Projects for Veek-MT2 board in Intel Quartus Lite V20.1

Version 1.2 from 16-Oct-23, published under GNU Free Documentation License

 Department of Control Eng.,

Faculty of Electrical Engineering

Czech Technical University in Prague

Content

1. About names and filenames

Beware, Quartus is a Unix program that runs in Cygwin, an extensive collection of GNU and Open Source tools that provide functionality similar to a Linux distribution on Windows. Thus, all folders and files should satisfy strict Unix name rules.

The filenames can contain upper and lower case ASCII letters, numbers, dots "." and underscore "_". Do not use spaces and special characters. For completeness, some Linux versions allow them, but we have bad experiences with their usage in Cygwin. You can try to see if they work for you, but if not, complain to yourself.

We recommend using only ASCII letters and numbers in Quartus entity names. They usually correspond with filenames because one entity is stored in the file of the same filename. Its identifier must start with a letter, the last character cannot be an underscore, and two connected underscores are not allowed. However, the entity names with underscores can sometimes collapse the internal Quartus simulator run if we try it. We do not find out why. It just sometimes happens.

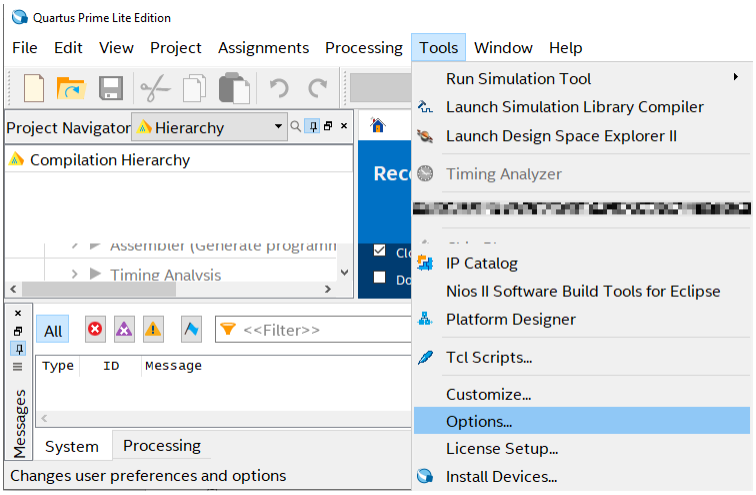
VHDL is not case-sensitive, but we should keep capitalization. If we define the symbol Data, the wrong style is referring to it as "data" or "DATA".

VHDL does not understand diacritics and non-ASCII characters, not even in comments.

2. Initial Configuration

Before creating the first project, you should perform permanent global configurations that will be reflected in all the following projects.

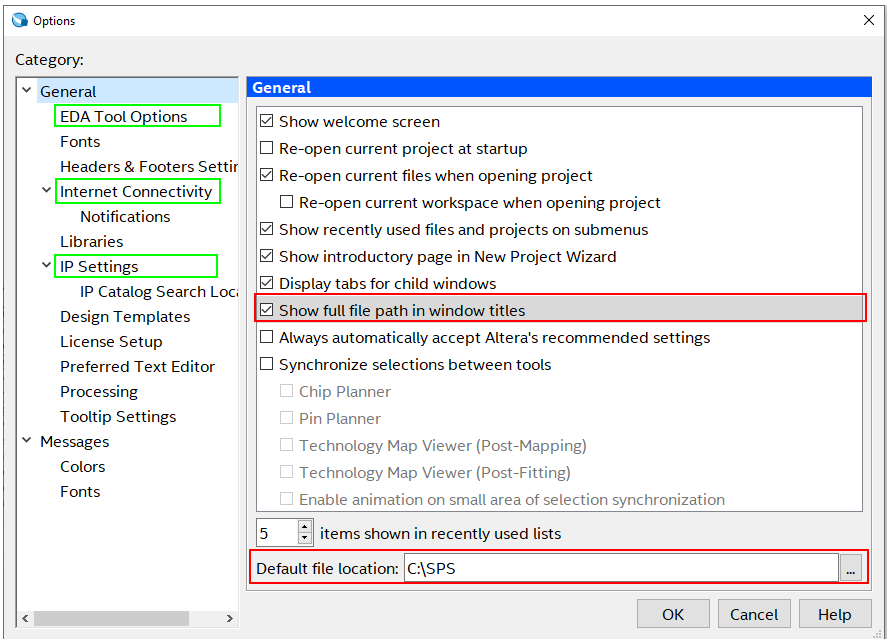
Select **Option** from Quartus menu Tools:



We start in the General tab, where we specify the default directory in which we plan to store our projects. Its choice is ours.

We recommend also checking:

"Show full file path in window titles" to improve orientation.



Then, we proceed to EDA Tool Options and set the path to ModelSim-Altera to ensure it will always be located.

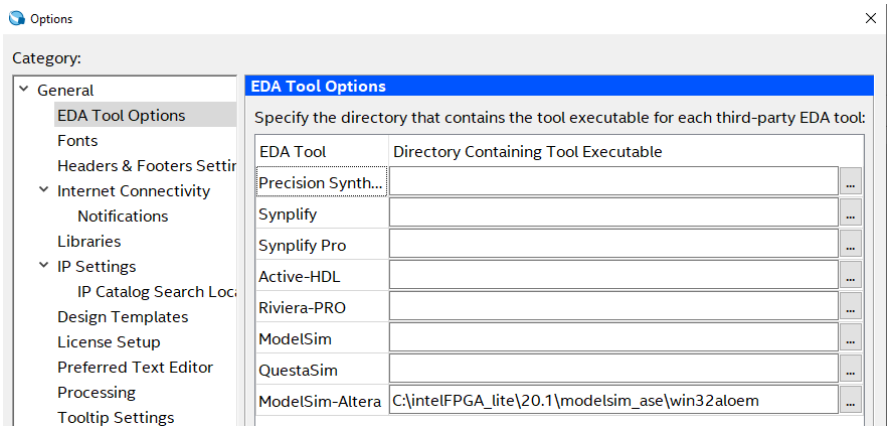
We find it in subfolder

modelsim_ase\win32aloem

of the install directory. The default is:

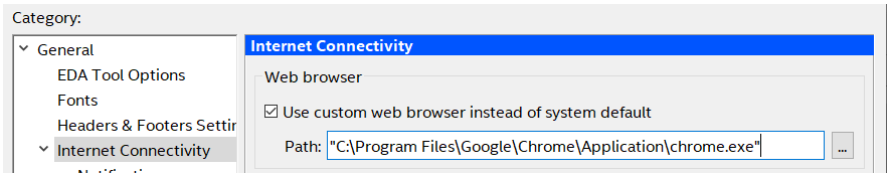
C:\intelFPGA_lite\20.1\

.



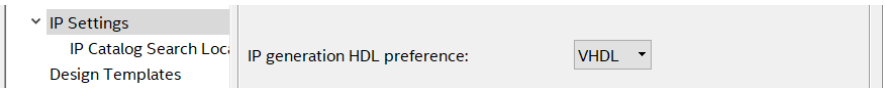
Note: EDA abbreviation stands here for Electronic Design Automation tools that chip designers use to analyze entire semiconductor chips. The Quartus installation contains only one free tool, ModelSim Altera, i.e., ModelSim with embedded Quartus FPGA libraries.

We can also set our default browser. Quartus uses it to display its helps.



In IP Setting category, select VHDL.

Note: IP is Intelligent Property here.



If you wish, you can adjust the primary and message fonts and, eventually, other options.

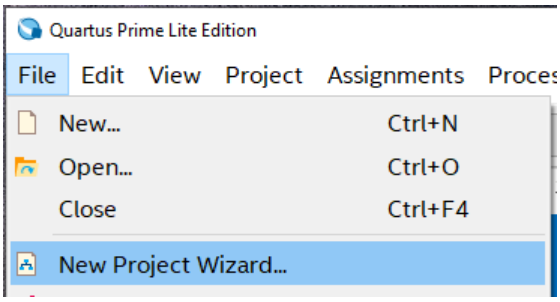
Finally, close the Option dialog by [OK] button to store changes; otherwise, they will be canceled.

Then, close Quartus Prime and rerun it so Quartus starts with new options.

3. Project Wizard of the First Project

Creating our first project is complicated by many settings, but it is one one-time task; the following projects are made in a minute; see Chapter 4 on page 12.

From the main menu Quartus, we choose File-> New Project Wizard.

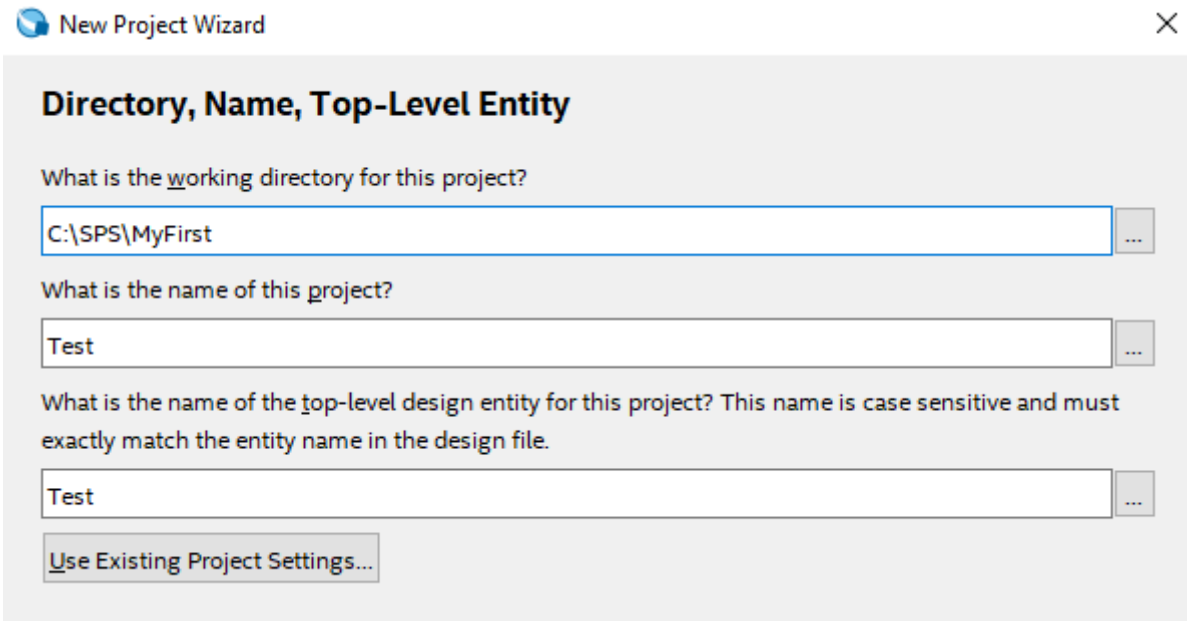


The Wizard's dialog consists of several pages; you can scroll from one page to another, forward by [Next], and backward by [Back].

