Quartus
New Project Tutorial

This tutorial guides you through the typical steps of design, simulation, synthesis and implementation of a simple project, designed from scratch, as those that you will do at the Applications for the Digital Integrated Circuits course.

Contents

Project File Management

The normal design flow goes through all stages in Figure 1. The Quartus II software arranges these stages in the *Compilation Flow* and guides you through it.

Figure 1: Quartus II Compilation Flow

Various files are employed in each design flow stage, all of them created and managed by Quartus, except for the source files, that are written by you, the designer. All files created by Quartus reside in the project folder (directory) and its subfolders. Your source files should also be saved inside the project folder.

To keep things as neat as possible, strictly follow these two rules:

- **Each project has its own folder.**
- **All (Verilog) source files are saved in the project folder.**

If the above rules are obeyed, the project folder should have the structure shown in Figure 2.

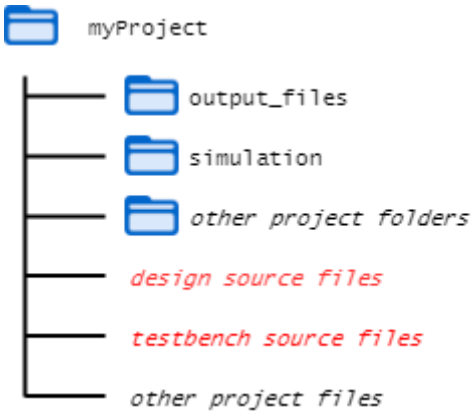


Figure 2: Quartus project folder structure.
Files shown in red are those that you create and edit.

Most of the files are taken care of by Quartus, so you need not bother about their names or where they are located. The only files you should care about are:

- design source files
- testbench source files

Remember:

- **All design source files and testbench source files should be saved in the project folder.**

Create a New Project

Open the New Project Wizard either by clicking on its icon in the start window, or from the menu **File** -> **New Project Wizard …**, or, also from the menu, **File** -> **New …** -> **New Quartus II Project**.

In step 1 [Directory, Name, Top-Level Entity] of the New Project Wizard (Figure 3) set the name of the project folder (the working directory), the name of the project, and the name of the top-level design module. First, select a folder where you want your project's folder to be created (for example the folder `student`), then type a slash (/), followed by the name you choose for your project folder (`app1` for example).

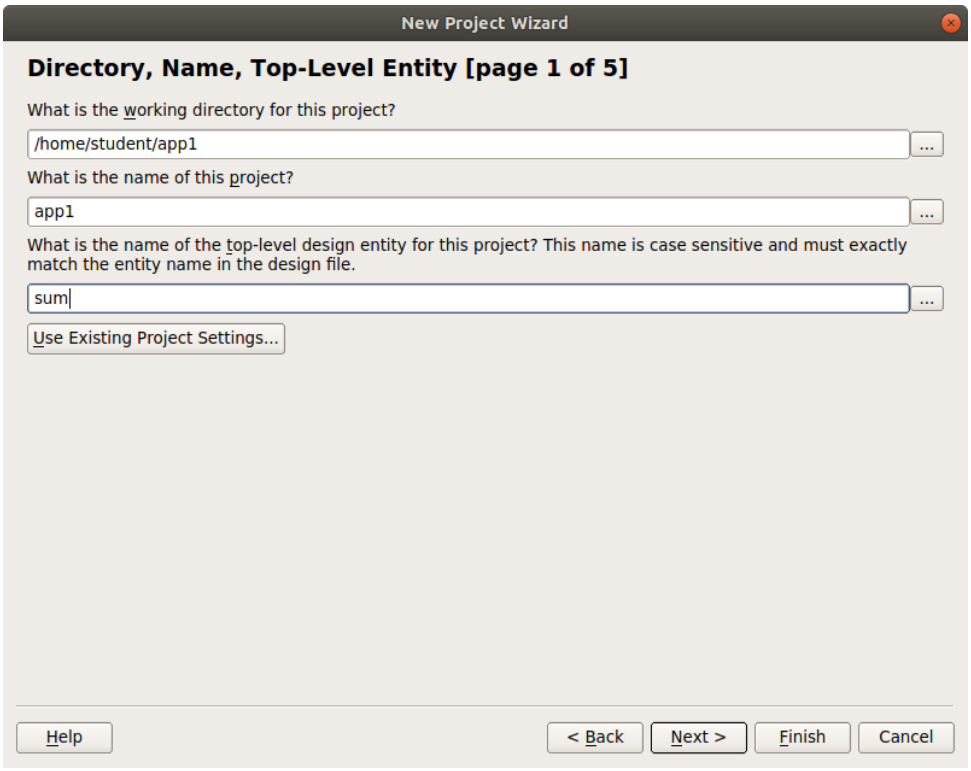


Figure 3: New Project Wizard step 1 (name and folder setup)

At this stage, the project is empty and no file is added. Skip this step and click **Next**.

In step 3 [Family & Device Settings] of the New Project Wizard select the 5CSEMA5F31C6 device from the list (Figure 4). Then click **Next**.

