Table of Contents:

Intentionally left blank

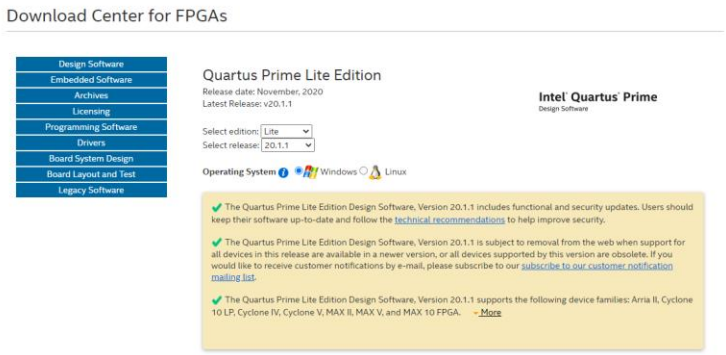
Installing Quartus & ModelSim: (V20.1.1)

Quartus is a development environment for implementing digital circuits within Intel FPGAs. (originally Altera)

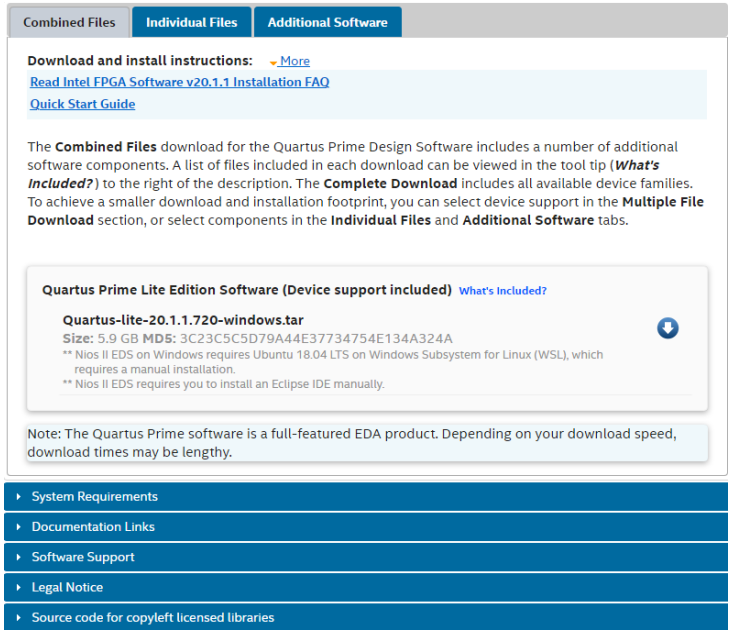
ModelSim is a discrete Event Simulator made by Mentor Graphics. A special version of ModelSim is bundled with the Quartus download. This special version contains timing libraries for Intel FPGAs.

Download and Install the Software:

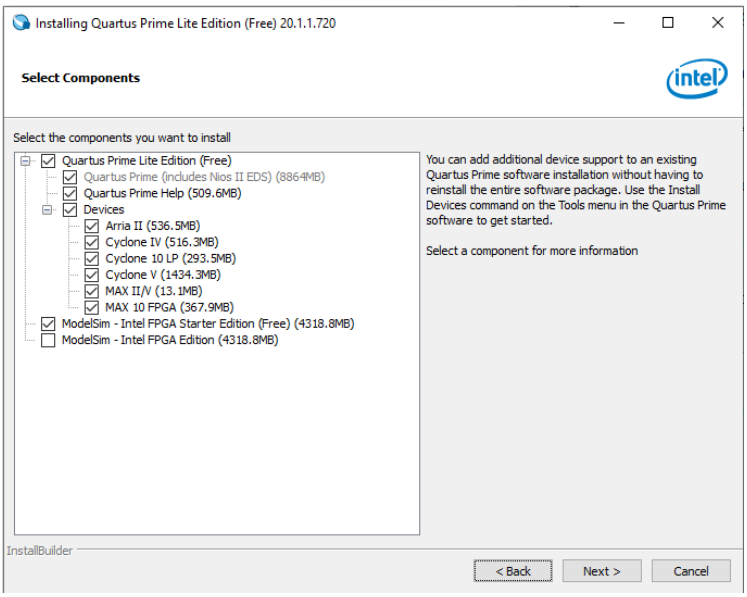
1) Determine whether a previous version of Quartus and/or ModelSim is already installed.
2) Un-install if necessary.
3) Find the download page for the latest version of Quartus-Prime-Lite. (free version). It is easy to find the download link. Start by searching with the keywords “Intel”, “Quartus” and “download”.



