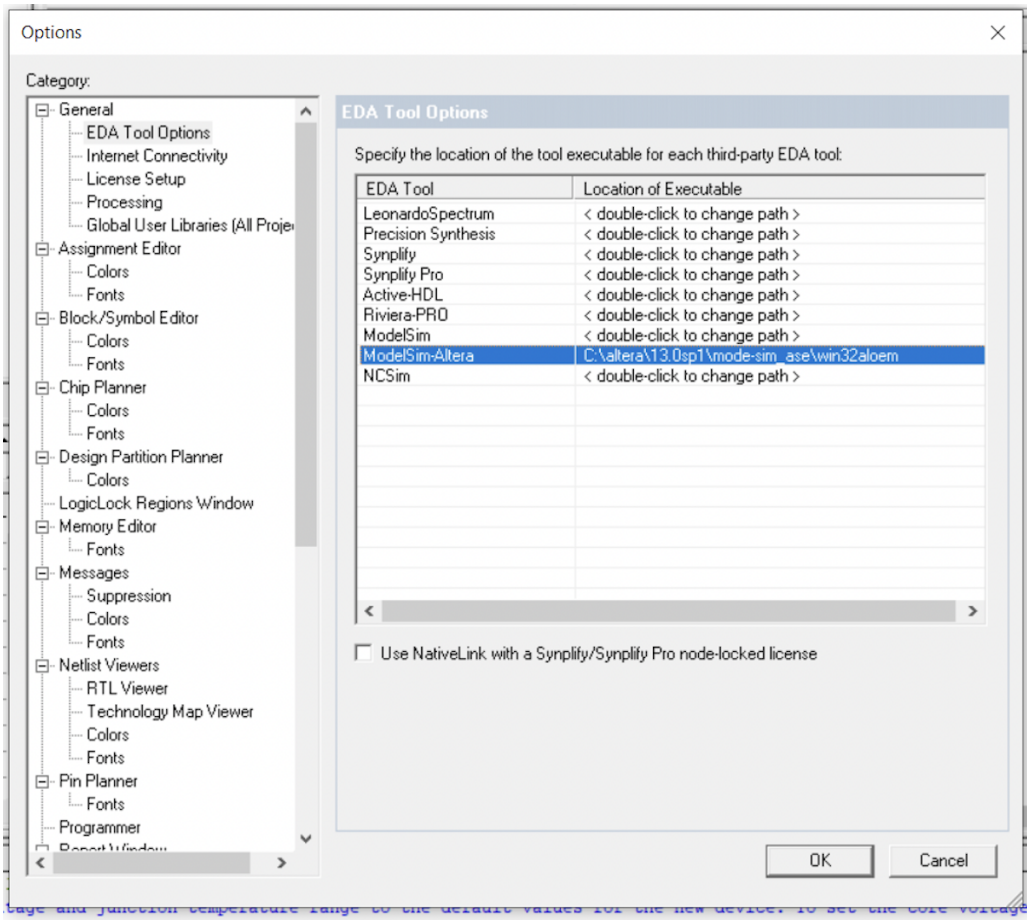
# 6 Writing Hello Gate code in VHDL

The following VHDL code describes an OR gate (available online at https://www.edaplayground.com/x/A4). Create a new VHDL file within your project and write/paste the code in the new file. Make sure to select the OR gate as the top-level design entity (see figure below).

```
-- Simple OR gate design
library IEEE;
use IEEE.std_logic_1164.all;

entity or_gate is
port(
  a: in std_logic;
  b: in std_logic;
  q: out std_logic);
end or_gate;

architecture rtl of or_gate is
begin
  process(a, b) is
  begin
    q <= a or b;
  end process;
end rtl;
```