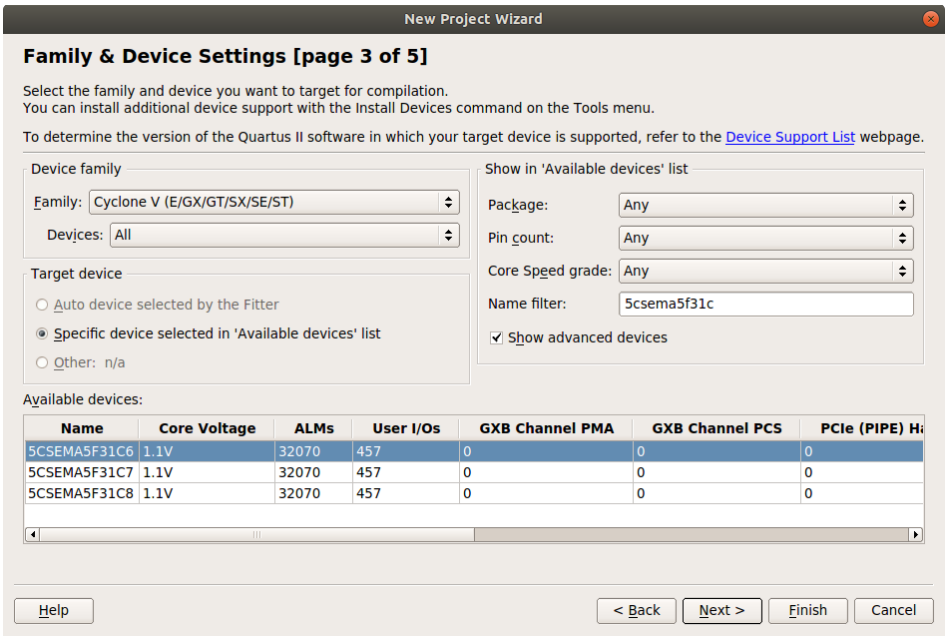


Figure 4: Device selection

In New Project Wizard step 4 [EDA Tool Settings], change the Format for Simulation from **VHDL** to **Verilog HDL** in the drop-down list (as in Figure 5), then click **Next**.

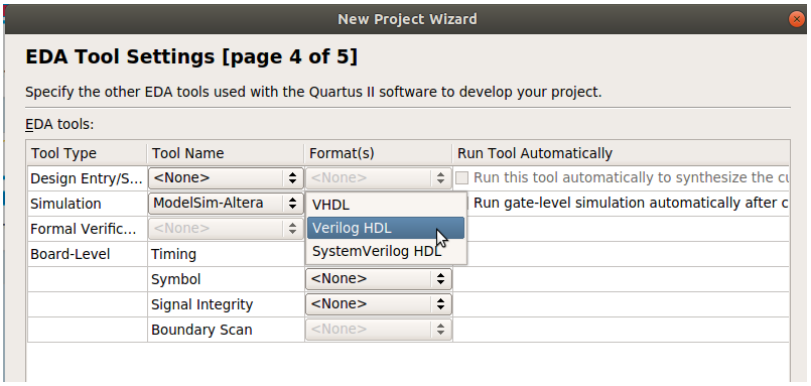


Figure 5: Choose Format for Simulation

Click **Finish** to close the New Project Wizard.

Create Source Files

To create a new design source file, select from the menu **File** -> **New** (Figure 6).

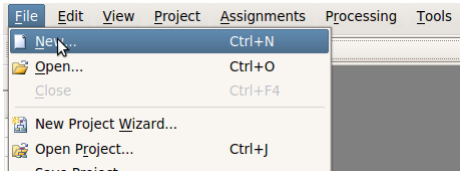


Figure 6: Open a new file

In the popup window that opens (Figure 7), select **Verilog HDL Files** and click **OK**.

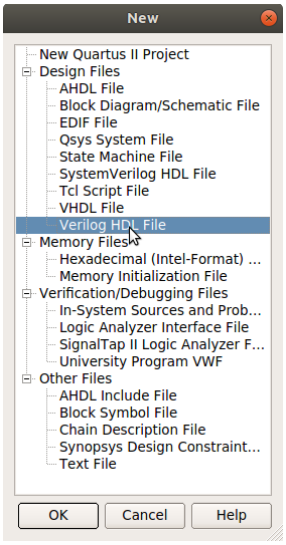


Figure 7: New file type selection

All source files that you will write during Applications, be they design source files, or testbench source files, will be of Verilog HDL type.

After you have clicked **OK**, the file opens in the Editor panel, with a default name, Verilog1.v (or Verilog2.v, or other automatically numbered default name). It is recommended that you immediately save the file with a suitable name. From the menu select **File** -> **Save As …** (Figure 8).

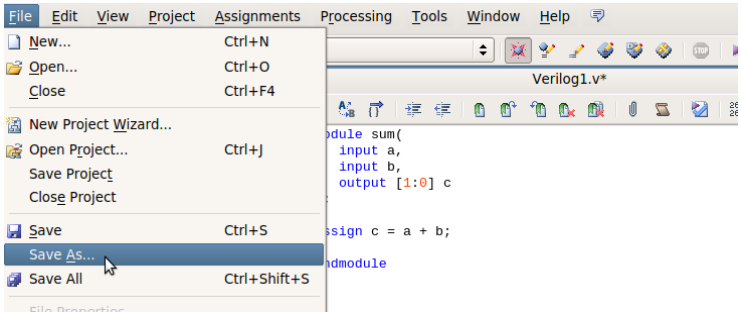


Figure 8: First save of a source file

A file browser opens (Figure 9). If you followed strictly all previous steps as recommended, the browser will open showing the content of the project folder. You need only to type the desired name for the source file, then click on **Save**, and the file will be saved in the project folder.

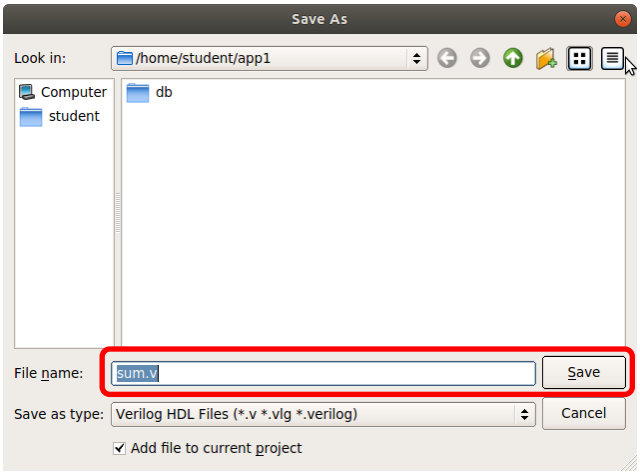


Figure 9: Choose a filename for a source file

Note: If the browser opens in some other folder, browse through the folders and select the project folder. You should see the project folder path name (for example `/home/student/app1` if you setup the project folder to be `app1` in the `student` home directory) in the **Look in** field at the top of the Save As window.