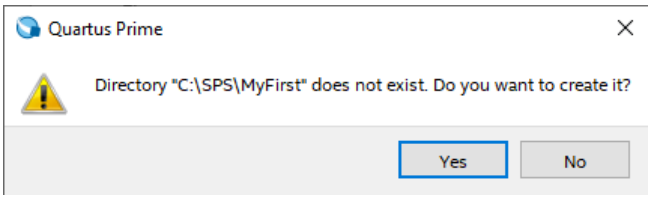
►Introduction — continue by [Next >]

►Directory, Name, Top-Level Entity

Enter any folder name inside an existing directory, e.g., "MyFirst" ─ the folder should not exist yet; it will be created. We also choose names for our project and entity, e.g., "Test". Do not use anything that could accidentally be an HDL language keyword, such as "else", "input", "output", and so on.



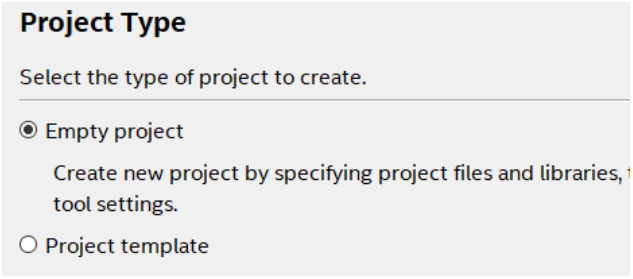
After entering all items, click the [Next >] button and confirm the creation of a new directory. If you see this question, it is a good sign; Quartus projects cannot share directories.



Note 1: *Quartus does not allow the storage of two projects in one folder! Each project has its directory.*

Note 2: *Unlike older Quartus versions, the button [Use Existing Project Settings] does not allow copying many settings in Quartus 20.1. We show the better way on page 12.*

►**Project Types** —We select Empty and press [Next >] because project templates do not exist for our educational board Veek-MT2.

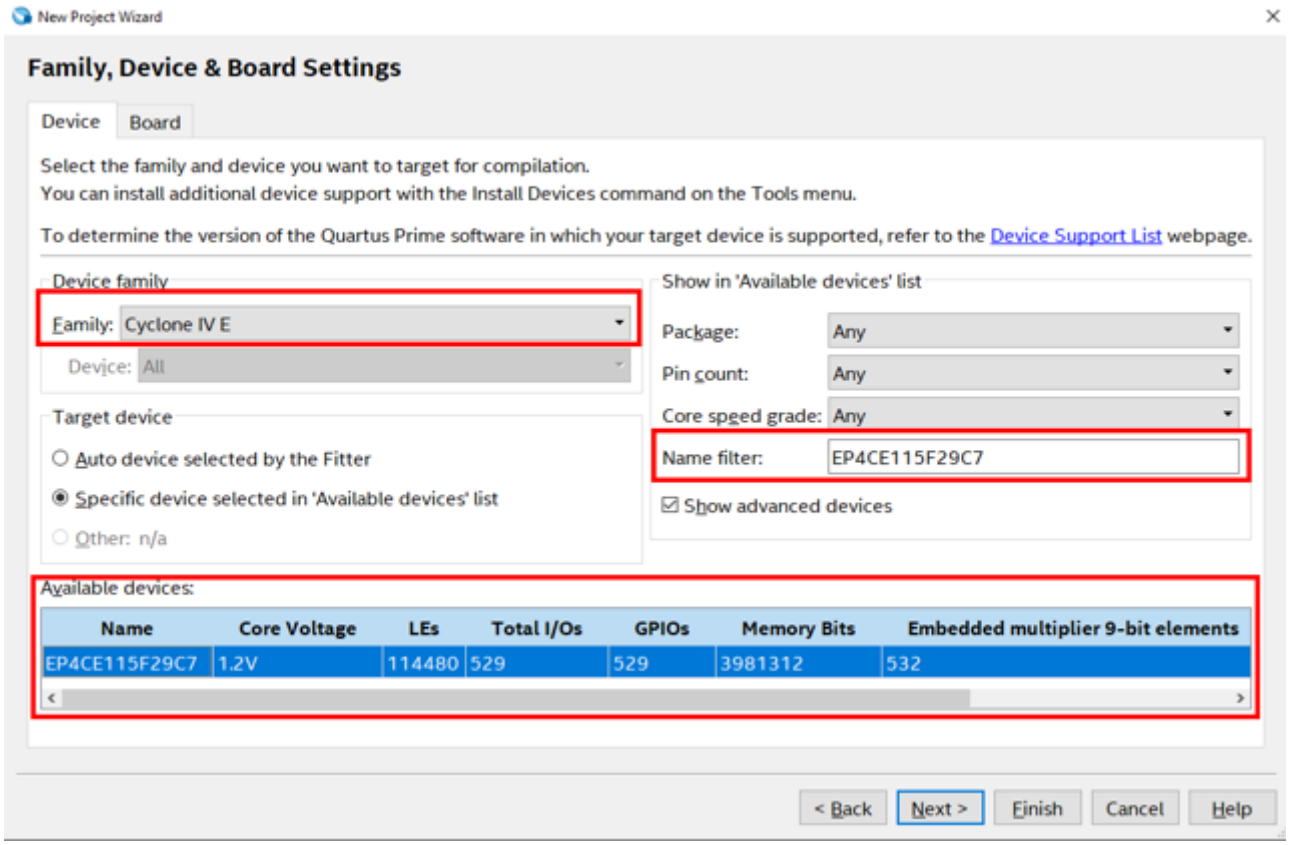


► **Add Files** - skip by [Next >]

►**Family Device & Board Settings** —enter the following for development board Veek-MT2:

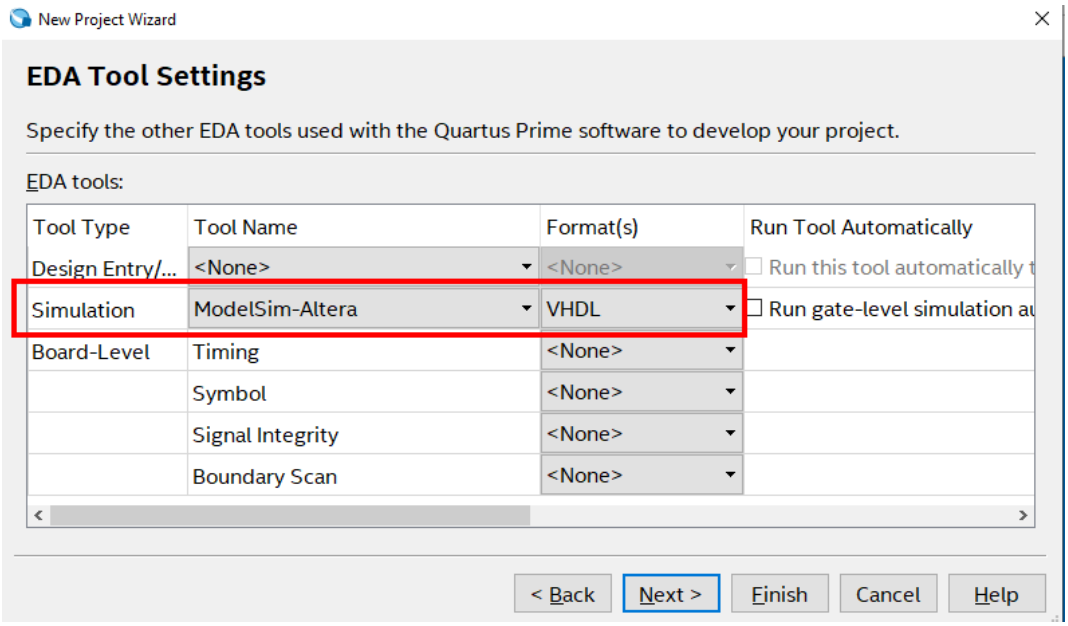
- First, we must set the family: **Cyclone IV E**
- Then, we write/copy in the name filter: **EP4CE115F29C7**

In the list of available devices, we select FPGA EP4CE115F29C7 by mouse double click.