4) Choose the “Combined Files” version that contains ALL the stuff.
5) You will be asked to sign-in to your Intel Account – register if you don’t have an account. The download took approximately 20 minutes on my computer. (5.87 GB)

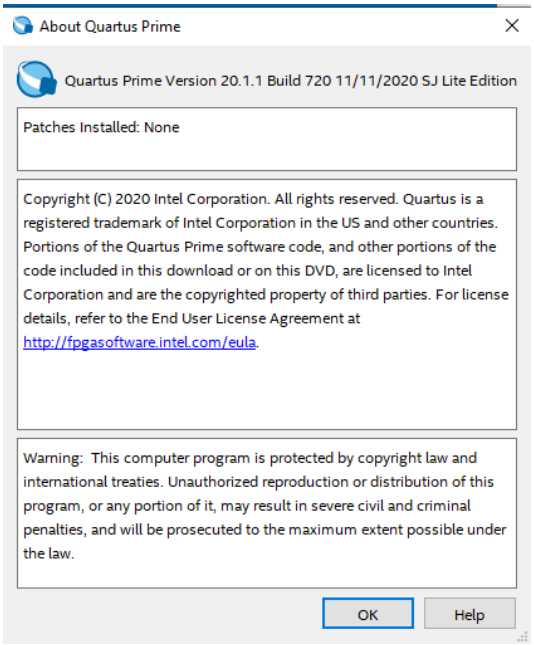


6) Run the installer. Installation takes approximately, 20 minutes



7) Shortcuts to the software may occur using the name “Intel”.

- Run ModelSim and using the menu help->about ModelSim verify the version and licence.
- Run Quartus and using the menu help->about Quartus Prime verify the version and licence.



Filing Structure:

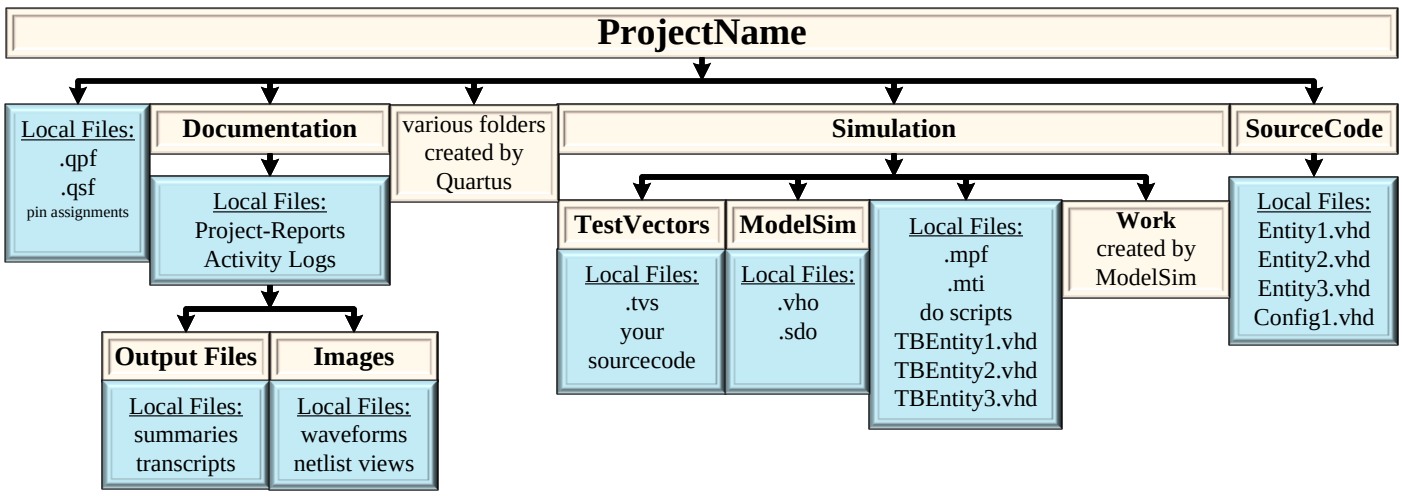
A single project will contain many files and folders, some of which will be created by Quartus & ModelSim. For all projects use the filing hierarchy described below.

