Quartus II:
An Introduction for COE 4DM4 LABS

By: Mohammadreza Binesh Marvasti

This document provides a basic introduction to using Quartus II software. The example given has been used on the latest Quartus II (V.10, V.11 and V.12). This document is broken down into the following sections.

1. Getting Web version of Quartus
2. Creating a Quartus Project
3. Creating a new VHDL file or Opening an existing VHDL file in Quartus
4. Compiling in Quartus
5. Analysis and Synthesis Settings
6. Compilation Report
7. Timing Analysis
8. RTL Viewer
9. Floorplan Editoor

1.) Getting Web Version of Quartus

It is assumed that Quartus is already installed onto your computer. If not, Altera has a free version of Quartus called Quartus II Web Edition that you can download for free here:
https://www.altera.com/download/software/quartus-ii-we
Just download the Quartus II Web Edition as shown in the screen capture below.

Download Option 1: Individual Files

Quartus II Web Edition	Platform	File Name	Size
Quartus II Web Edition Service Pack 2	Windows	12.0sp2_quartus_free_windows.exe MD5: 3aa4e0871aaa29ffae5ecd3b42dd6a7b	2.8 GB
Quartus II Web Edition Service Pack 2	Linux	12.0sp2_quartus_free_linux.tar.gz MD5: eacce40008d33b5eca7277b1f822e621	3.7 GB

Note in order to fully download you may need to register at Altera. By clicking "Quartus II Web edition" it will take to a screen which you can login in to obtain a license file or create a user account. These steps are intuitive. The goal here is to create an account, download install Quartus II .
Note that in Quartus version 8.1 or newer, a license file is no longer required for Quartus II Web Edition software.

Now, in order to have a simulator installed in your machine, you need also to install the Modelsim-Altera starter as follows:

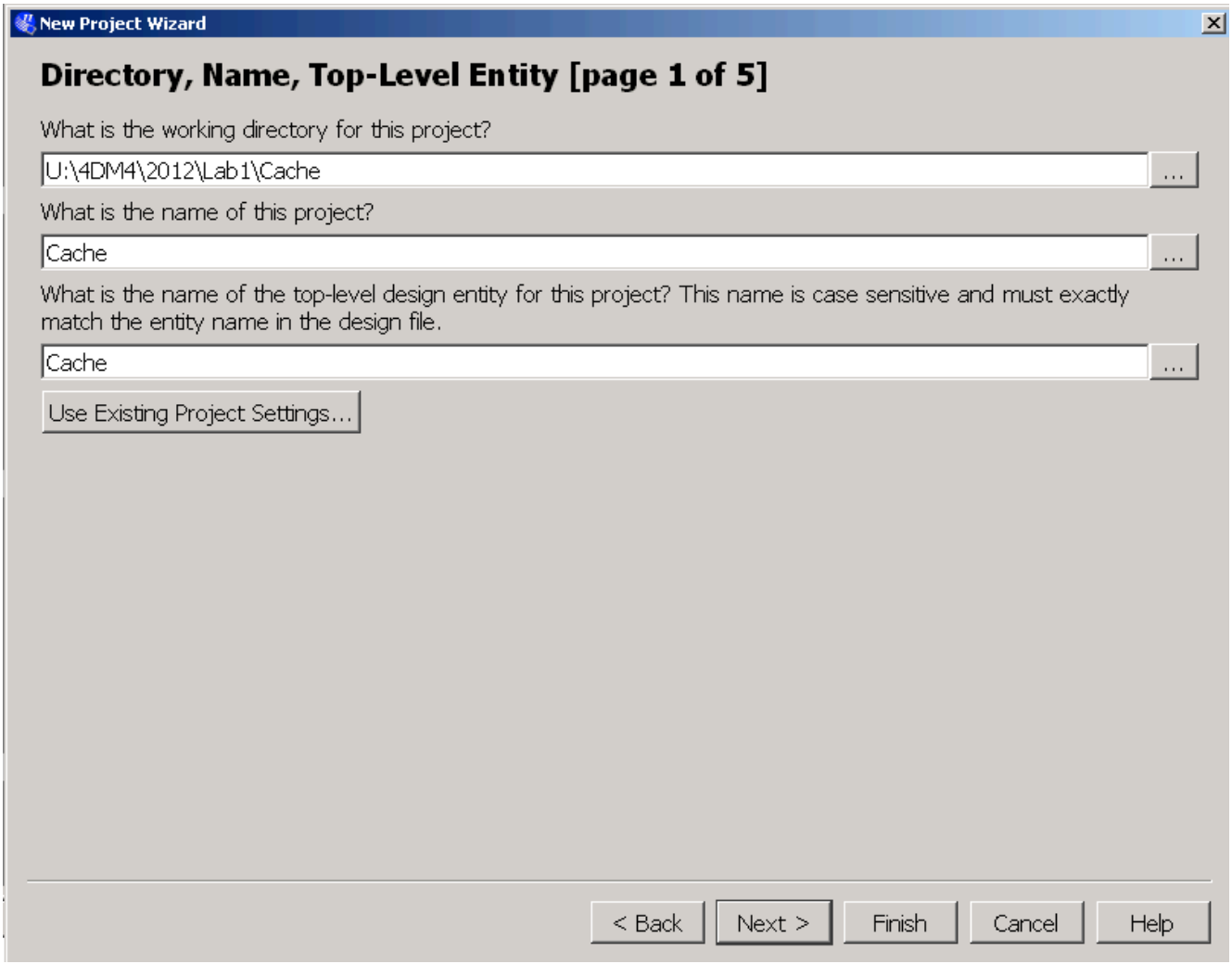
Other Individual Download Files:

- Quartus II Subscription Edition
- ModelSim-Altera
- ModelSim-Altera Starter
- Nios II EDS
- DSP Builder
- Programming Software

Writing a testbench and Setting-up the Modelsim-Altera were provided in a separate video documentation.