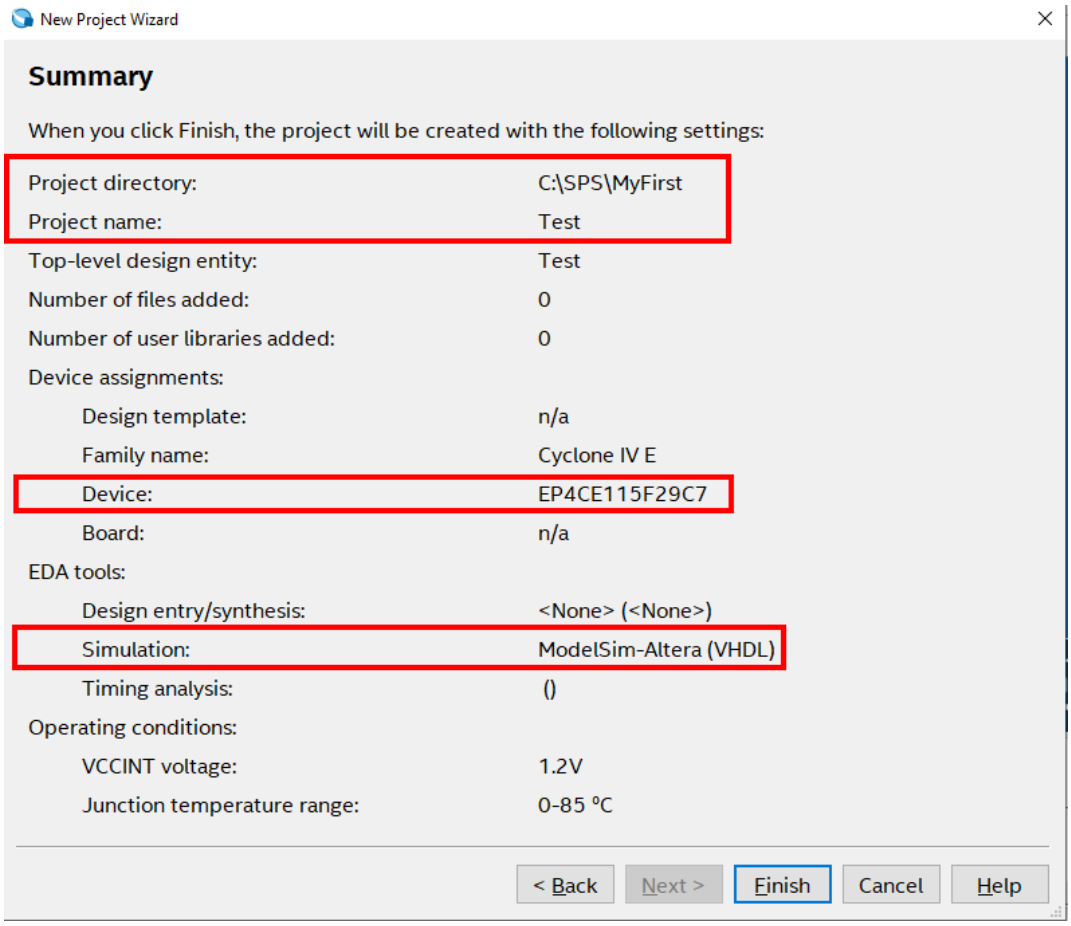
Click [Next>]

►**EDA Tools Settings –** Set Model-Sim-Altera in VHDL and leave the other items in default states.



Click [Next >]

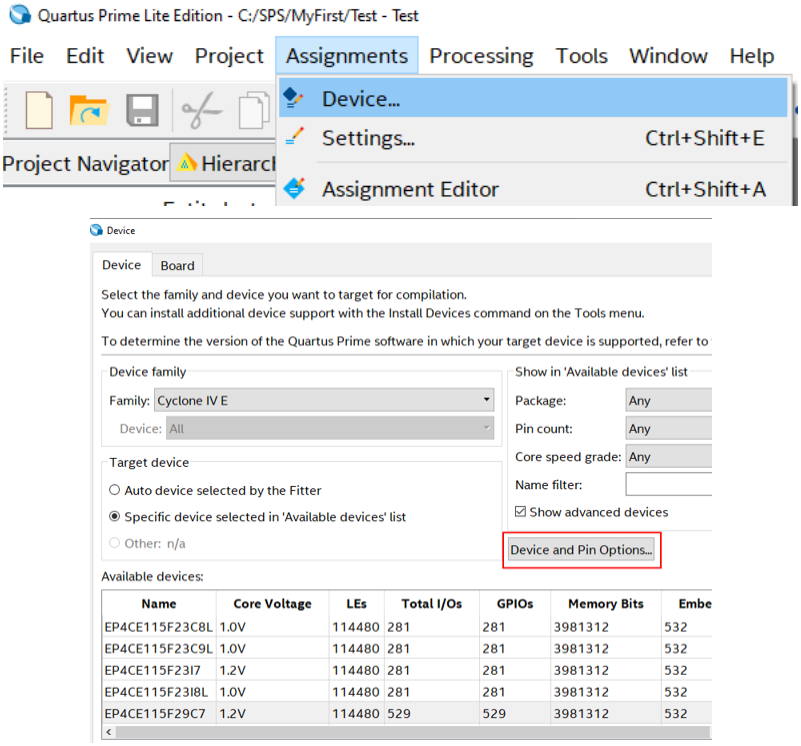
►**Summary** – check if everything is OK.



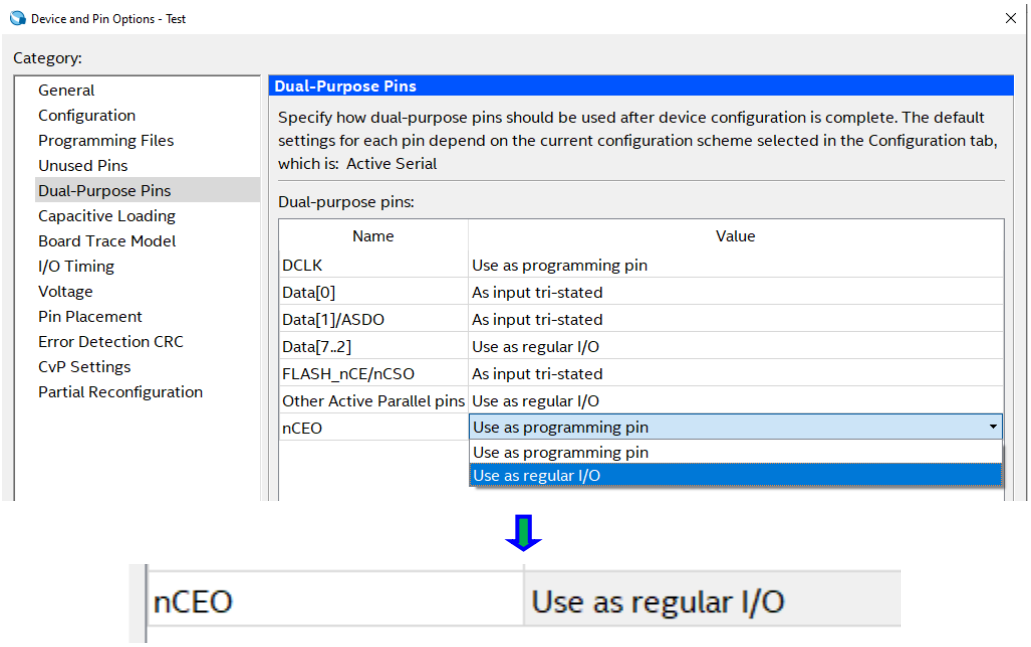
If it is OK, click [Finish]

3.1 Additional Project Setting for Board Veek-MT2

Open the Device dialog from the Quartus menu Assignments->Device:



We saw this dialog in Project Wizard, but now, the new button [Device and Pin Options] appears here. Click on it and select in the Dual-Purpose Pins tab that nCEO pin of FPGA will be used as regular I/O.



By [OK] buttons, we first close the sub dialog "Device and Pin Options" and then the "Device" dialog.

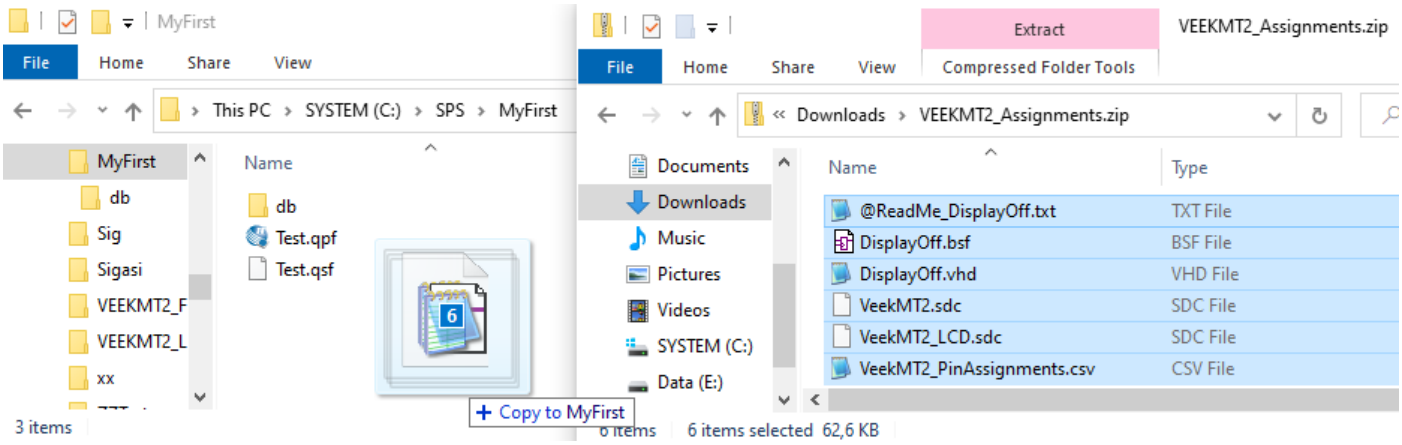
Note: Pin nCEO (Chip Enable Output) serves in JTAG communication as the selection of an FPGA device, to which we download configuration. VEEK-MT2 board contains only one FPGA chip, which is always selected. Its printed board utilizes nCEO as an LCD input; therefore, we must inform Quartus about this modification.