Keep in mind these rules:

- **Each module is in a separate source file.**
- **The name of the source file is the name of the module.**
- **Save any source file in the project folder.**
- **The top-level module name and its source file name must match the names you chose when the project was setup.**

First Time Analysis

This step is necessary in order to be able to start the simulator from Quartus. When you do this first time, it's recommended that the design is at a minimum, such that Analysis quickly finishes and passes without errors, or if there are errors, they are easy to fix. Once the simulator starts, you may play with the source file code without the need to run through Analysis every time you change something.

In the Tasks panel switch the Flow to RTL Simulation (Figure 10):

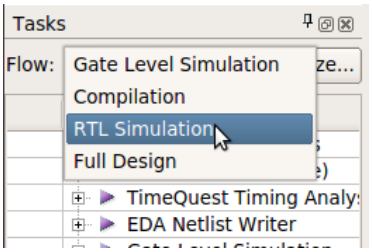


Figure 10: Choose the flow.

The RTL Simulation flow has only two tasks. Double-click on Analysis & Elaboration task (Figure 11).

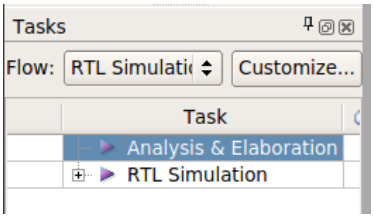


Figure 11: Start Analysis and Elaboration

If there are no errors in your design, the analysis should finish with 0 errors reported in its last message:

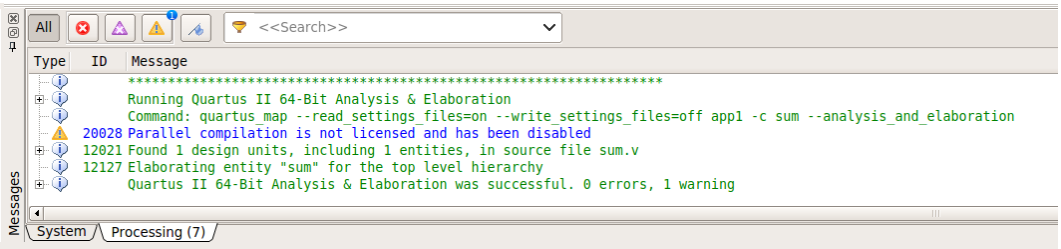


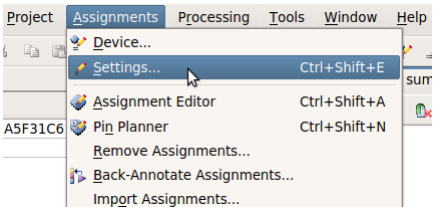
Figure 12: Successful Analysis & Elaboration

If, instead, there are errors, you must fix them and run Analysis & Elaboration. Repeat this procedure until no errors reported, otherwise the simulator cannot be launched from Quartus.

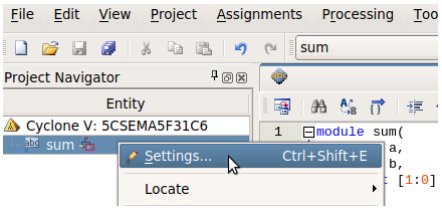
Set up Simulation

Prior to start the simulator, you need to specify what is the testbench file to be run.

From the menu select **Assignments** -> **Settings…** (a) or click right on any item in the Project Navigator panel and select **Settings…** (b).



a)



b)

In the Settings Simulation window (Figure 13), change **NativeLink settings** from **None** to **Compile test bench**. If the testbench is not yet set, click **TestBenches…** to select a testbench.