Create a tree of folders as shown.
Scripts that you create or that I create on your behalf will assume that files may be located accordingly.

- Use the convention that files associated with a specific design entity are named with a prefix, using the entity name "EntityName"
  or
  a suitable abbreviation using capital letters ie. "**EN**".
- Testbenches have the same name as the entity being tested, with the prefix "**TB**".
- The root of the ModelSim project folder should contain
  - ✓ VHDL-Testbenches & Test Vectors, VHDL-Configurations,
  - ✓ ModelSim Scripts, and Transcripts.
- The folder Simulation/ModelSim is used by Quartus.
- The folder TestVectors is to be used as workspace for programs that generate test vector files.

Avoid directory names in the path that contain "spaces", use underscores instead.

ProjectName:	The top-level folder to be used as Quartus' project folder.
Documentation:	Used to collect files needed for project documentation.
SourceCode:	Used to store the synthesisable VHDL design files.
Simulation:	Used as the ModelSim project folder. (contains testbenches, test vectors, scripts)
ModelSim:	Used by Quartus to store post-fit netlists and timing files.
TestVectors:	Used by you to create Test Vectors using a programming language of your choice.



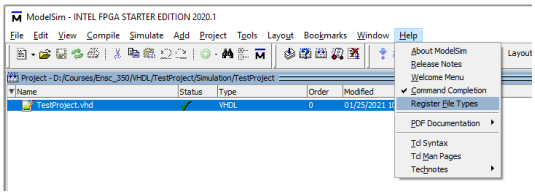
Setting up ModelSim:

ModelSim Project Files:

A general design will contain many VHDL source files for Entities and Testbenches.

ModelSim must know about theses files and related settings, as such, your first step will always be to create a ModelSim Project File, "ProjectName.mpf".

- Store the ModelSim project file within the folder "Simulation"
- To ensure that you can start ModelSim by double-clicking on a ModelSim Project file use the menu help->register filetypes. (you will need to restart the OS)
- Verify that a ModelSim icon is now associated with .mpf files.



Notice that ModelSim creates:

- a folder called "work",
- a ModelSim Project File, (.mpf)
- a Transcript File, &
- maybe a memory data file. (.mti)

- WARNING: do not use the OS to register .vhd files with ModelSim. An infinite recursion will occur. When opening, ModelSim will try to open historical .vhd files which will then invoke a new instance of ModelSim.
- If registering using the OS, It is best to register .vhd files with your chosen text editor. (ie. Notepad++)

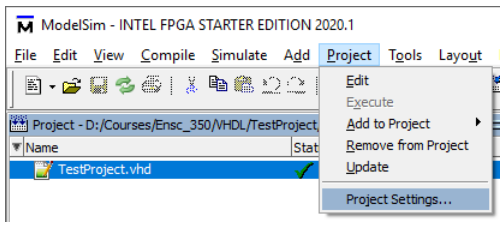
Controlling Compiler Output:

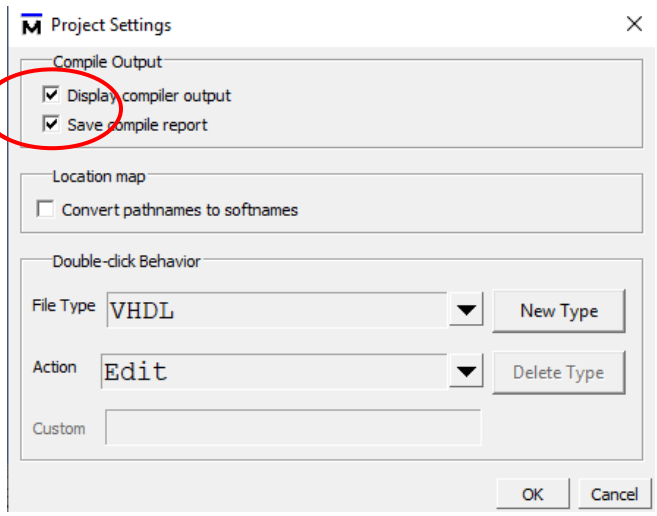
Make a projects generate a full compiler report.

- With project pane active,

  select menu, project->project settings.

  Make sure that both compile output flags, are ON
- Sometimes it is preferrable to turn this setting OFF.
  Compiler output is directed to a transcript file. If you wish to create a macro by editing the contents of the transcript file, the compiler output would obscure the commands in the transcript file.





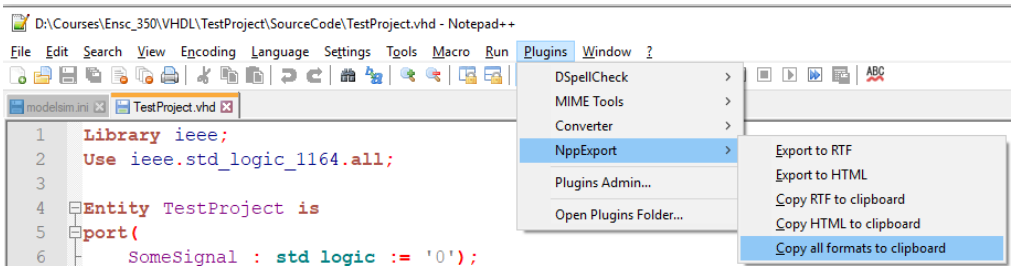
Editing SourceCode:

ModelSim contains it's own text editor that interfaces with the menu/window system of the environment. Standard ascii text files do not contain formatting information. File formats such as "rich text" are needed if you wish to retain syntax highlights with the source code.

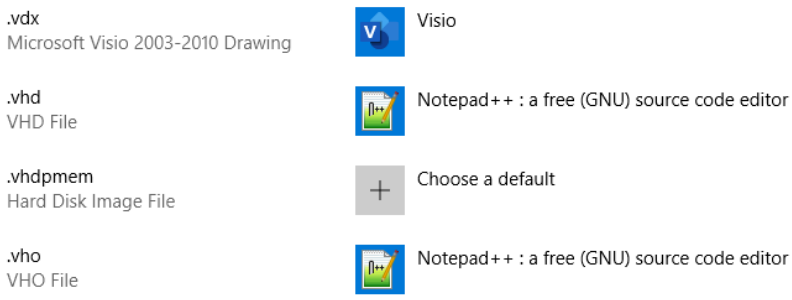
Notepad++ is a simple text editor that allows export with syntax highlighting. You must include syntax highlights when placing source code in your final documents.

You may choose your favourite text editor but remember, final documentation must contain syntax highlights. If you use Notepad++:

- Source code with highlights may be transferred to the clipboard using the menu Plugins->NppExport->Copy all formats to Clipboard