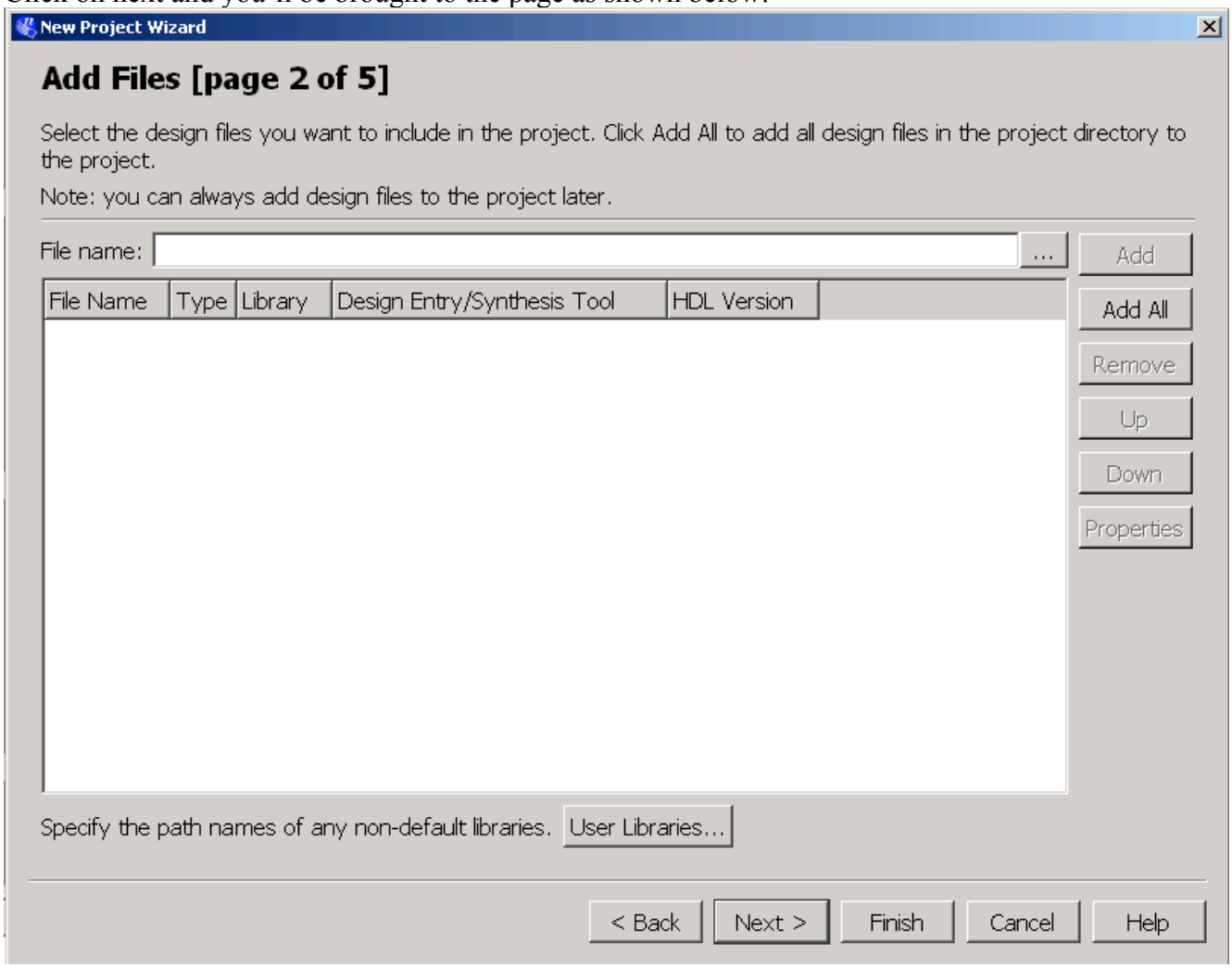
2.) Creating a Quartus Project

Now that Quartus is on your computer and has a valid license, the next step is to create a Quartus project. Creating a Quartus project is ABSOLUTELY mandatory; otherwise none of the VHDL files you create or open in Quartus will compile. Think of a Quartus project as a "suitcase" that stores all of your .vhd (by convention VHDL files have a .vhd extension, not .vhdl), simulation files. In Quartus, go to **File→New Project Wizard**, and you'll be brought to the following screen (if you do no see the screen below, click NEXT and it will show up):



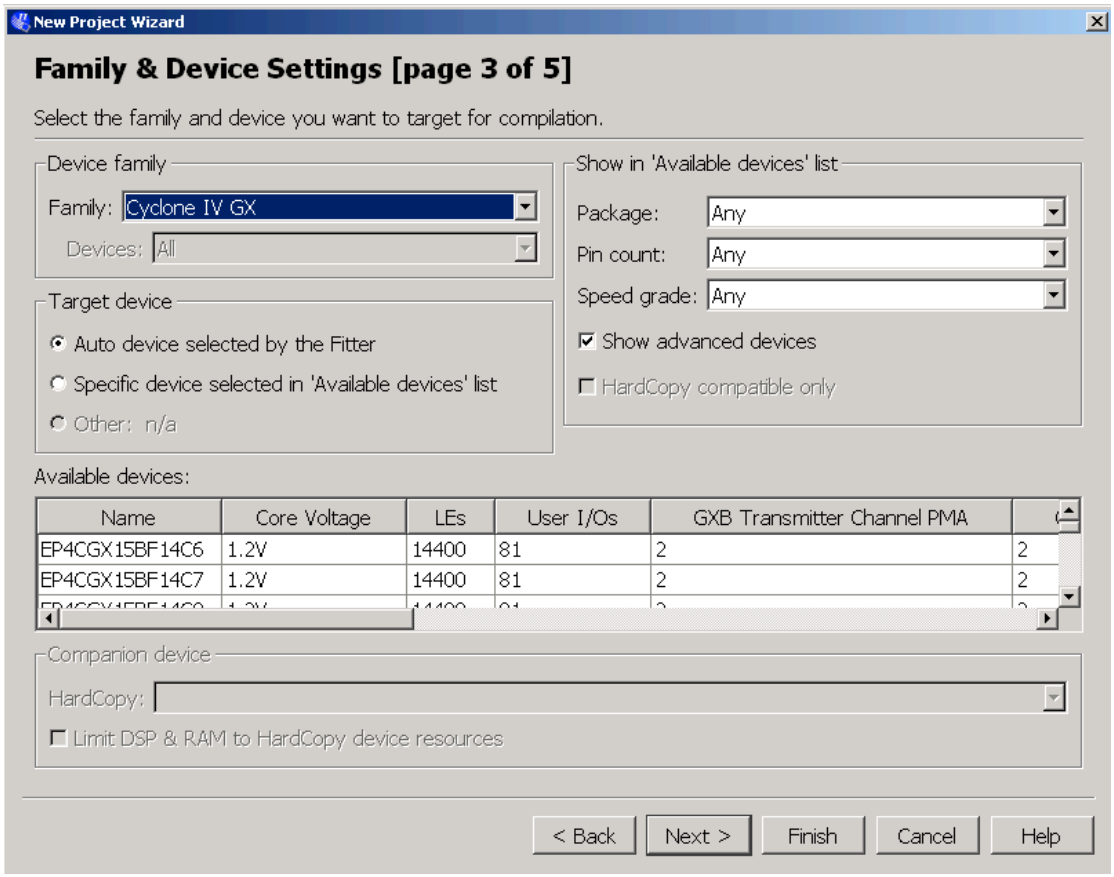
In the first line, identify where you want to store your project on the hard drive. By clicking the icon with the 3 dots, you can create a new folder to store your project files. In the second line, identify a name for your project. In this example it is called 'Cache'. In the third line, identify the top level entity for the project. Here it is called 'Cache'. NOTE: It's not by coincidence that the name of the project and the top level entity are the same name (that being "Cache"). **You MUST make sure that the project name, the top level entity name, as well as the .vhd that will include the code for your entity have the SAME name, that being "Cache".** If you don't use the same name, then your project won't compile and you'll get strange compilation errors.