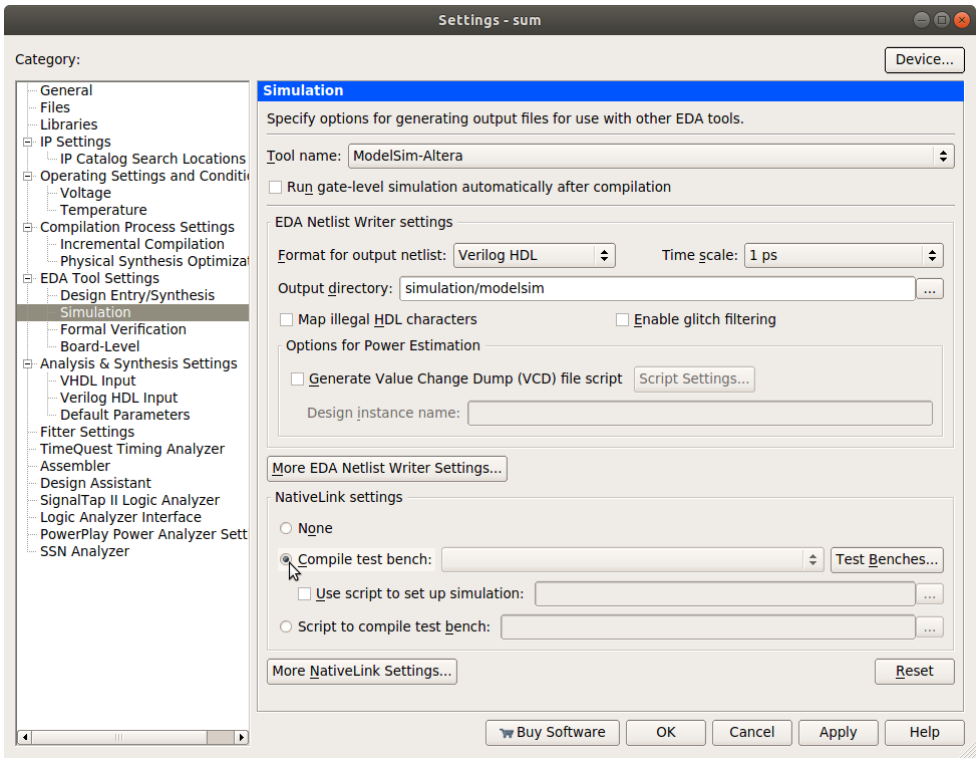


Figure 13: Change NativeLink settings from None to Compile test bench

A popup window appear that shows testbenches of the project (Figure 14).

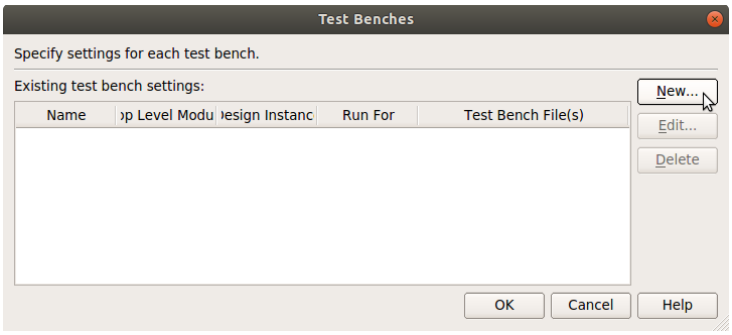


Figure 14: Testbench settings

The project being totally new, there is no testbench in the list yet. If you have already created and saved a source file for the testbench, it needs to be added to the project in this step. Click on **New…**, to add it to the list of testbenches. A new window opens that enables you to set this testbench (Figure 15).

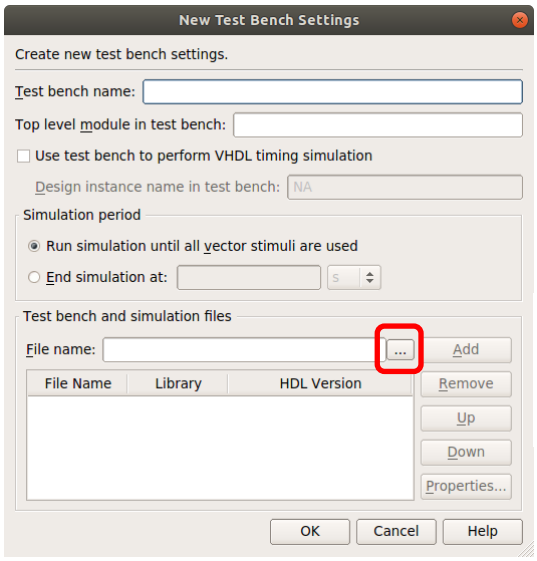


Figure 15: Add a new testbench to the project

First of all specify the file name of the testbench source file. Click on search button to open the file selection dialog box. The file selection dialog box will open in the project folder (Figure 16). Select the testbench source file and click **Open**.

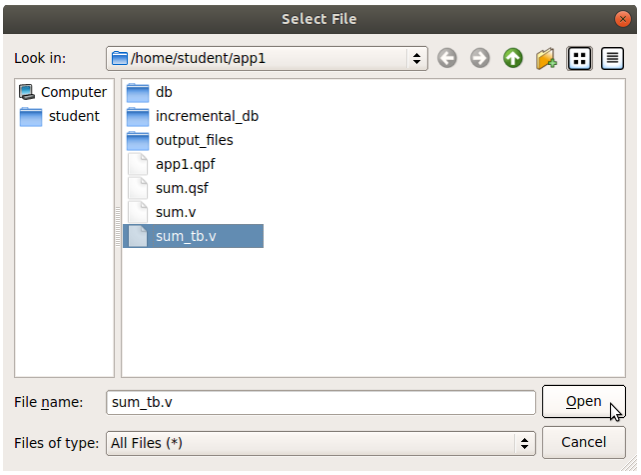


Figure 16: Select a testbench file in File selection dialog window

The File Selection dialog closes and the selected testbench file is listed in the File name field. Click **Add** to add the testbench file to the list. Then type the filename (without extension!) in the **Test bench name** field at the top of the New Test bench Settings window. Supposing you saved the testbench file with the same name as that of the testbench module, the **Test bench name** will be mirrored in the text field below. After that, the New Test bench Settings window should look as in Figure 17.

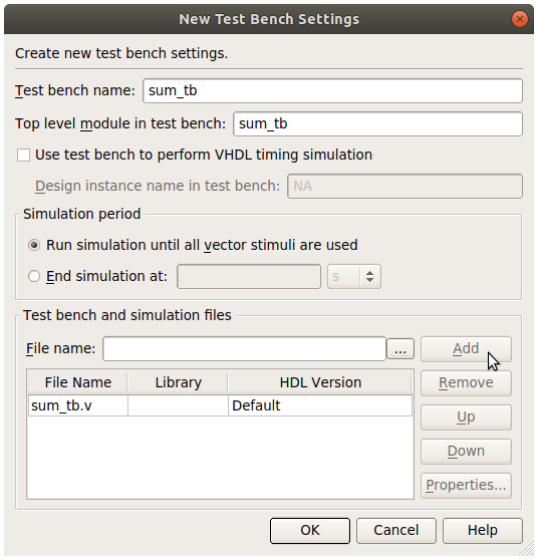


Figure 17: New Test Bench Settings after completion

Click **OK**. The New Test Bench Settings closes and the newly selected testbench appear in the list of Test Benches window (Figure 18).