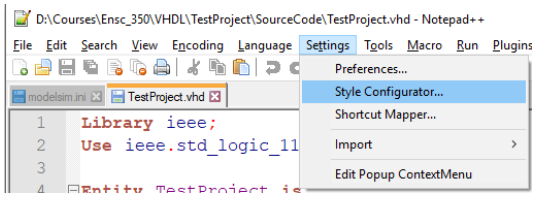


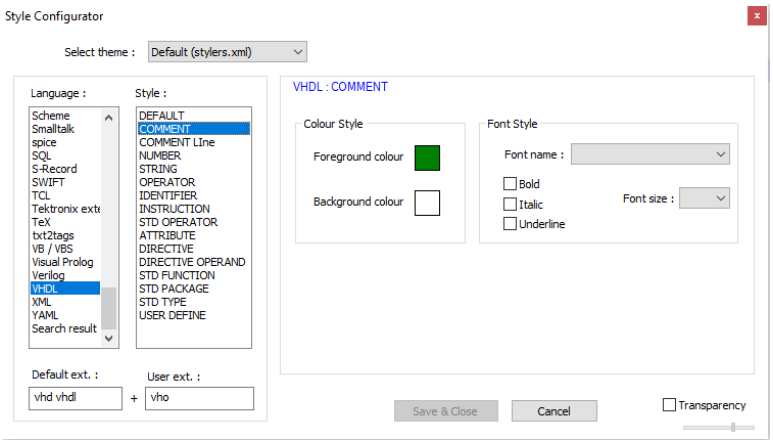
- I set the OS to associate .vhd files with a text editor rather than an IDE. This makes sense because the OS will not allow both ModelSim & Quartus to have associated .vhd files. The applications may however be set to invoke the text editor.



Editing SourceCode:

- You can modify the highlighting rules using the menu Settings->Styles Configurator Then, on the left configure colours for VHDL language elements. I set the colours to be similar to those of the ModelSim default editor.

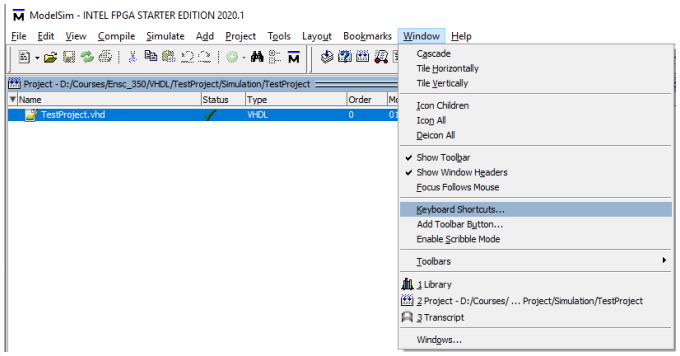


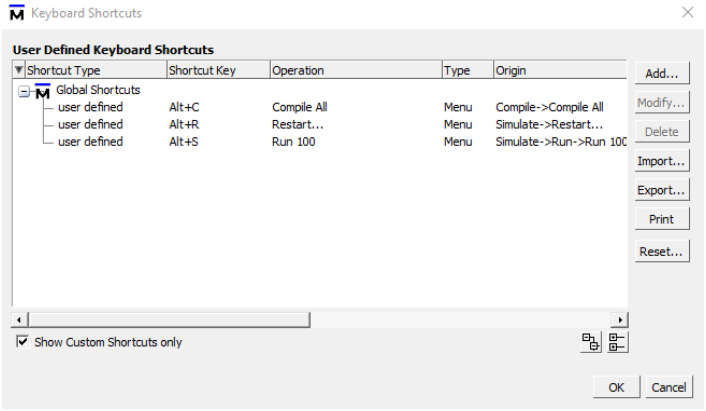


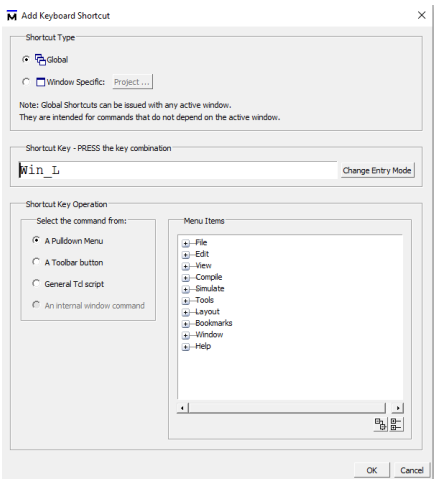
Creating Keyboard Shortcuts:

You can create custom keyboard short cuts using the menu window->keyboard shortcuts.