Click on next and you'll be brought to the page as shown below:



This page asks you which files you want to include into your project. If you have already made .vhd file(s) prior to creating the Quartus project, then search for the files in your hard drive that you want to add to the project by clicking on the icon with the 3 dots. If you haven't made any .vhd files yet, just click **next**.

Next you'll be brought to a screen asking you to choose which device you want your VHDL code to compile to, as shown below:



Note that in the web version of Quartus II, only Cyclone family of Altera FPGAs are available.

You just made a Quartus project. This is a crucial step and many students think they can bypass creating a project and expect that opening a .vhd file in Quartus and clicking "compile" will automatically compile their .vhd file. Clearly, that won't work. You must create a Quartus project first.

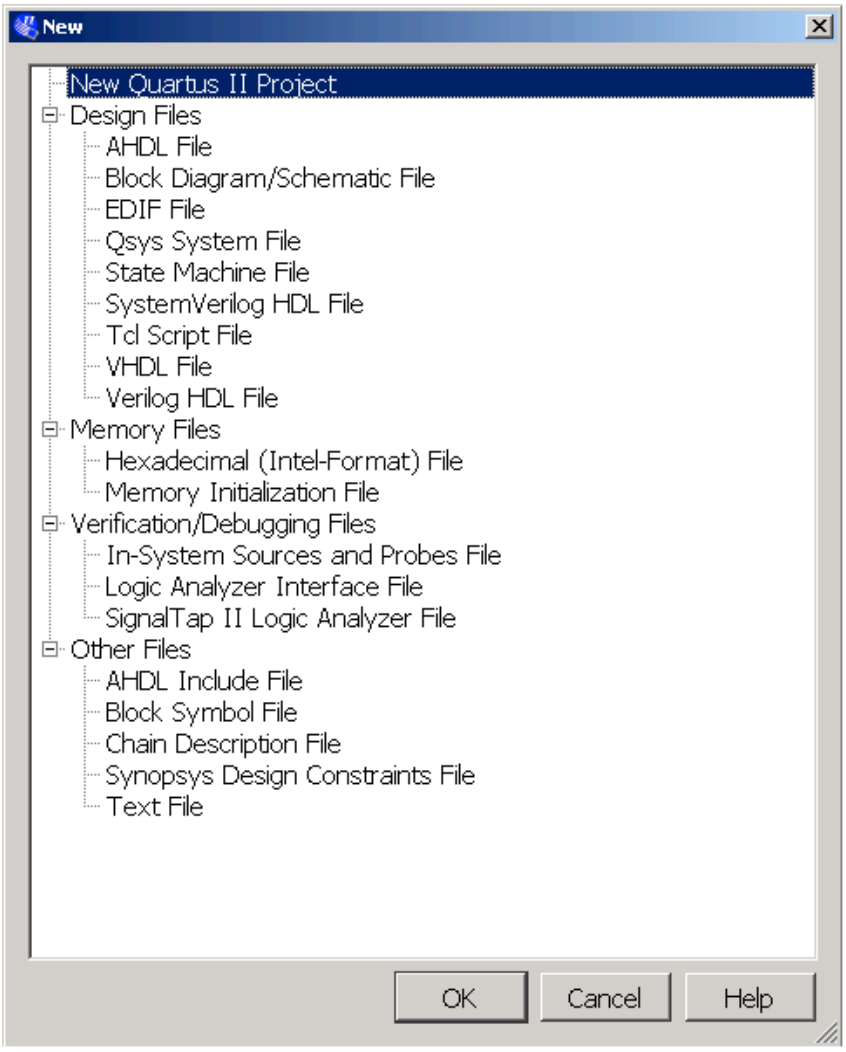
Summary

So here are the most important points to get out of this section:

1.) you must create a project in Quartus
2.) the name of the Quartus project, the name of top level entity, and the .vhd where you put your VHDL code for the top level entity must all be the same

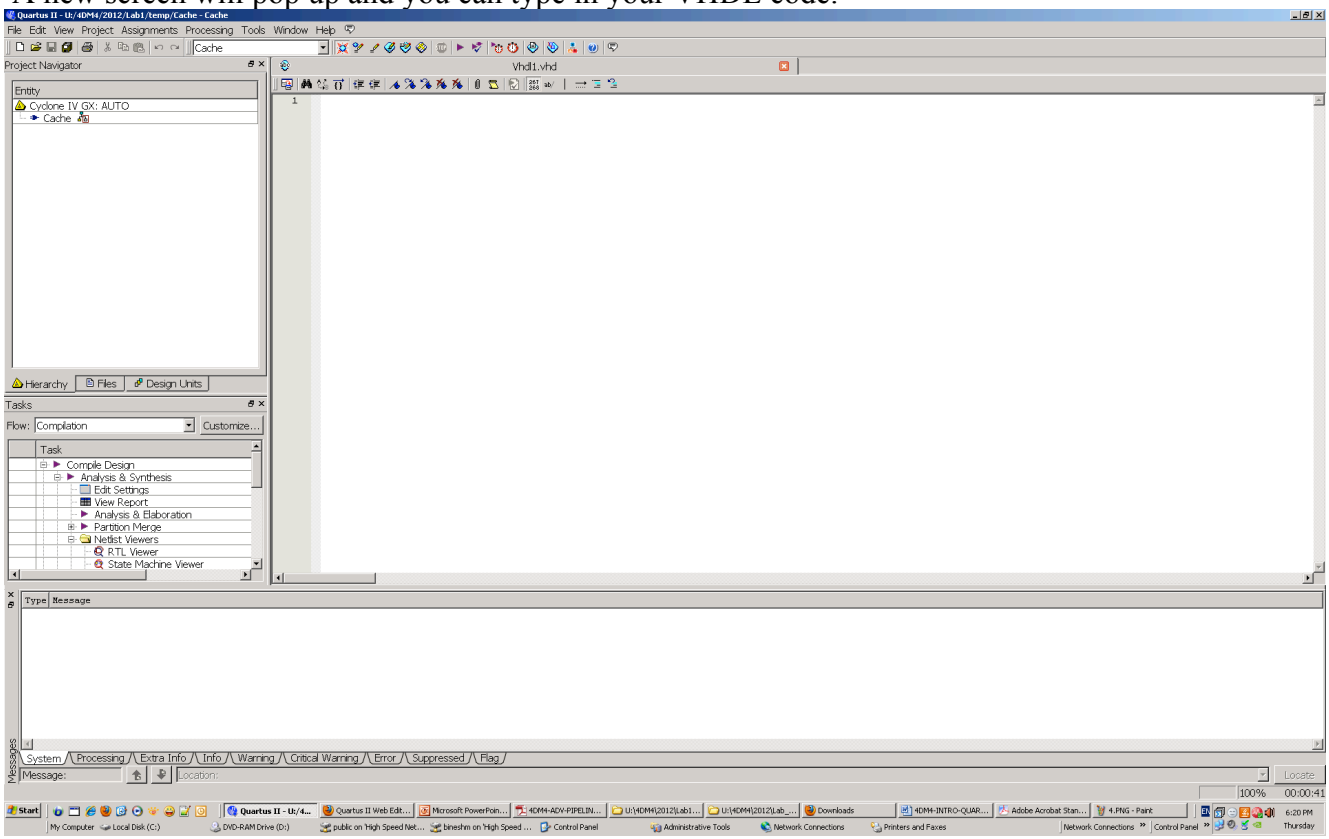
3.) Creating a new VHDL file or Opening an existing VHDL file in Quartus