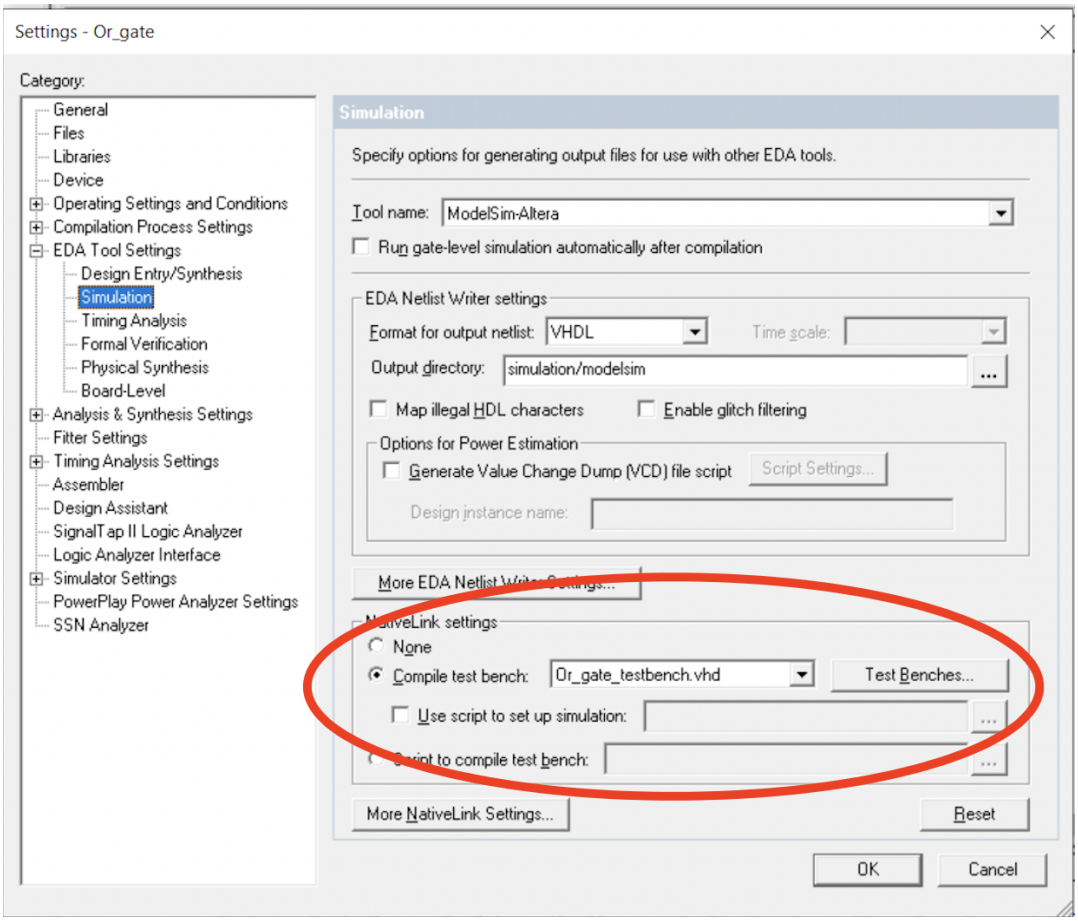



Next go to `Tools > Options > General > EDA Tools` to make sure that the path to Altera Modelsim is configured correctly. If you installed "Quartus with Altera ModelSim" the path should be similar to the one shown in the figure below (*i.e.*, on Windows: `C:\altera\13.0sp1\mode-sim_ase\win32aloem`, otherwise you will need to browse to where you installed Altera Modelsim and point it to the win32aloem directory (see figure below).