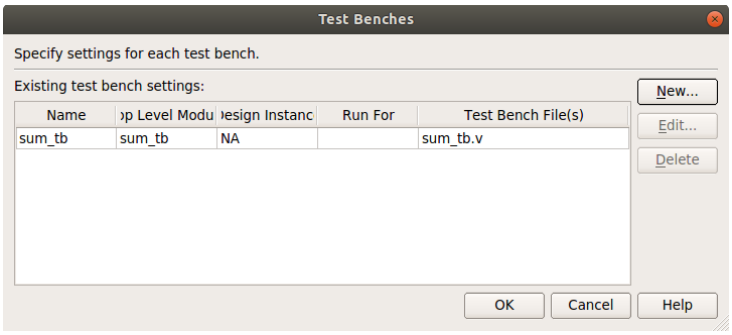


Figure 18: Test Benches list after the testbench was added

Click **OK** to close the Test Benches dialog box. The Simulation settings should now show the testbench in the **Compile test bench** field, as in Figure 19.

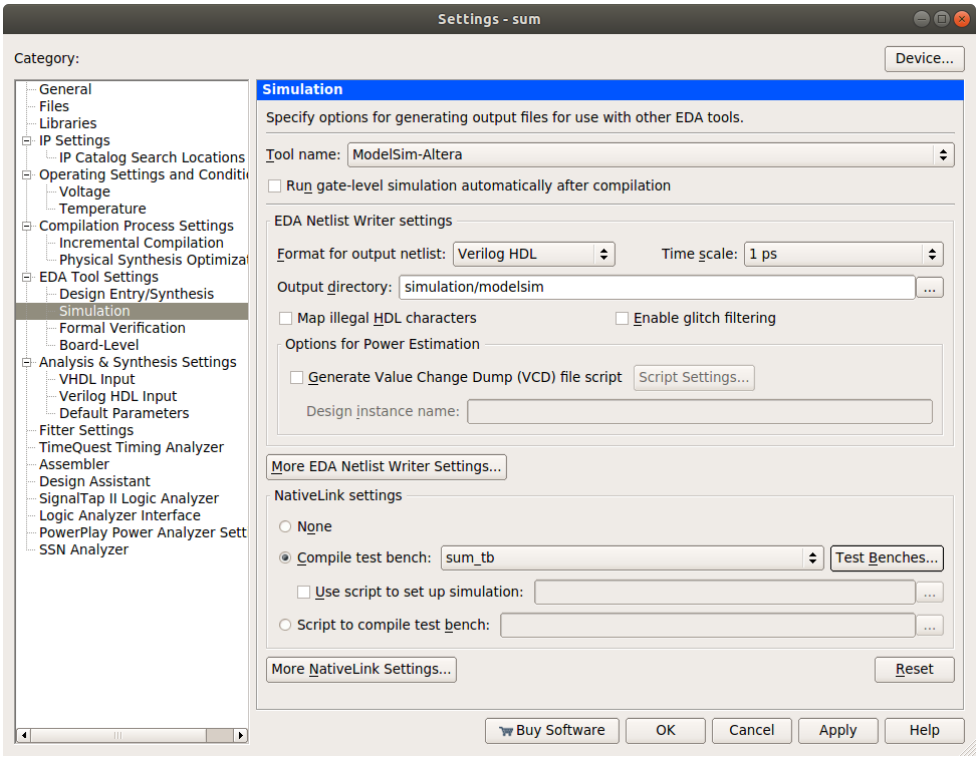


Figure 19: Simulation settings completed

Leave everything else as default and click **OK** to finalize simulation setting. You are now ready to start simulation.

Start Simulation

From the menu select **Tools -> Run Simulation Tool -> RTL Simulation** (Figure 20).

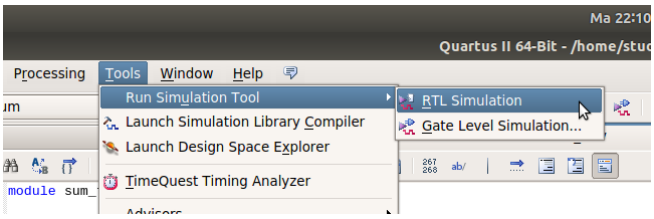


Figure 20: Start RTL Simulation from menu

You may, instead, click on RTL Simulation button in the toolbar (Figure 21):

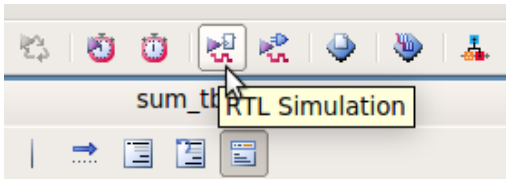


Figure 21: RTL Simulation button

The ModelSim-Altera simulator is launched (Figure 22). It takes a couple of seconds for it to open all its windows. If everything is correctly set up and the testbench file has no errors, the Objects panel is populated with testbench signals and the Wave panel will show their waveform.