Before going ahead with this step, make sure that you have a Quartus project already opened. To create a new VHDL file, go to **File→New**, and the following dialog box will appear:

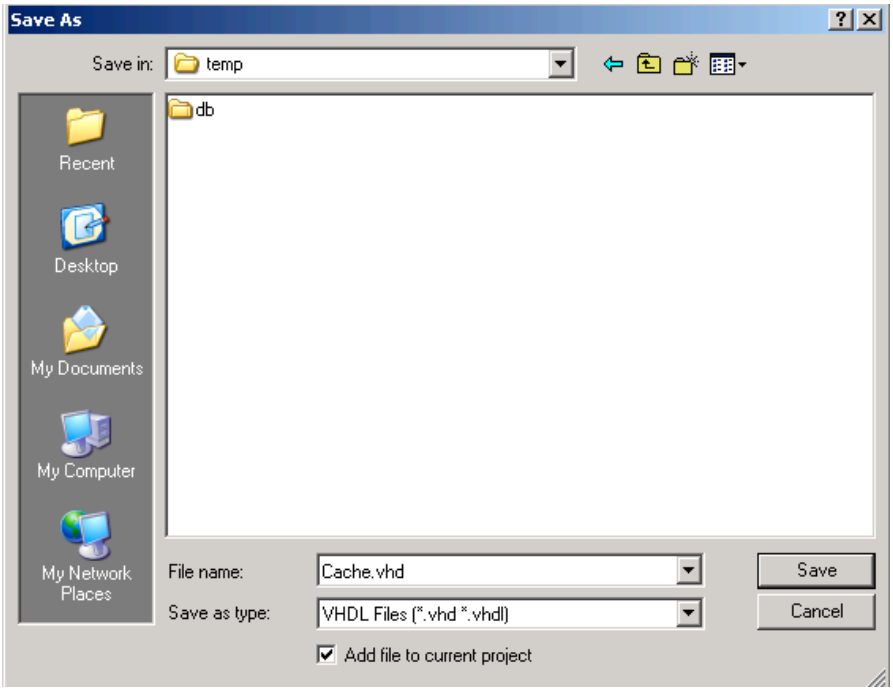


Choose "**VHDL File**" and click **ok**.

A new screen will pop up and you can type in your VHDL code.

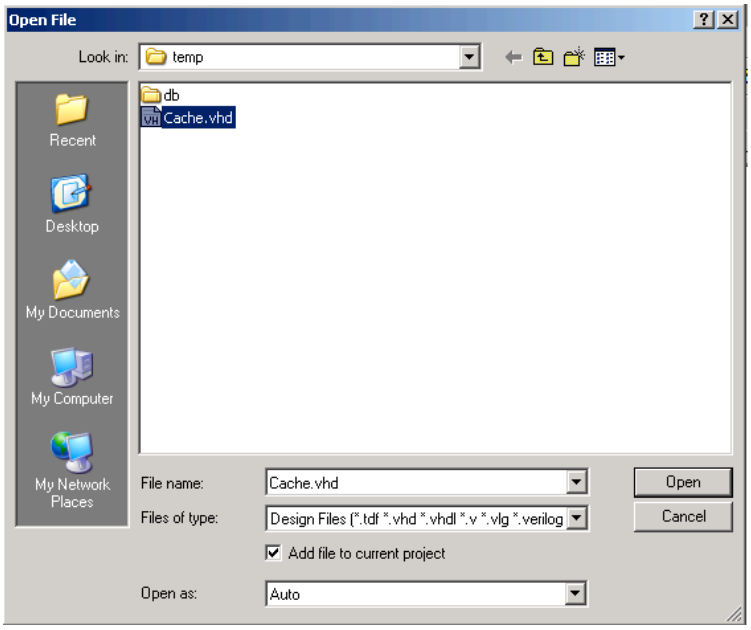


You can see in the figure above that the default name of the vhd file is Vhd1.vhd. Go to **File→Save** and save the .vhd file to SAME name as your project.



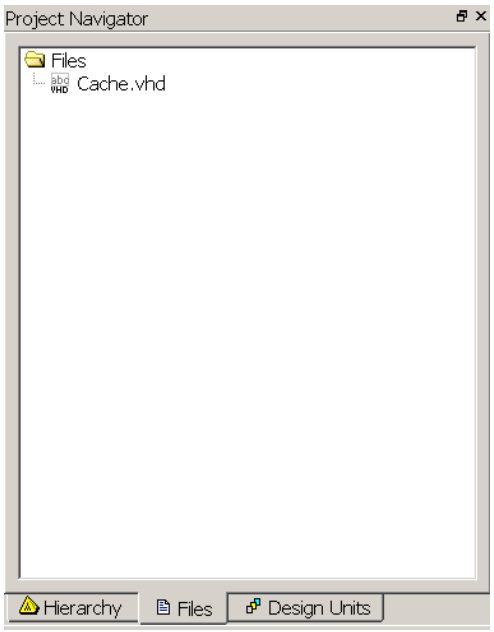
Make sure "**Add file to current project**" is checked; otherwise, this .vhd file will not be associated with the project and you will not be able to compile this VHDL file. (This is a common problem with many students in the lab—they forget to click "**Add file to current project**" so when they compile, they get weird errors about not being able to find the top level entity).

To open an existing .vhd file and to have it associated with your project, go to **File→Open**. A dialog will appear as follows:



Search for the file that you want to add to the project, and REMEMBER to check the "**Add file to current project**" checkbox. Click **Open** or press enter and your .vhd file should now appear in Quartus.