3.2 TimeQuest Setting

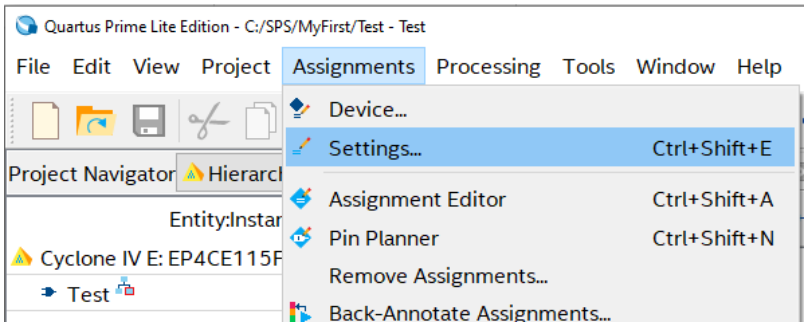
Download zip-file from our web site:

https://dcenet.fel.cvut.cz/edu/fpga/veek-mt2/VEEKMT2_Assignments.zip

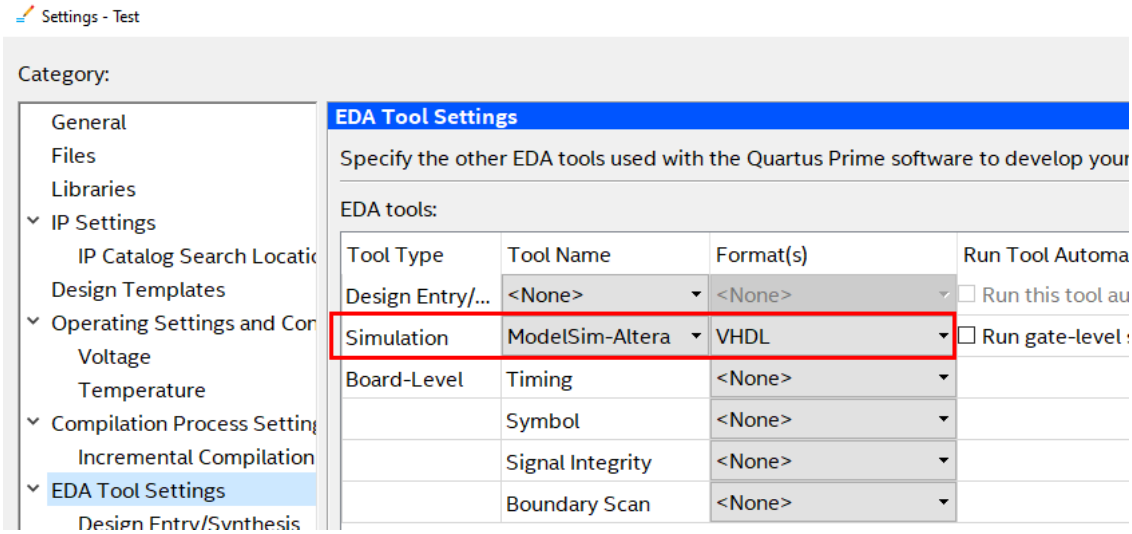
and unpack it into the directory of the created project.



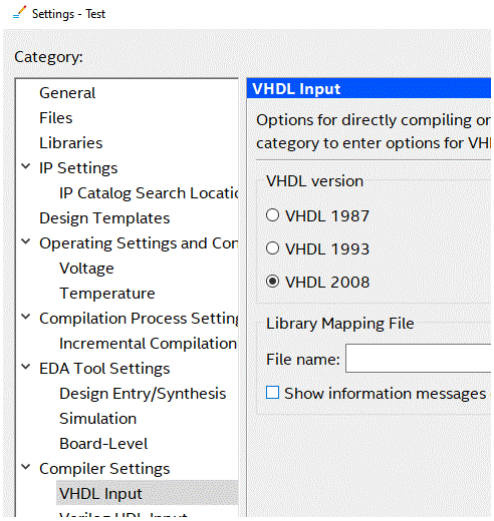
Invoke the Settings dialog from the Quartus menu:



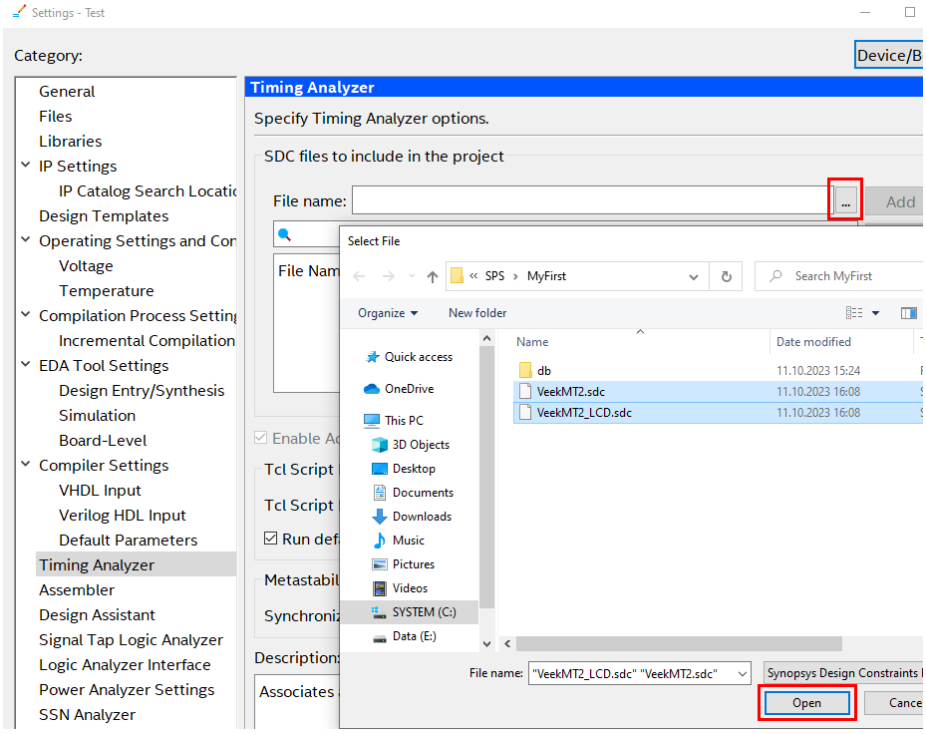
In the tree Category of the dialog Settings, select EDA Tool Settings. Set up the simulation tool to ModelSim Altera in VHDL. Do not change anything else here! The free version does not contain other EDA programs.



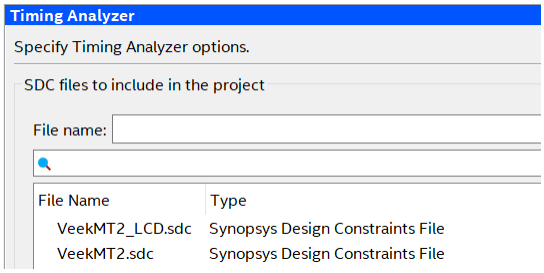
In the category Compiler setting -> VHDL input, check VHDL 2008.



In Timing Analyzer category, we browse for files *.sdc (Synopsys Design Constraints) that were previously extracted from the zip archive.



We select them both. They add to the list of Timing Analyzer constraints.

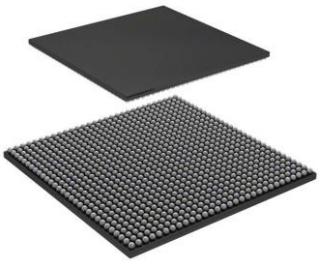


*Note: The *.sdc files tell Quartus that the educational board VEEK-MT2 contains a clock generator with a frequency of 50 MHz. The clock frequency is a critical specification. Without this information, our circuit will be compiled for a default frequency of 1 GHz. And the internal FPGA connections depend on frequency, as we explain in our lectures, so that the result will be nonfunctional.*

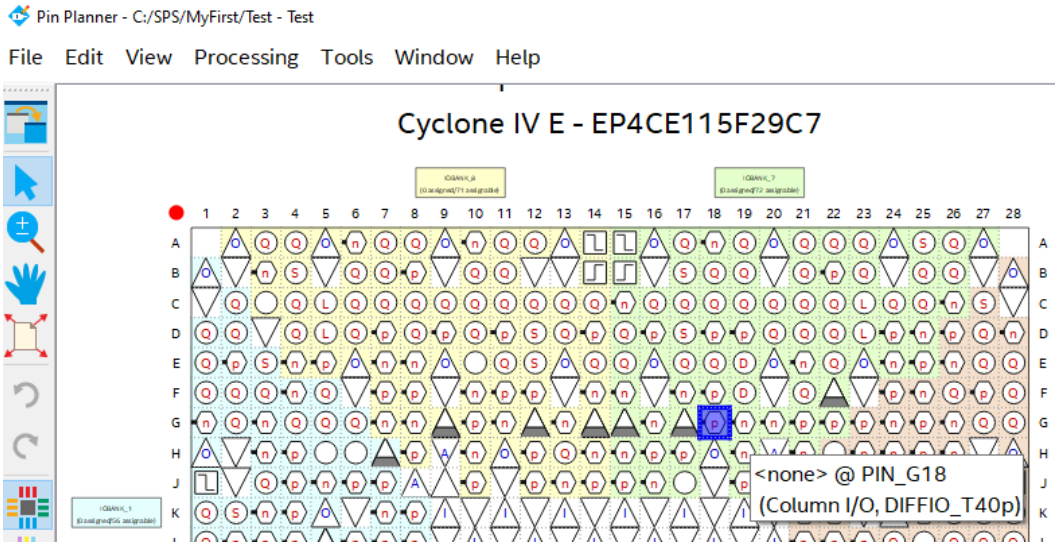
Finally, we close the Setting dialog by the **[OK] button** to accept data.

3.3 Pin Assignments

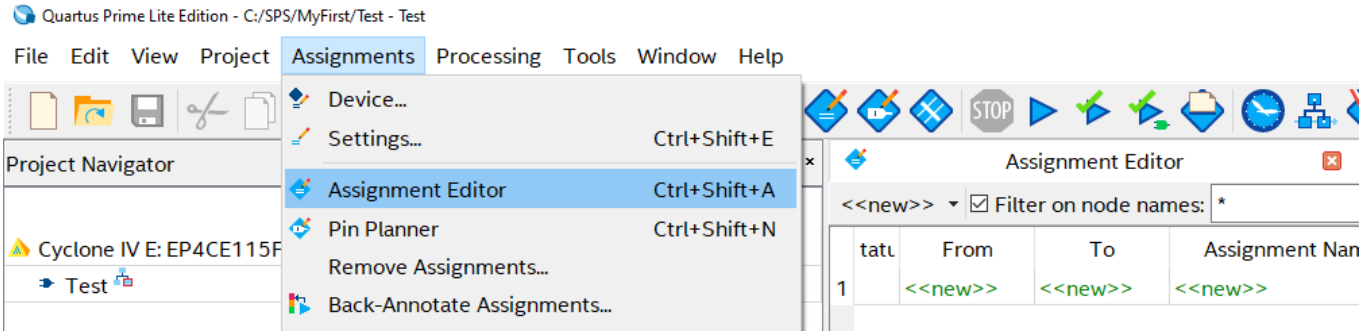
FPGA chip used in VEEK-MT2 has 780 solder balls arranged in a grid or array at the bottom of the package body for external electrical connections; see the left picture [source Intel]. Each solder ball is internally one pin.