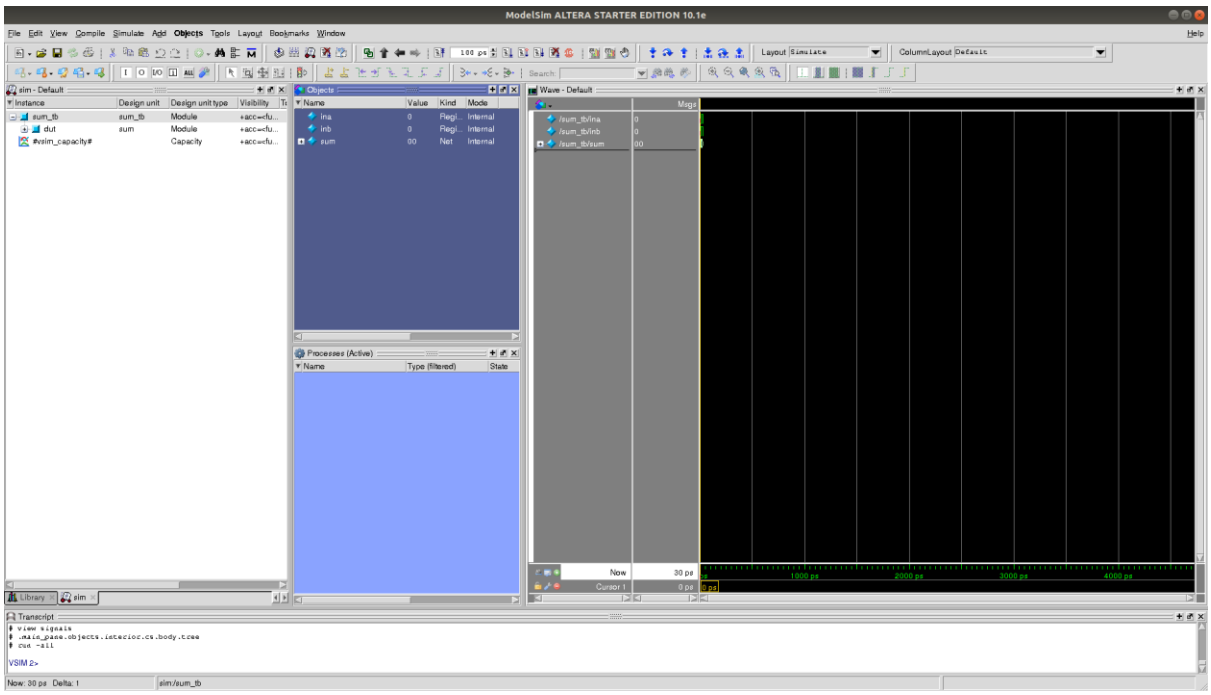


Figure 22: ModelSim-Altera simulator window right after launch

Note: If there are no signals in the Wave and Object windows, that means the testbench has not compiled because of compiling errors, or it has not loaded because the module it instantiates does not match the compiled top-level design module. In the Browser panel at the left of the simulator window you will see some instances, but not the testbench instance. In that case you must look for the errors in the Transcript panel at the bottom of the simulation window, close the simulator, fix the errors in the testbench and then relaunch the simulator.

Pin Assignment

TBD

Compilation

TBD

Device Programming

In the Tasks panel double click on **Program Device** (or, alternatively, select **Tools -> Programmer** from the Menu bar). The Programmer window will open (Figure 23).

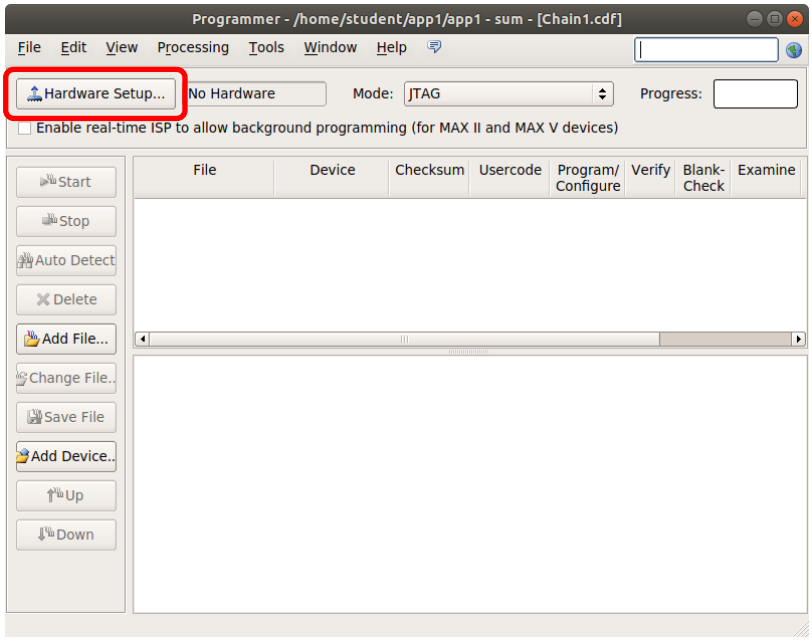


Figure 23: Programmer window

If there is nothing in the panels of the Programmer window, the first step is to connect to the physical board. Click on **Hardware Setup** button as shown above in Figure 23. The Hardware Setup dialog window shows up (Figure 24).

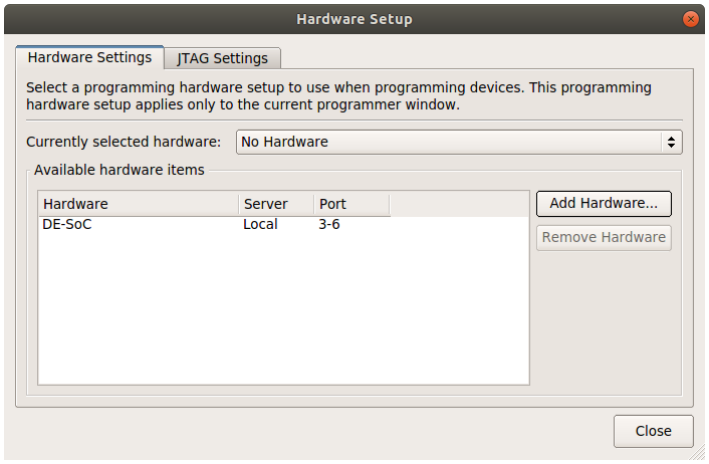


Figure 24: Hardware Setup window

There is no hardware shown in **Currently selected hardware** field, but there should be one in the **Available hardware items** list, as shown in the caption above.