You can verify with yourself in Quartus to see if your .vhd files are indeed linked to the project by going to **Project Navigator** in the Quartus workspace, as shown below (you may need to click on the middle tab to see it):



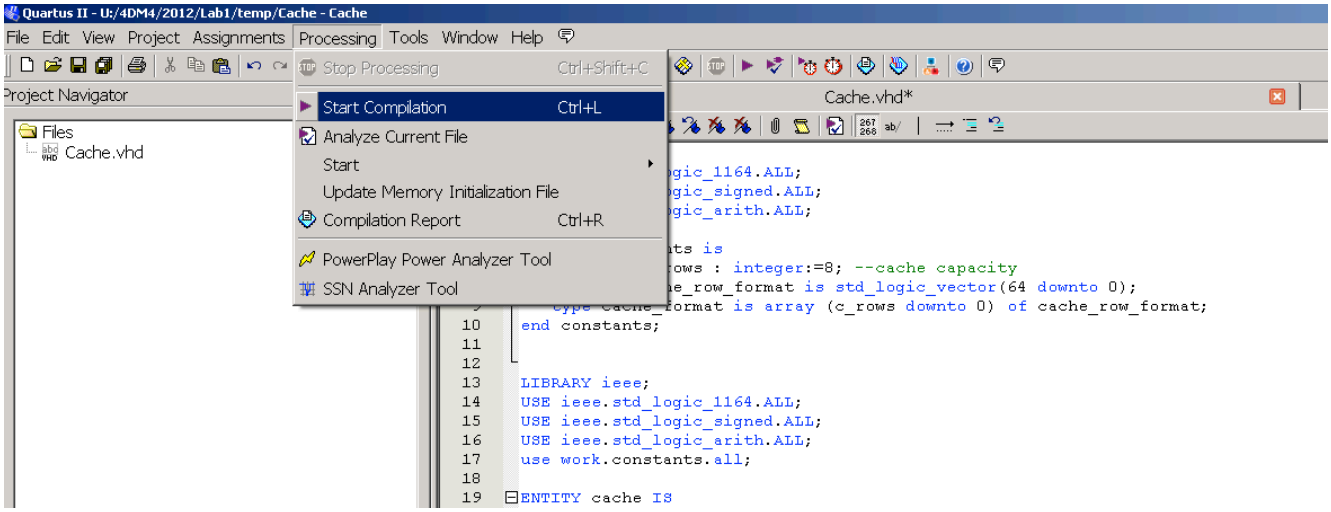
If you don't see your file listed in the **Project Navigator**, then that means the files you created/opened aren't linked to the project, which means that you won't be able to compile your newly created nor your opened .vhd files. If this is the case, go to **File→ Open** and open your files again, making sure that you do check on "Add **files to Current Project**".

Summary

-Make sure you click on "**Add files to Current Project**" when opening a .vhd file!

4.) Compiling in Quartus

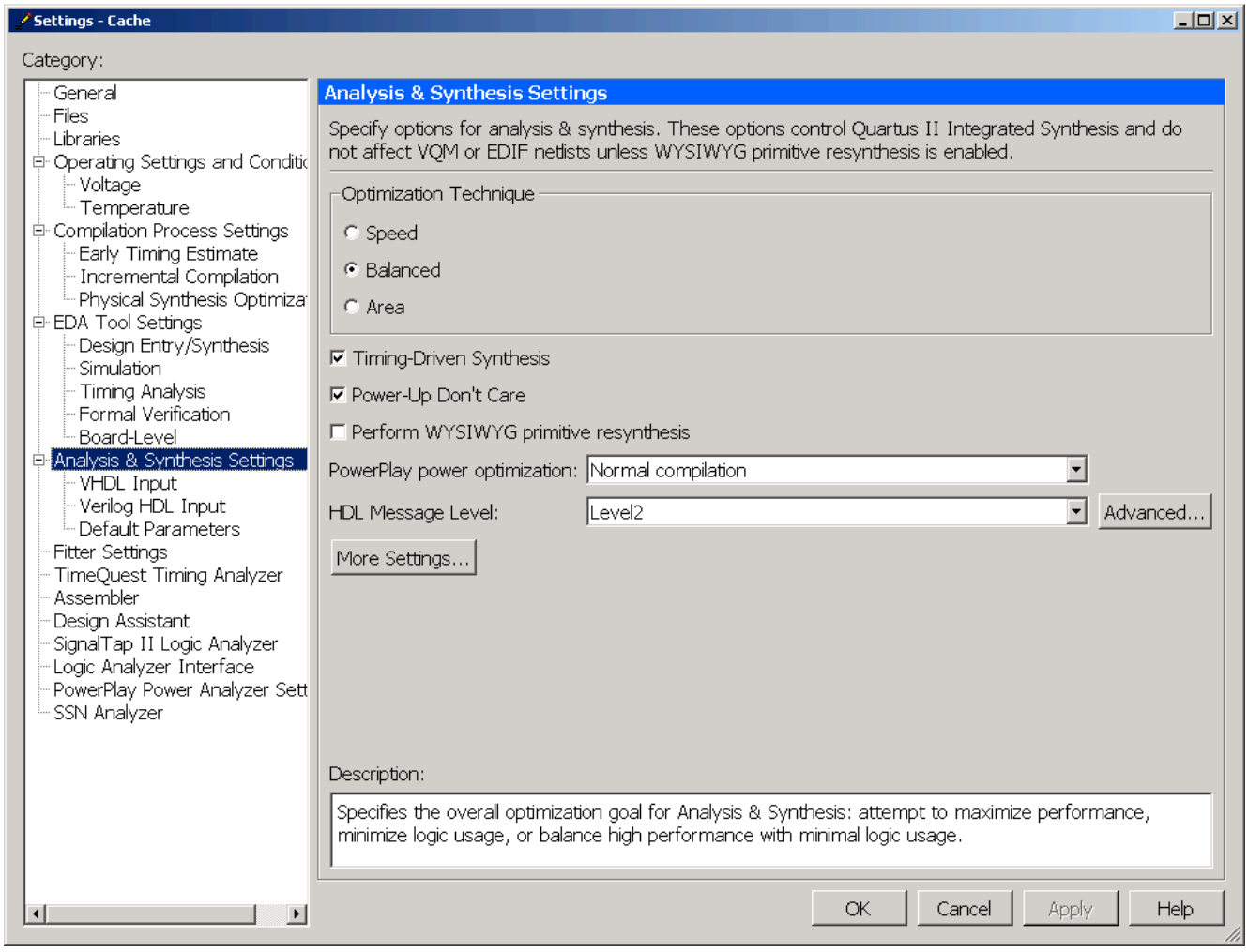
Now, you're ready to compile your code (.vhd file(s) associated with a Quartus project). For compilation you can go to **Processing→Start compilation** as shown below.



You can also press the PLAY icon on the menu bar.

5.) Analysis and Synthesis Settings

To change Analysis & Synthesis settings, on the **Assignments** menu, click on **Settings**, and then select **Analysis & Synthesis Settings**.