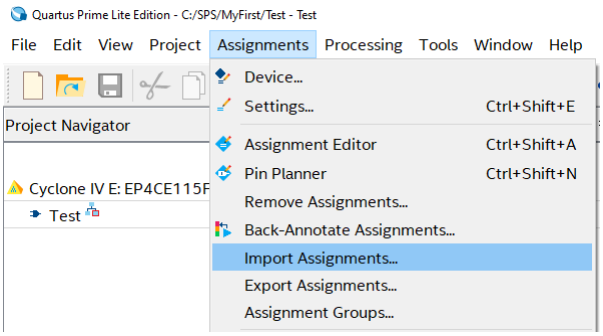
Users can direct inputs or outputs of their circuit designs up to 528 pins. This possibility can simplify printed board designs. The pins are indexed with a letter and a number. We can see them in Pin Planner.



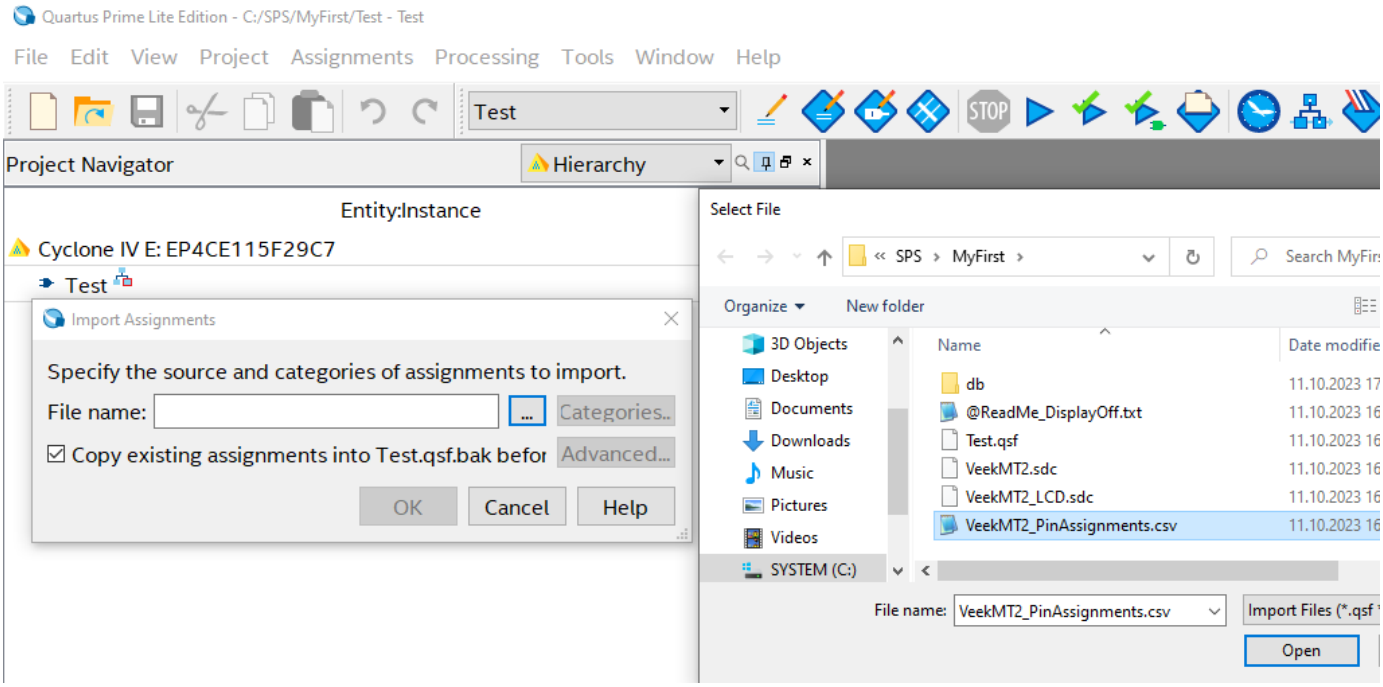
We work with a ready-made board; therefore, we must follow their placement by loading the manufacturer's pin assignments. Their absence is a frequent error in beginner's designs. Always check if the Assignment Editor shows pins and not an empty list as below:



The original pin indexes are complex to remember, so we import their association with their symbolic names defined by the Pin Assignment spreadsheet table. In the main Quartus menu, select: **Assignments->Import Assignments.**

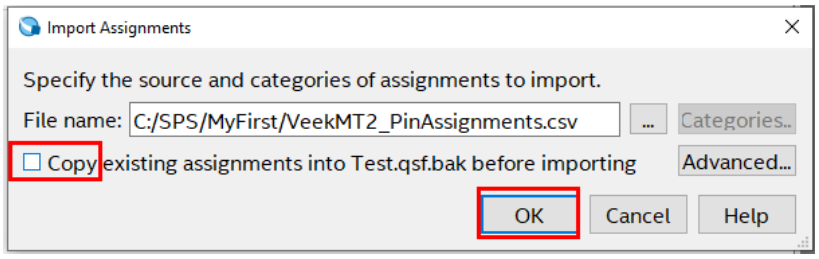


In the Import Assignment dialog, we browse for the download file VeekMT2_PinAssignments.csv.

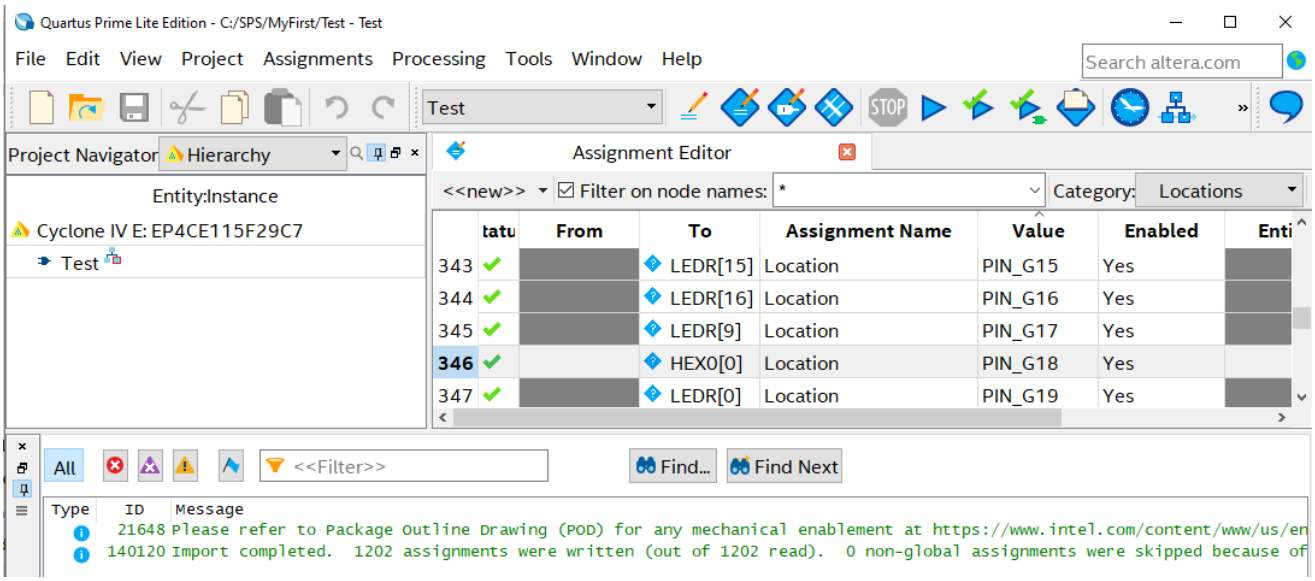


We confirm the choice by [Open] and then [OK] buttons.

We have no assignment yet, so we unchecked the copy option. We import by [OK].



If we look at **Assignments-> Assignment Editor**, the list now contains associations. We can find out that PIN_G18 from the previous pin planner figure was named HEX0[0], i.e., it is the upper led segment in the least significant 7-segment digit :-)



<u>It was the last setting. We have our first project.</u>

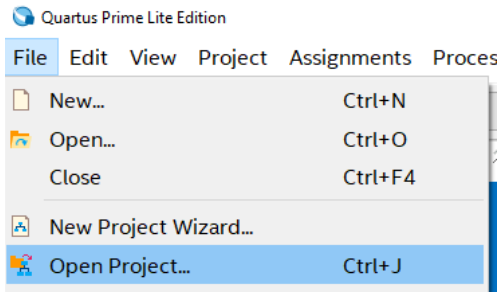
~ o ~

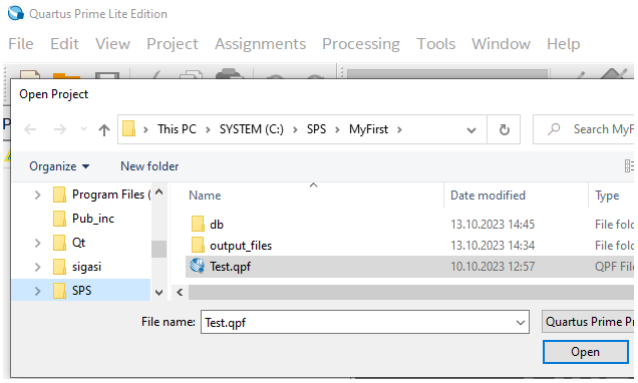
Never edit the pin assignment list; keep compatibility by using predefined names!

4. The Express Copy of a Created Project

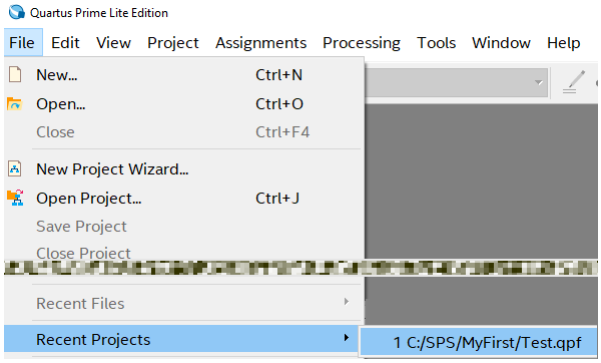
If I have already created at least one complete project from scratch, we can copy it to obtain another project in a minute.