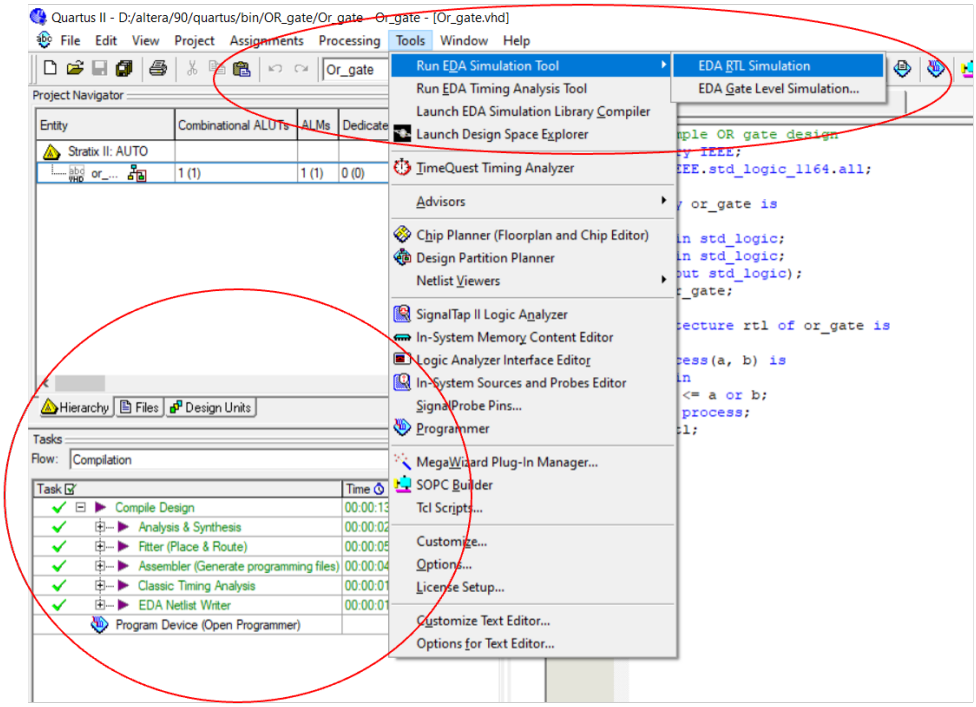


Now we need to create a testbench file. Test benches are used to test our VHDL code. In the testbench file, we will typically define the VDHL code of our hardware as the device under test (DUT) and then define the input vectors to test the DUT. The VHDL code that describes the testbench for the OR gate is provided below. Write/paste the code in a new VHDL file within your project. This file should be defined as the testbench in the settings. To specify the testbench go to this path `Assignments > Settings > EDA Tool Settings > Simulation` and specify your testbench file as shown in the figure below.