⚠**Attention!** If the device doesn't appear in the list or the list is empty, make sure that you powered up the development board (otherwise the Quartus II could not connect to it), or that it is properly connected to your PC.

In **Currently selected hardware** field select from the drop-down list the De-SoC device, as shown in Figure 25 below:

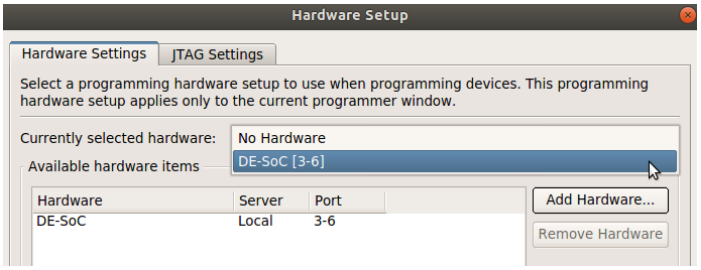


Figure 25: Hardware selection

Click on **Close** button. Now the Programmer Window shows the De-Soc device and also has the *Auto-Detect* button enabled (Figure 26).

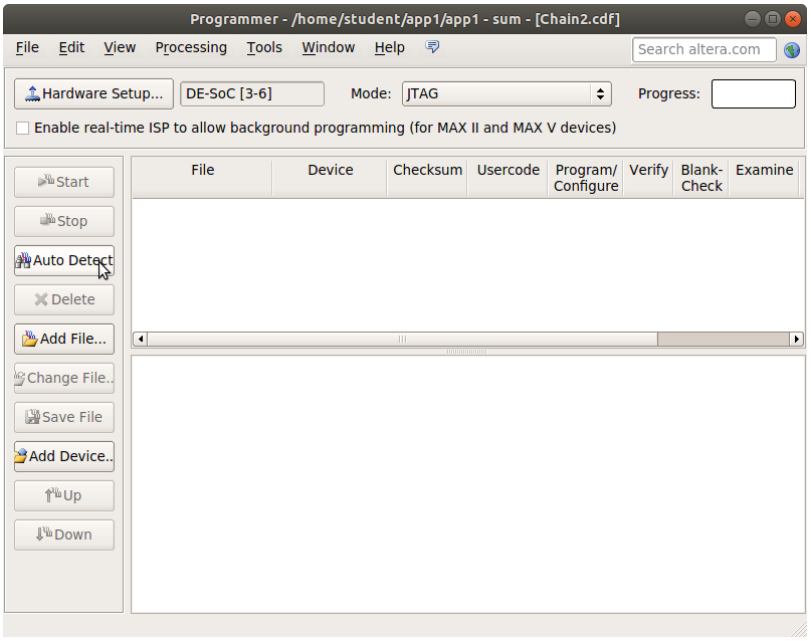


Figure 26: Programmer Window ready to configure the device

Click on the **Auto-Detect** button as shown above. A small pop-up window (Figure 27) appears where you should check the right device.

Figure 27: Select Device pop-up window

Check **5CSEMA5** device, as it is the type of Cyclone V FPGA on the development board, and then click **OK**.

Now the Programmer window panels are filled up with a chain of two devices detected on board, and two entries that allow you to program them (Figure 28).

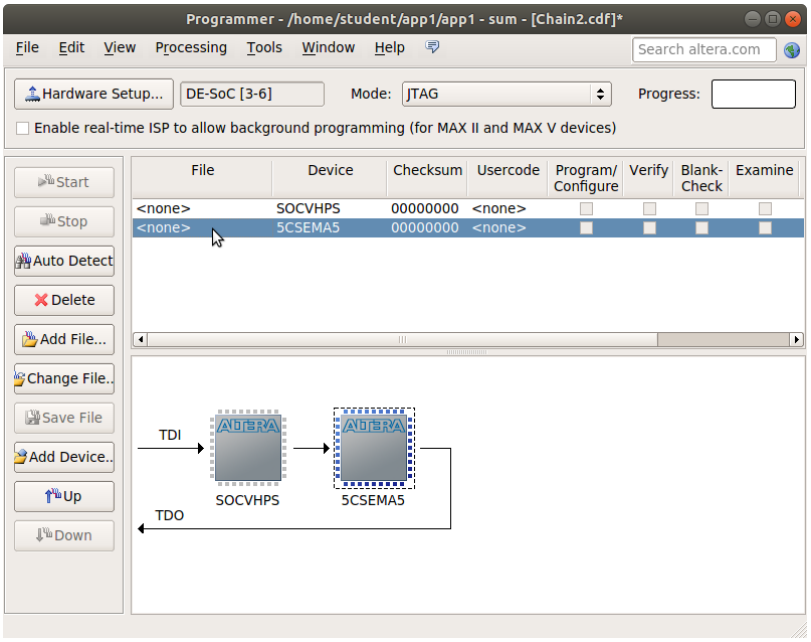


Figure 28: Programmer window with detected chain

You will always program the FPGA, therefore double click on the 5CSEMA5 row (second row) under the *File* column as indicated in Figure 28. A file selection dialog box (Figure 29) allows you to select the FPGA programming bit file:

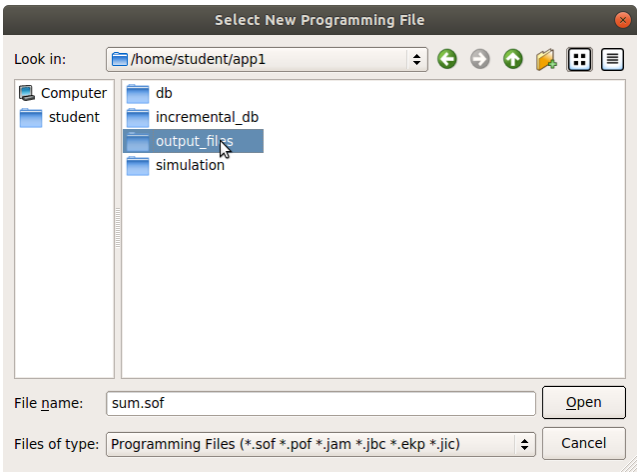


Figure 29: Select programming file window

In the project folder (in which the explorer opens by default) select the `output_files` subfolder. There you will find the `.sof` file of the project. It has, by default, the name of the top-level module.