From the Quartus menu, we open any of our projects, e.g., we use MyFirst:

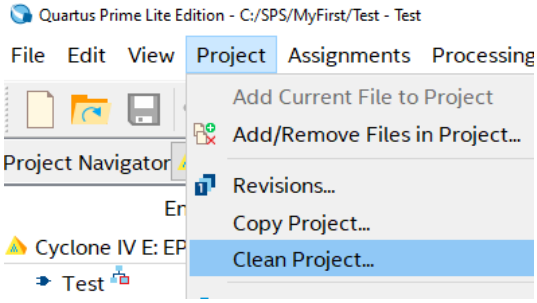


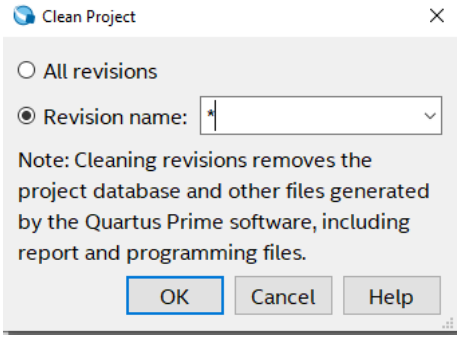


Eventually, we can access it from the "Recent Project" offer:

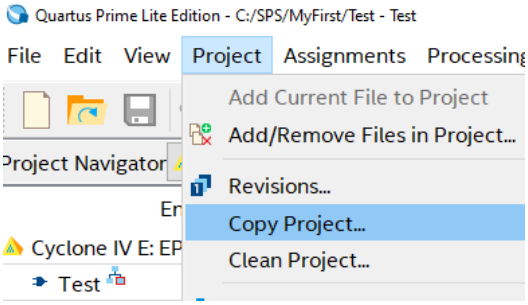


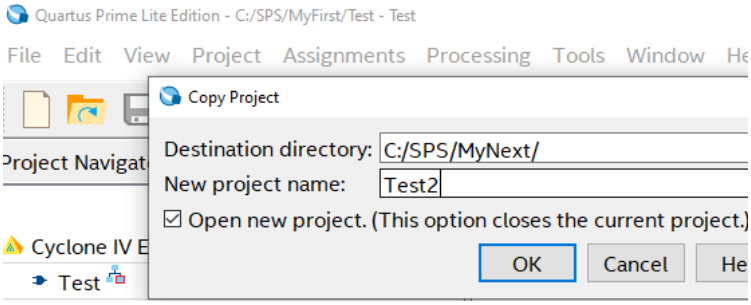
First, we select "Project-> Clean project" to remove all temporary files, and by [OK], we confirm either the default option or All revisions.



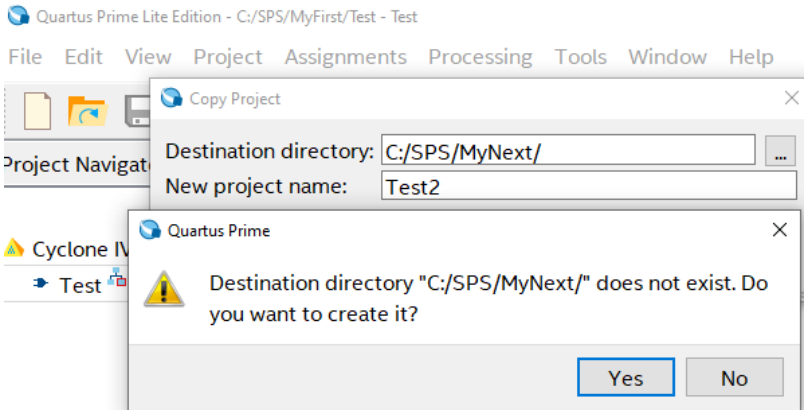


Then, we open the "Project->Copy Project" dialog. We enter our project name, e.g., Test2, and specify a directory where we want to store its files.





After confirming our choice by [OK], the message below signals that we are creating our project in the new directory, which will be its own, as required.



We have already created our new project with all the necessary settings. It also contains all the Test project files.

The following chapter explains how

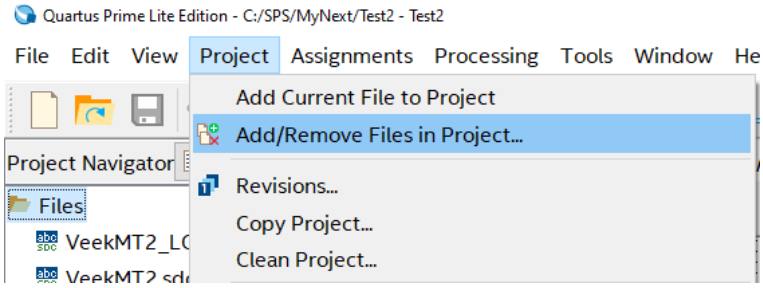
- to change Top-Level Entity;
- to adjust our project by adding new files or removing unused ones;
- to move our project to another directory or computer;
- to change the compilation result file name.

5. Working with Projects

The following steps are more adjustments than requirements.

5.1 Reorganizing files in Project Navigator

From the file dialog, we can reorganize files in our project.



It is possible to add new files or remove ones no longer needed. The changes are performed only in the list. Files are not deleted from the hard drive; any file explorer can do it.

We always keep all our design files in the directory of the project. We easily recognize wrongly placed items by their extra path.

In the left figure, wrongly located files are marked by red box.
The first one

../MyFirst/DisplayOff.vhd

is in the parent's subfolder:

C:/SPS/MyFirst/DisplayOff.vhd

i.e., outside our project.

The second file

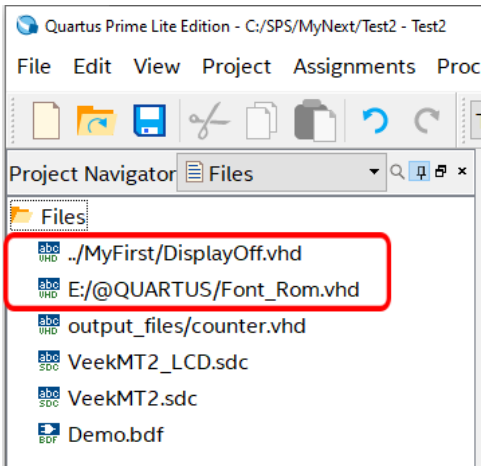
E:/@QUARTUS/Font_Rom.vhd

is stored even on a different drive.

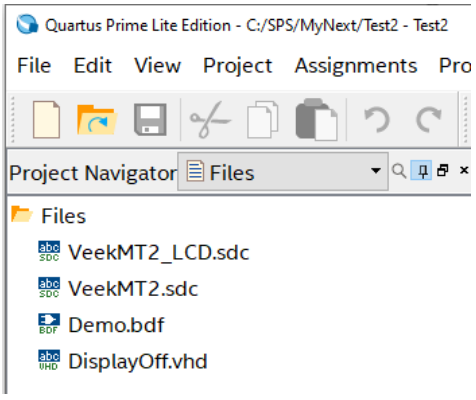
The third file

counter.vhd

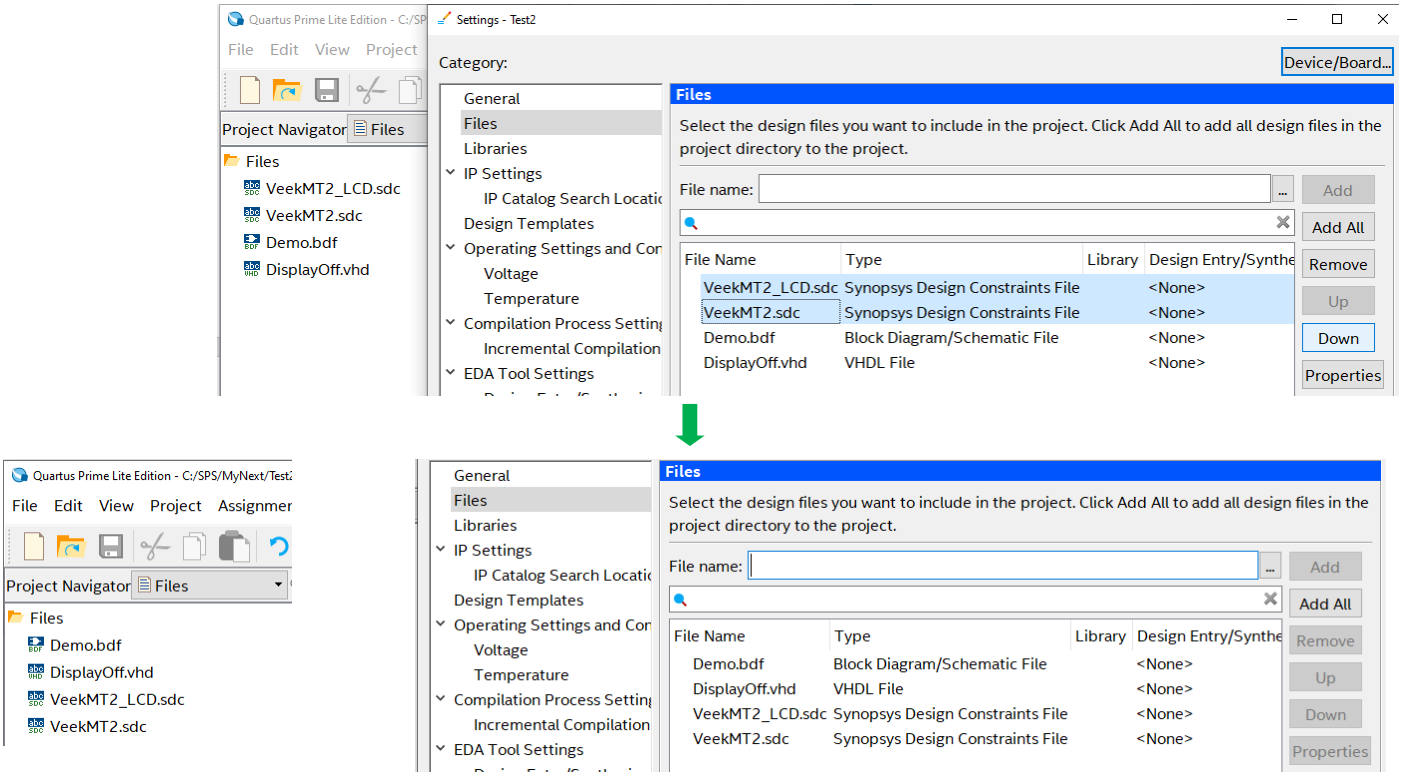
is in a subfolder output_files of our project. It is acceptable but not recommended for small projects.