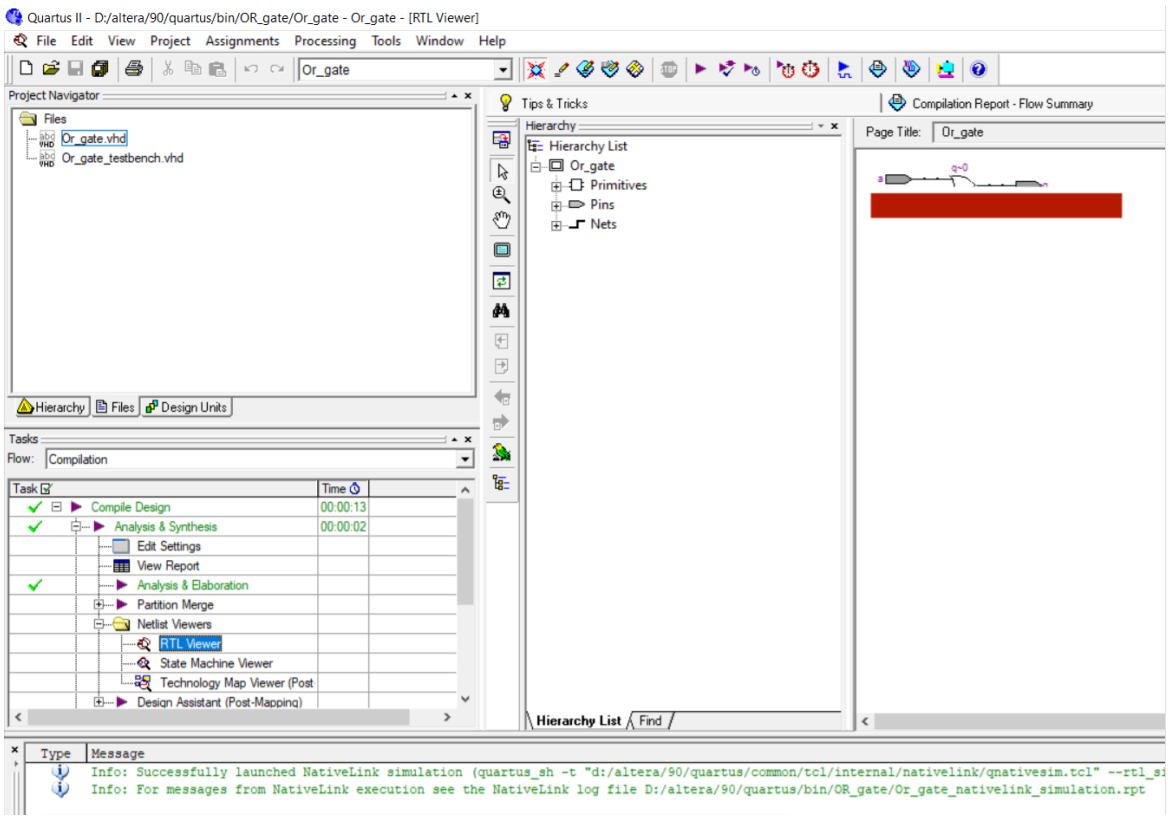
```
-- Testbench for OR gate
library IEEE;
use IEEE.std_logic_1164.all;

entity testbench is
-- empty
end testbench;

architecture tb of testbench is

-- DUT component
component or_gate is
port(
  a: in std_logic;
  b: in std_logic;
  q: out std_logic);
end component;

signal a_in, b_in, q_out: std_logic;

begin

  -- Connect DUT
  DUT: or_gate port map(a_in, b_in, q_out);

  process
  begin
    a_in <= '0';
    b_in <= '0';
    wait for 1 ns;
    assert(q_out='0') report "Fail 0/0" severity error;

    a_in <= '0';
    b_in <= '1';
    wait for 1 ns;
    assert(q_out='1') report "Fail 0/1" severity error;

    a_in <= '1';
    b_in <= 'X';
    wait for 1 ns;
    assert(q_out='1') report "Fail 1/X" severity error;

    a_in <= '1';
    b_in <= '1';
    wait for 1 ns;
    assert(q_out='1') report "Fail 1/1" severity error;

    -- Clear inputs
    a_in <= '0';
    b_in <= '0';

    assert false report "Test done." severity note;
    wait;
  end process;
end tb;
```




Now you can compile your code by clicking on "Compile Design" in "Tasks" window. After receiving all green check-marks in the Tasks window, follow the path: `Tools > Run EDA Simulation Tool > EDA RTL Simulation` to see the output waveform (see figure below). The ModelSim tool will open with a plot of the simulation result.



You can also view a schematic of the designed circuit under RTL representation, which is the gate-level representation of your design. Go to the "Netlist Viewers" section from the "Analysis & Synthesis" tab in Quartus (see figure below).




# 7 Deliverables

You are required to submit the following deliverables on MyCourses. Please note that a single submission is required per group (by one of the group members). Reports for VHDL assignments should be generated on a computer and converted to PDF format - only this will be accepted. All VHDL assignment files should include the names and McGill IDs of all team members. Once ready, upload your files (PDF and zip, as explained below) into the relevant assignment under the "Assignments" tab on myCourses. Note that you can upload as many times as you want before the deadline. Only the last upload will be kept by the system.

- Lab report. The report should include the following parts: (1) Names and McGill IDs of group members, (2) an executive summary (short description of what you have done in this VHDL assignment), (3) answers to all questions (if applicable), (4) legible figures (screenshots) of schematics and simulation results, where all inputs, outputs, signals, and axes are marked and visible (if applicable), (5) an explanation of the results obtained in the assignments (mark important points on the simulation plots), and (6) conclusions. Note - students are encouraged to take the reports seriously, points will be deducted for sloppy submissions.
- Project files. Create a single .zip file named `vhdl#_firstname_lastname` (replace `#` with the number of the current VHDL assignment and `firstname_lastname` with the name of the submitting group member). The .zip file should include the working directory of the project.