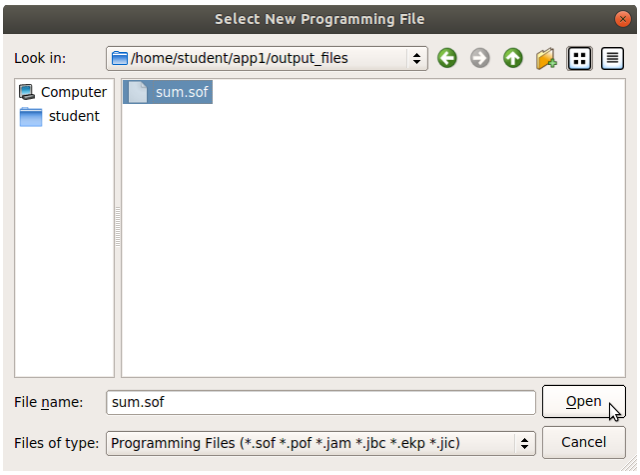


Figure 30: Select programming file window

Select the desired `.sof` file and click **Open** (Figure 30).

The file selection dialog box closes and the Programmer window (see Figure 31) has now the row of the FPGA device filled with all needed details (don't bother, if you followed all steps as in this tutorial, they are all OK).

One last thing you need to do before programming the FPGA. Click the **Program/Configure** checkbox in the device row (5CSEMA5) as shown below (Figure 31).

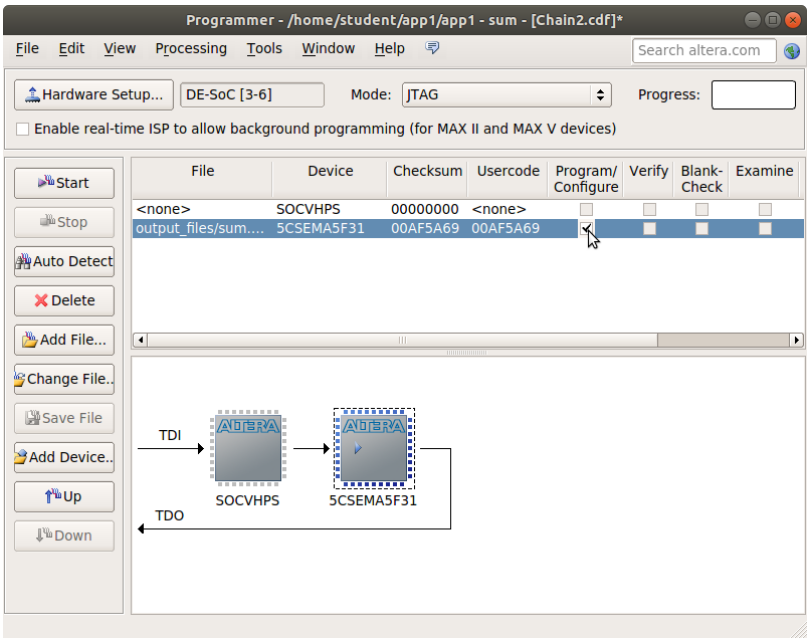
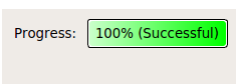


Figure 31: Programmer window ready to program the FPGA

Now cross your fingers and click the **Start** button in the Programmer window. A couple of seconds are needed to transfer the .sof file to the FPGA on De-Soc board. If the Progress field on the top right of the Programmer window stops at 100%, the board is programmed and ready to play with. **Congratulations!**



Quick Project Guide

- **New Project Wizard**
  - Step 1 [Directory, Name, Top-Level Entity]
    - **working directory**: *myProject*
    - **name of this project**: *myProject*
    - **name of the top-level design entity**: *myModule*
  - Step 3 [Family & Device Settings]
    - Select device (5CSEMA5F31C6)
  - Step 4 [EDA Tool Settings]
    - **Simulation** -> **ModelSim Altera** -> **Verilog HDL**
- Design source files
  - **File** -> **New …** -> **Verilog HDL File**
  - Save as *myModule.v* (if it's the file for the top-level module) or *mySubmodule.v* (for other source files)
  - Edit & Save
- **RTL Simulation** Flow -> **Analysis & Elaboration** (fix errors and rerun if needed)
- Testbench source files
  - **File** -> **New …** -> **Verilog HDL File**
  - Save as *myModule_tb.v*
  - Edit & Save
- **Settings** -> **Simulation Settings**
  - check **Compile test bench**
  - **Test Benches…** -> **New…** -> New Test Bench Settings
    - select *myModule_tb.v* file and **Add** it
    - type *myModule_tb* in the **Test bench name** textfield
- **Tools** -> **Run Simulation Tool** -> **RTL Simulation**
  - Analyse, debug, edit, recompile, restart, run
- **Assignments** -> **Pin Planner**
  - choose **Location** for each pin
- **Compilation** Flow -> **Compile Design**
- **Compilation** Flow -> **Program device**
  - Power on the physical board (its LEDs start flashing)
  - **Programmer** -> **Hardware Setup…** -> select De-Soc device
  - **Auto Detect** -> select 5CSEMA5
  - Select *myModule.sof* file for 5CSEMA5 device and check **Program/Configure** checkbox
  - **Start**