The preferable file locations are in the project directory, i.e., without paths.



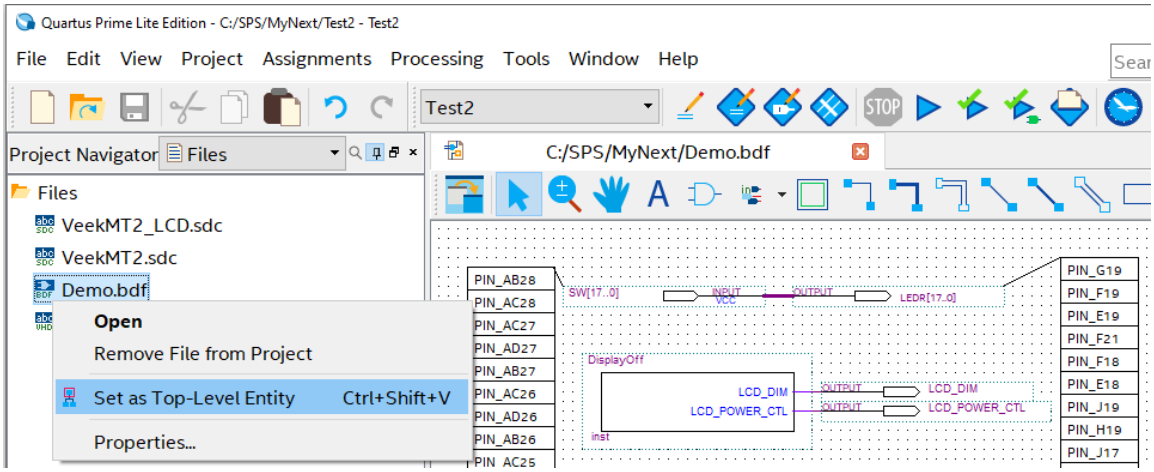
Eventually, we can rearrange the Project Navigator list with buttons [Up] or [Down]. Then, we confirm the changes by clicking [OK] button.

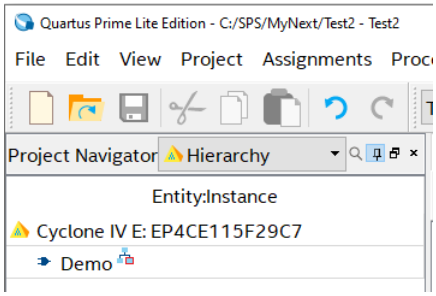


5.2 How to specify Top-Level Entity

Our Top-Level Entity selects the design file from which the compilation begins. After we write down some synthesis files, we can dedicate anyone as the Top-Level Entity from the Project Navigator Files.

For instance, we add DisplayOff.vhd, extracted from the zip mentioned in Chapter 3.2 on page 8, and create a simple Demo.bdf that copies slider switches to red LEDs. Then, we mark it as a Top-Level entity from its context menu (a right mouse click on file).





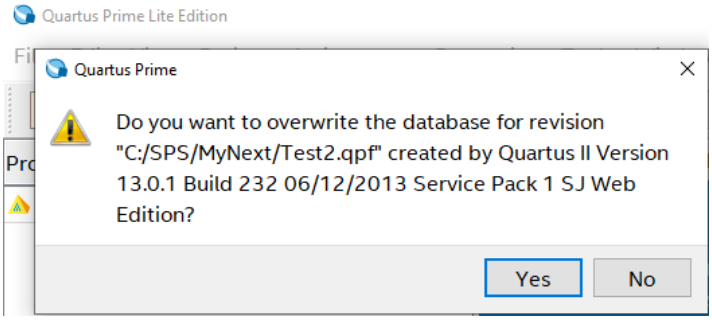
If we switch Project Navigator to the Hierarchy tab, we see that Demo has become our Top-Level Entity.

5.3 Moving and zipping our project

Quartus is very versatile. We can open a project created in version 20.1 in some of its previous versions, e.g., in Quartus 13.1, and any lower version project in upper Quartus.

If we have all files in the project directory, as was explained in the previous chapter, Quartus keeps their path relative. So we can rename the project's root directory or move it to another location.

When we move our project to a different computer, directory, or Quartus version, we can see a message similar to the picture below:

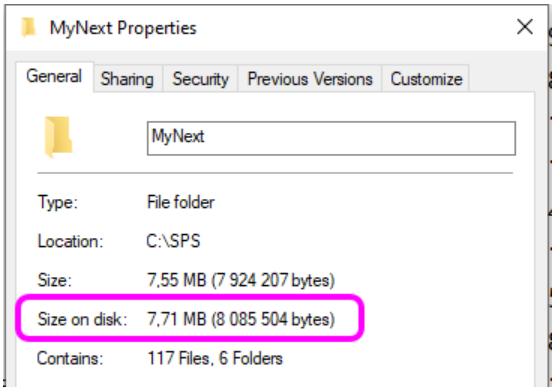


We should always confirm [YES] if we have a small project.

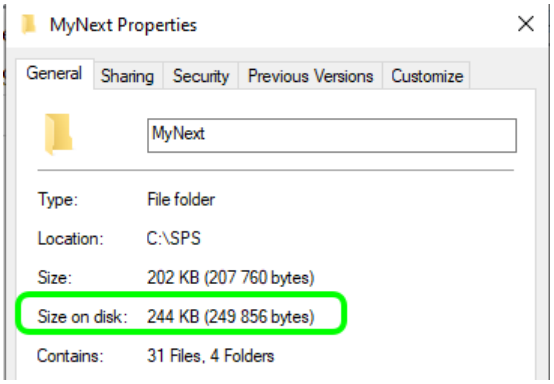
Note: Compiling huge industrial designs can sometimes last for several hours. Therefore, the Quartus asks if we allow deleting previous results. It is better to let the Quartus begin with clean intermediate databases in small projects.

However, our tiny project can surprisingly contain many temporary files created during compilation and placement into an FPGA chip; see the figure below.

The project with Quartus compiler temporary files

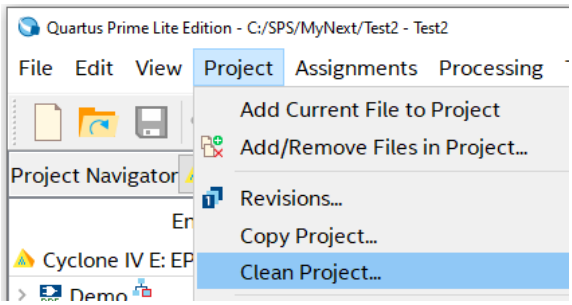


without the Quartus compiler temporary file

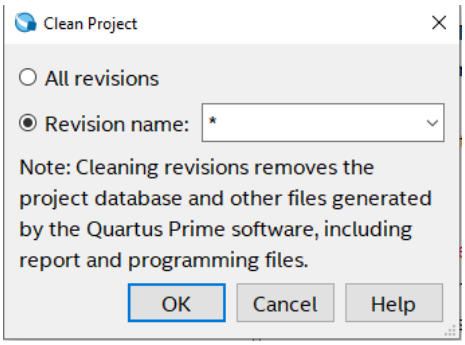


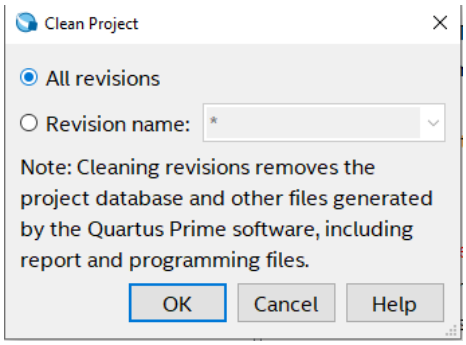
Even if our design had a few files, the FPGA is big. The compiler needs to configure everything in its whole chip, i.e., in our case, it mostly deactivates non-used parts.

We can reduce the project size by removing temporary files before zipping or copying the project to a USB stick by the Project -> Clean Project, equivalent to the "make clean" command in classic programming.



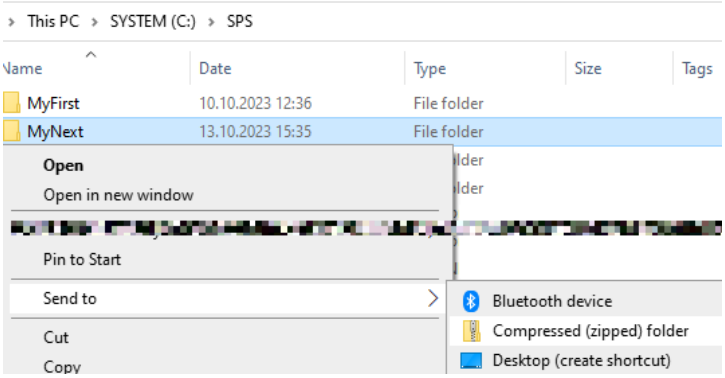
Any of the selections below leads to removing all Quartus compiler temporary files.



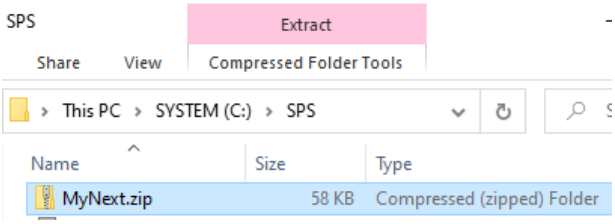


Then, we close the Quartus to release its locked files and run the File Explorer.

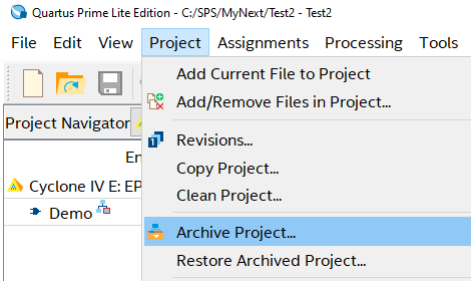
We select the Send to->Compressed (zipped) folder from the context menu of the root directory folder.