- As you progress you will discover particular sequences that occur commonly. Get into the habit of making shortcuts. The ability to delay gratification is a virtue.





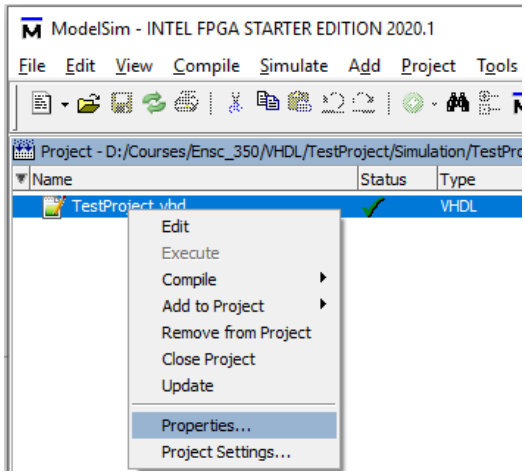


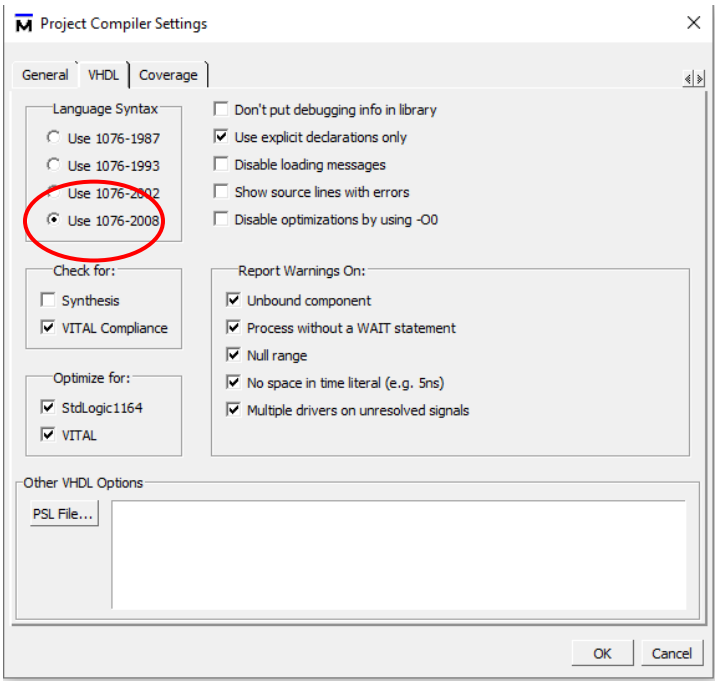
VHDL 2008 Compatibility:

A VHDL source file may need to be compiled using the VHDL-2008 compatibility.

- Testbenches may or may not use File I/O which requires VHDL-2008 compatibility.
- Some VHDL useful conversion functions require VHDL-2008 compatibility.
- You should check a source file's setting by <right-click>->properties to bring up the dialog. Make sure that VHDL-2008 is selected as a default in the project compiler settings.

This may not be necessary when a scripts explicitly requests 2008 compilation.





Default Simulation Timing Resolution:

The default is set to ps. For now we'll leave the default alone.

We can change the default timing unit later. The default should be chosen to be a convenient unit based on the speed of the selected FPGA.