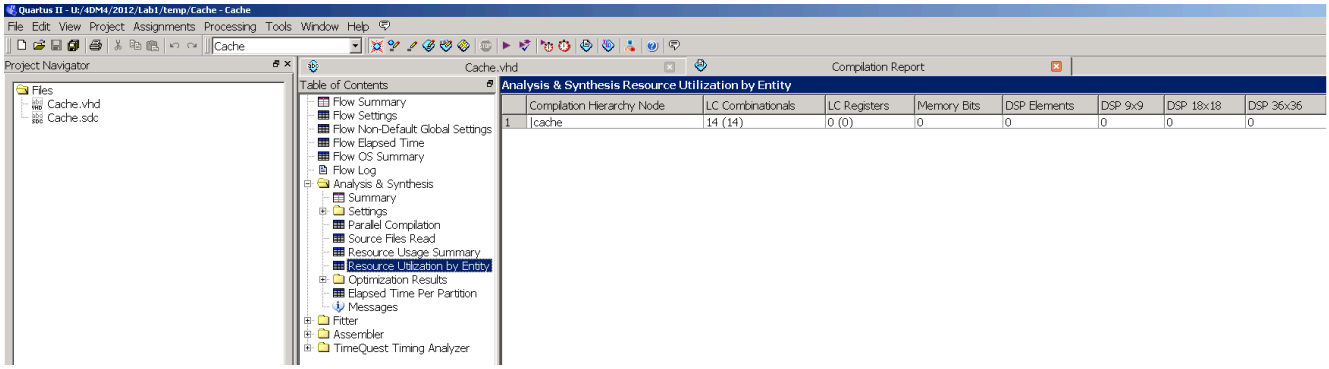


Quartus II offers different optimization techniques. If resource utilization is an important concern for you, select **Area** for the Optimization Technique. If clock frequency is important for you, select the **Speed** Optimization Technique option to specify that all combinational logic in or between the specified clock domain is optimized for speed. For the best trade-off between area and speed, select the **Balanced** Optimization Technique. The **Balanced** setting typically produces utilization results that are very similar to what produced by the Area setting, but with better performance results.

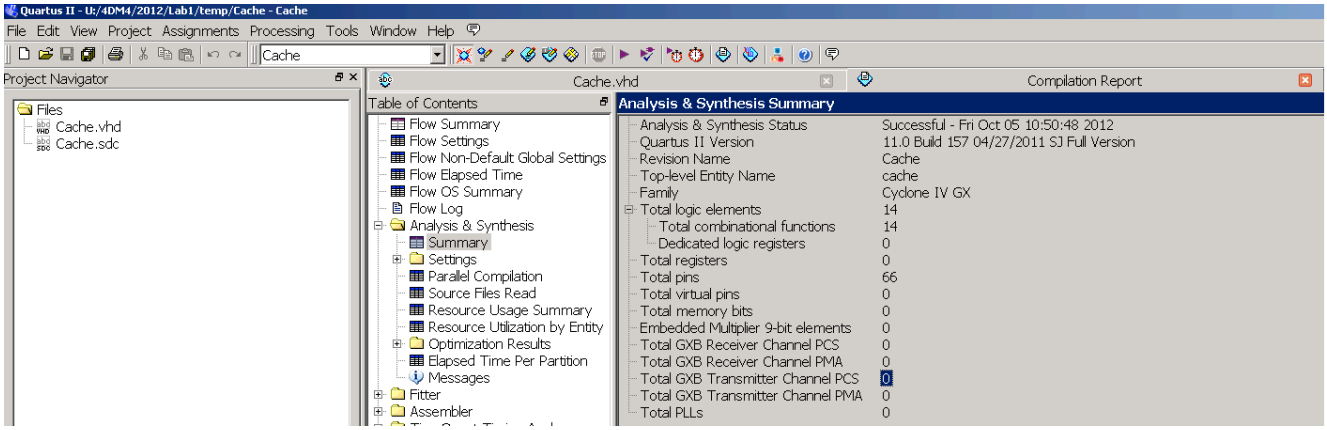
You can also click on “More Settings” to access more options.

6.) Compliation Report:

After full compilation, click on the “Compilation Report” tab and go to the “Analysis & Synthesis Resource Utilization” section which yields the screen shown below. It shows the number of LUT-only LC’s (combinational) and Register-only LC’s. The ‘Logic Cells’ in Cyclone-based FPGAs are considered as Logic Elements (LE), and in Stratix-based FPGAs are considered as Adaptive Logic Module (ALM).
The upper-bound of total LC’s is the sum of the two. However, the synthesizer sometimes fits one combinational LC and one Register LC into one LE or ALM.

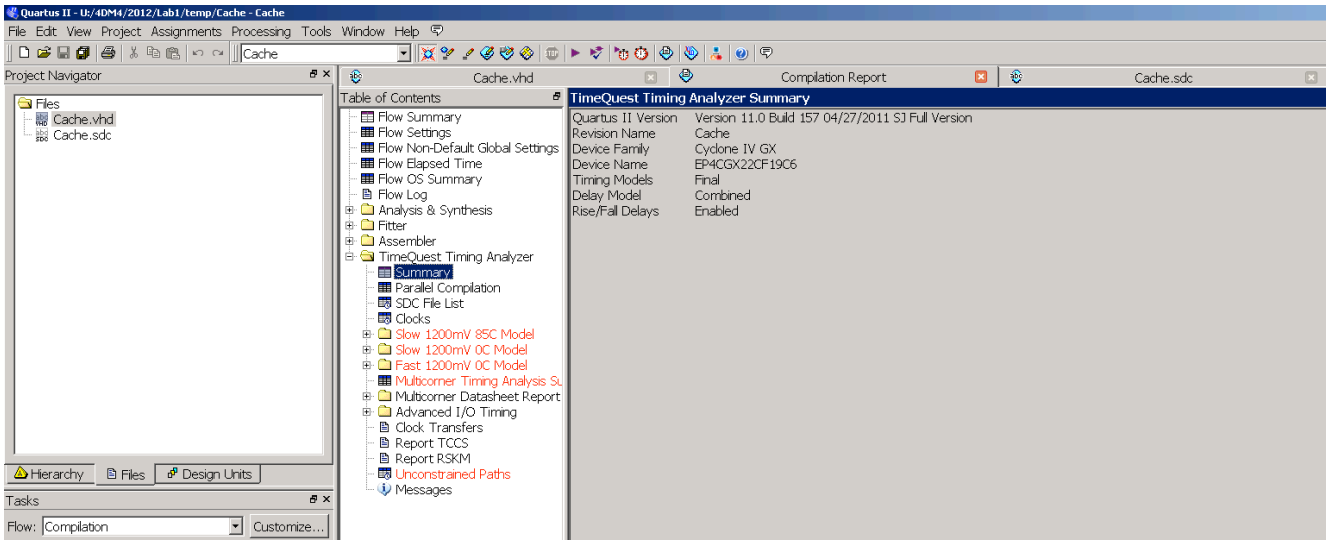


Below shows an example of analysis & synthesis summery in Cyclone family of FPGAs:



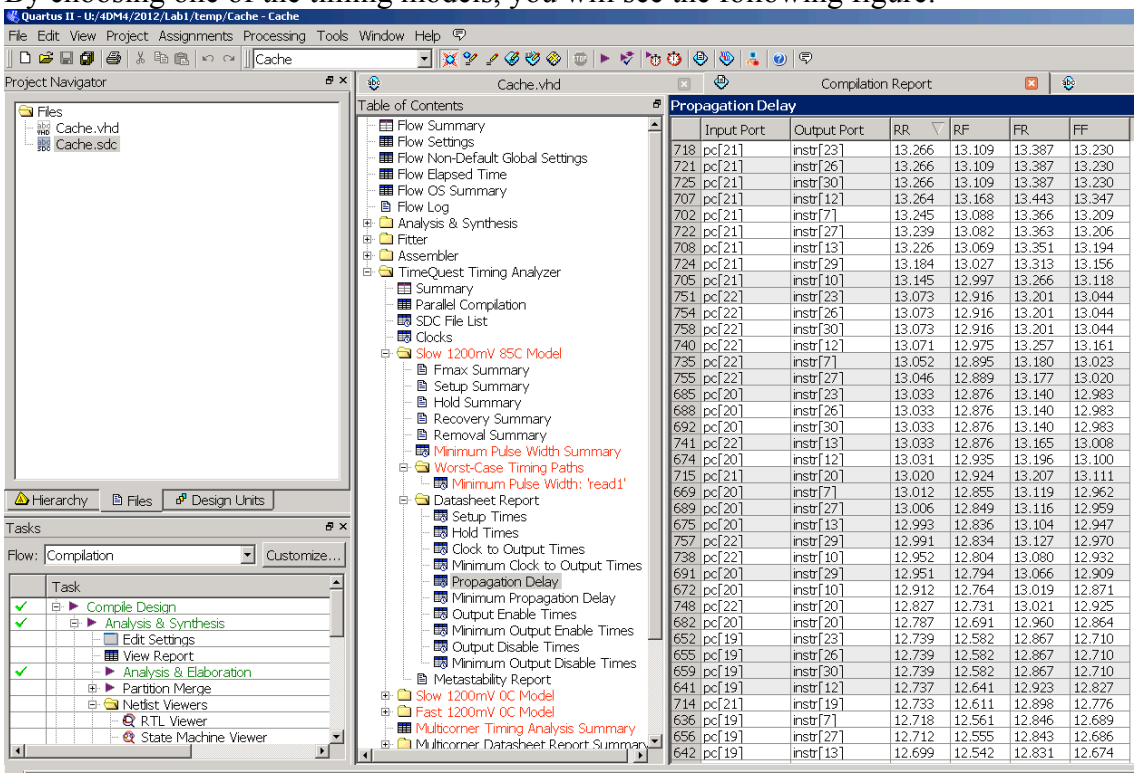
7.) Timing Analysis:

After full compilation, from Compilation Report go to “TimeQuest Timing Analyzer” which is shown below:



The TimeQuest timing analyzer offers you some timing models (slow & fast) depending on the selected device. In each model, the timing specifications for an environment condition are reported. For example in the figure shown abive, the first model applies when the power voltage is 1200mV and device temperature is 85 degree Celsius.

By choosing one of the timing models, you will see the following figure:



If your design has a clock, you can see setup times, hold times, and maximum frequency of your design. Otherwise, you should go to the **Datasheet Report** and see the timing results. The helpful reports are as follows:

1. **Fmax**: The maximum clock frequency that can be achieved without violating internal setup ($\mathbf{t_{SU}}$) and hold ($\mathbf{t_H}$) time requirements.
2. **Setup Times**: Reports the setup times for clocks in the design.
3. **Hold Times**: Reports the hold times for clocks in the design.
4. **Propagation Delay**: Reports longest delay in nanoseconds between the edges of a signal propagating from an input port to an output port.
    a. RR shows the longest delay measured from rising edge to rising edge
    b. RF shows the longest delay measured from rising edge to falling edge
    c. FR shows the longest delay measured from falling edge to rising edge
    d. FF shows the longest delay measured from falling edge to falling edge

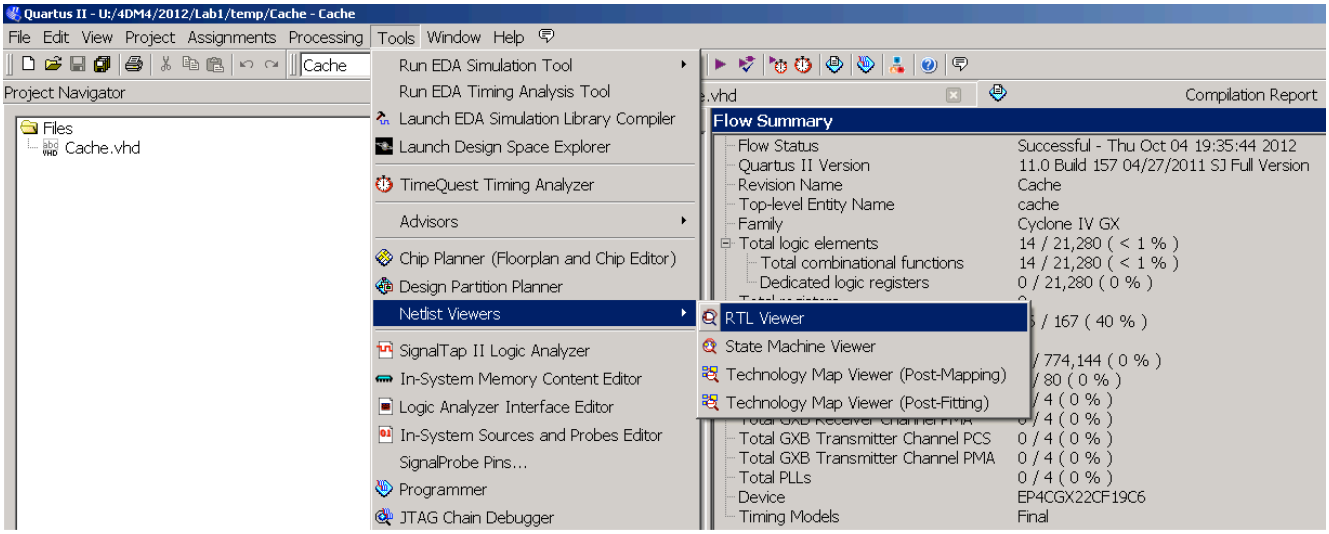
    Note that, in the propagation delay the highest delay (Worst-Case) is important.