The result of the compression:



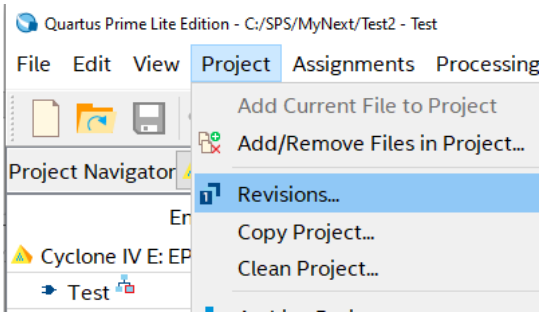
Note: We prefer zip to the Quartus menu command the "Archive Project":

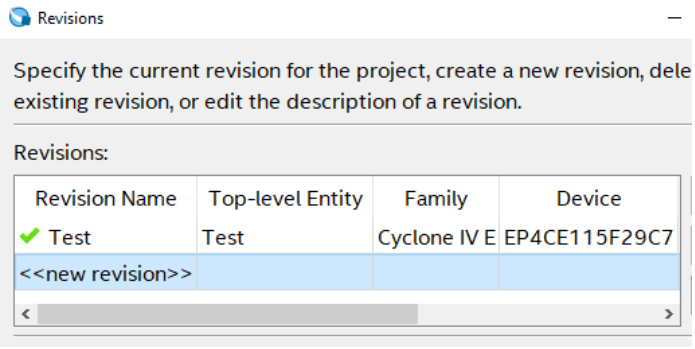


*Its result has an internal format *.qar that can be depacked only by Quartus. The zip file is more versatile.*

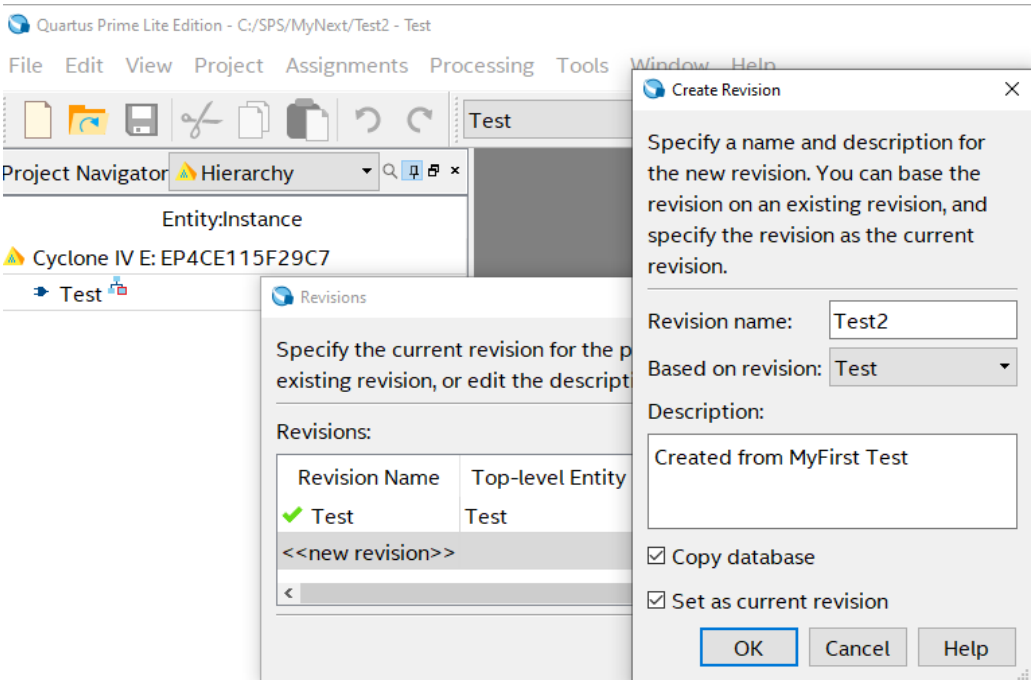
5.4 Changing Revision = Compilation Result Filename

The compilation generates the result, which filename is derived from the revision name. For example, the result can be Test.sof (sof=SRAM Object File). The Revision is also copied with the project. We can quickly change it from the Revision dialog:



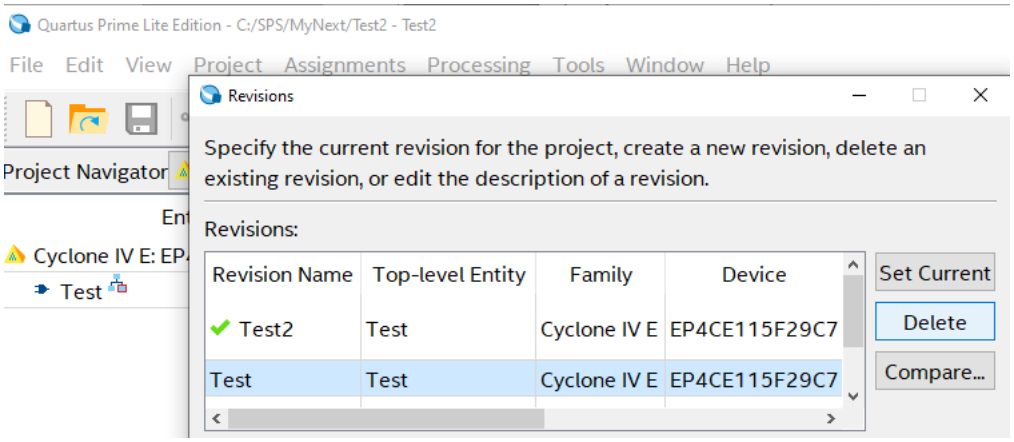


By mouse double click into <<new revision>> table row, we open sub dialog:

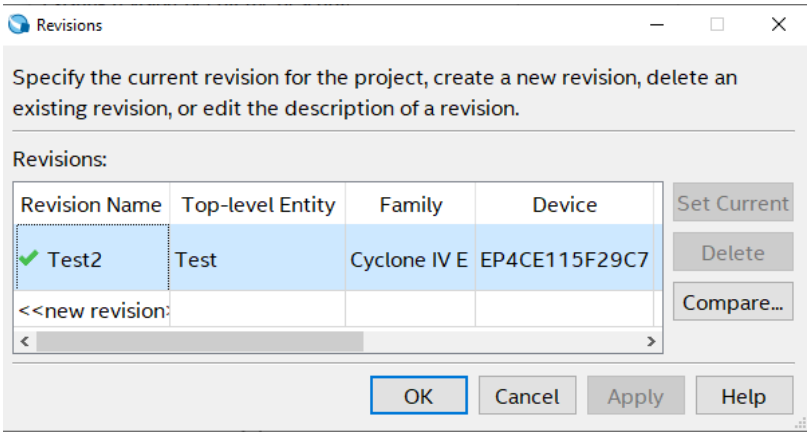


We enter, for example, Test2 as the name of our new project and possibly some description if we wish.

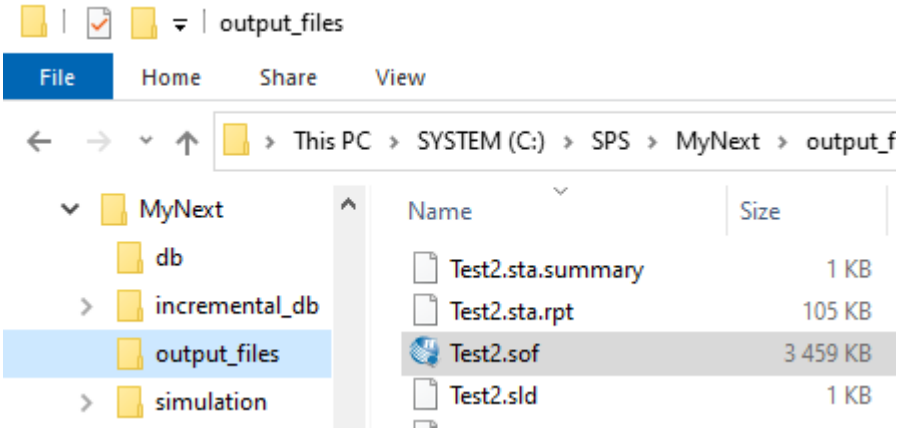
After confirming by [OK], we again open the menu: Project ->Revisions.



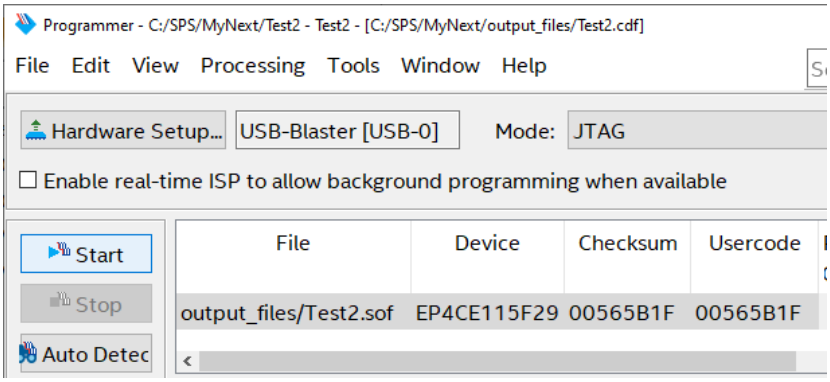
After deleting the old revision Test, we obtained the project, which compilation leads to Test2.sof.



The result Test2.sof can be downloaded into the FPGA chip by the Quartus programmer.



The Quartus menu: Tools->Programmer shows the dialog below.



We explain its setting in practical exercises of our LSP course. Its brief explanation is on page 40 of the document:

https://dcenet.fel.cvut.cz/edu/fpga/de2-115/My_First_Fpga.pdf

Note: VEEK-MT2 is a combo of LCD and DE2-115 board; thus, DE2-115 manuals are valid, but DE2-115 Pin Assignments define only a part of VEEK-MT2 pin assignments.

The programmer more advanced functions, which we do not use in the LSP course, are described in the Intel document:

https://www.intel.com/content/www/us/en/docs/programmable/683039/20-4/introduction.html

~ o ~

*** The End ***