In the Quartus II you can specify some timing constraints in your design which describe the clock characteristics, timing exceptions, signal transition arrival and required times. These constraints should be written with Tcl commands (Tool command Language) and should be saved in a file, named Synopsys Constraint File (SDC).

The Quartus II assigns a default frequency of 1GHz (1ns) for clocks that have not been constrained, unless any constraint exists in the design.

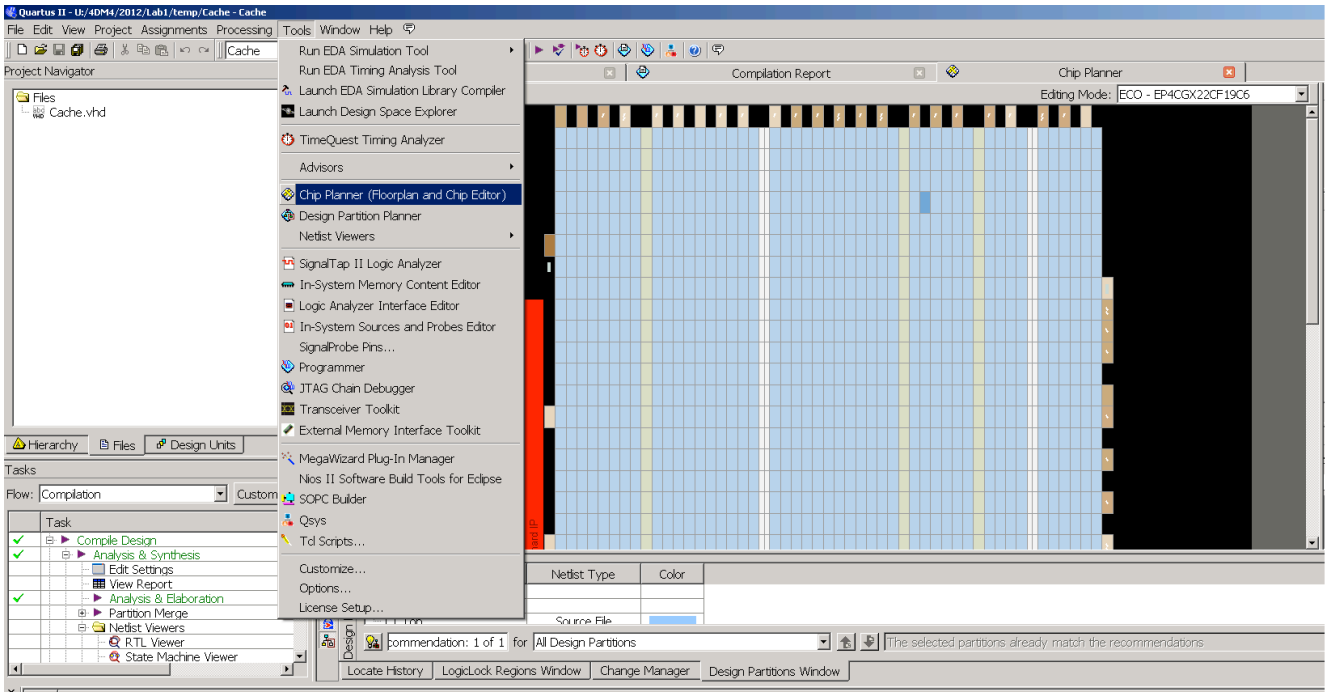
8.) RTL Viewer:

RTL Viewer can be accessed through the **Tools** menu and open "**Netlist Viewer**" then click on **RTL Viewer.**

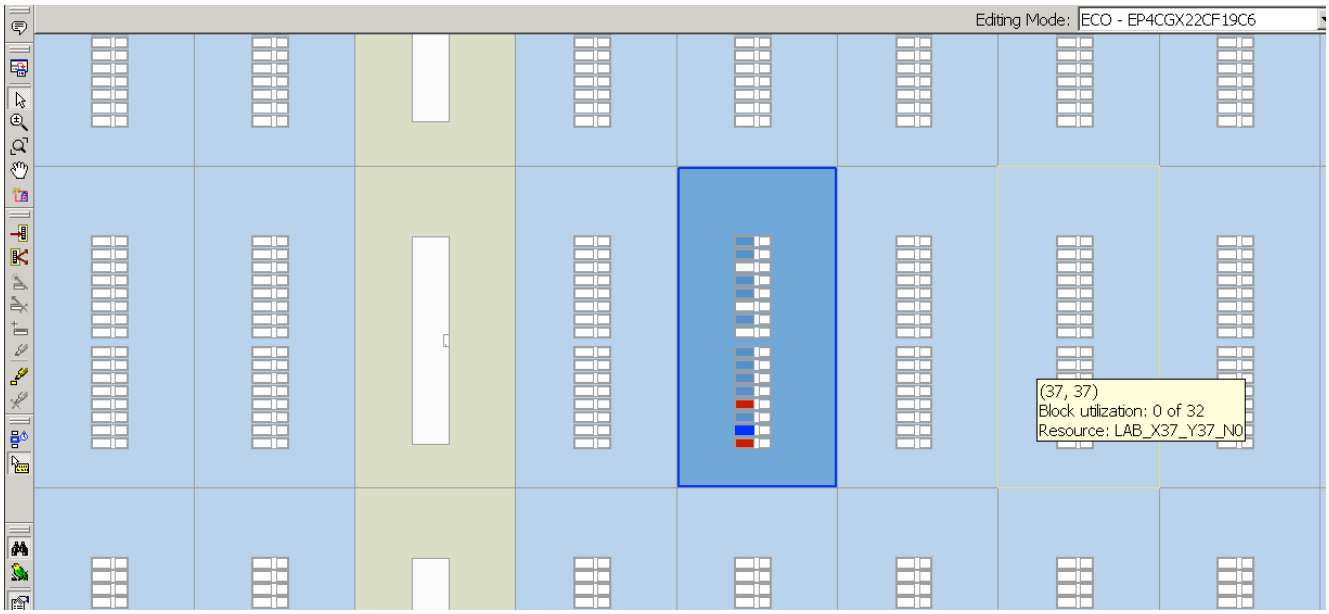


9.) Floorplan Editor:

Floorplan editor can be accessed through the **Tools** menu and click "**Chip Planner**".



In this figure, each rectangular denotes an Array Logic Block (ALB). In each ALB, there are some LEs or ALMs. Each dark blue rectangular represents a used ALB in your current design. If you zoom-in the floorplan, as shown in below, you can see the used ALMs or LEs in an ALB.



Hopefully, this tutorial has given you enough information to get started with both VHDL and Quartus.