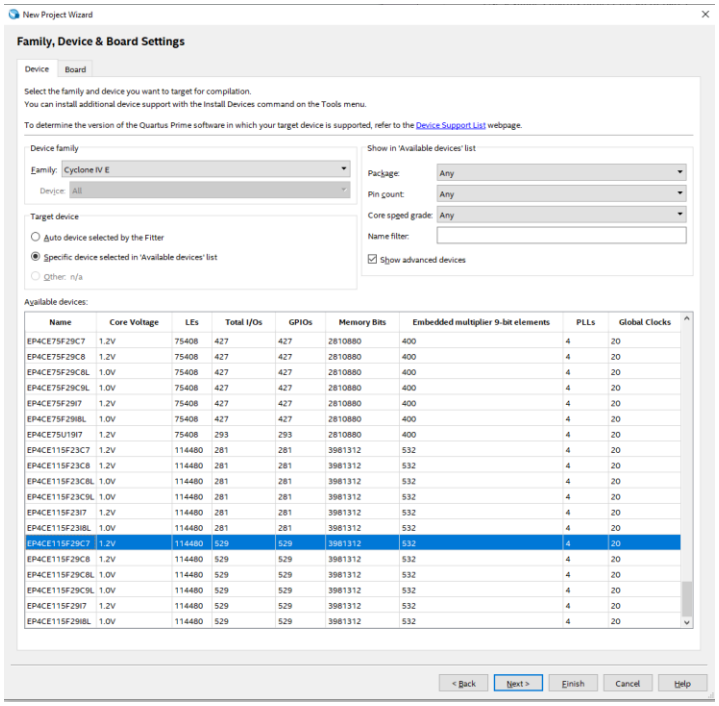
Setting up Quartus:

Quartus Project Files:

Like ModelSim, and generally all IDEs, working with Quartus requires creation of a project file at the onset. The filing organisation described previously specifies that **Quartus Project Files** (.qpf) should be stored within the root project folder.

Start Quartus.

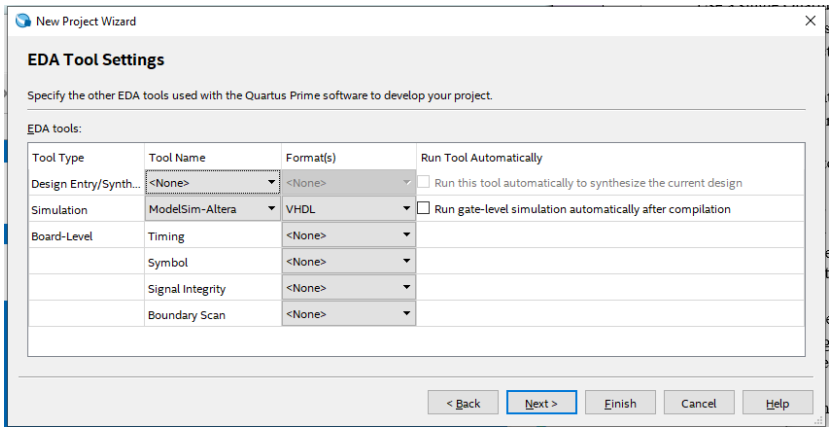
- Make a new project using a Cyclone IV FPGA. – EP4CE115F29C7



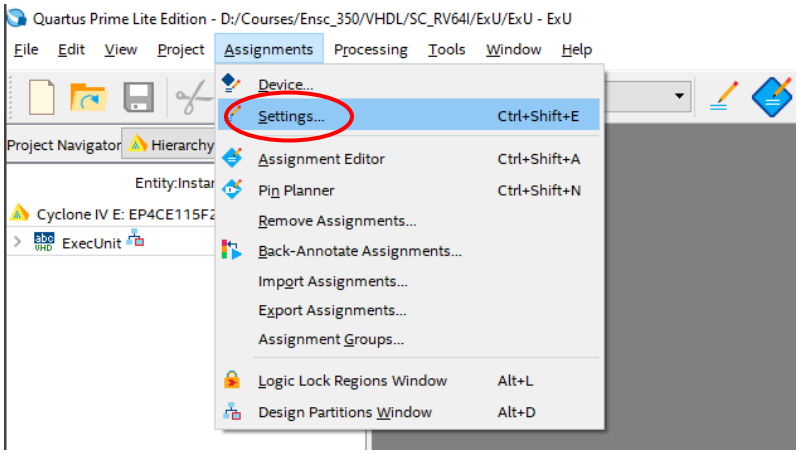
Notice that Quartus creates:

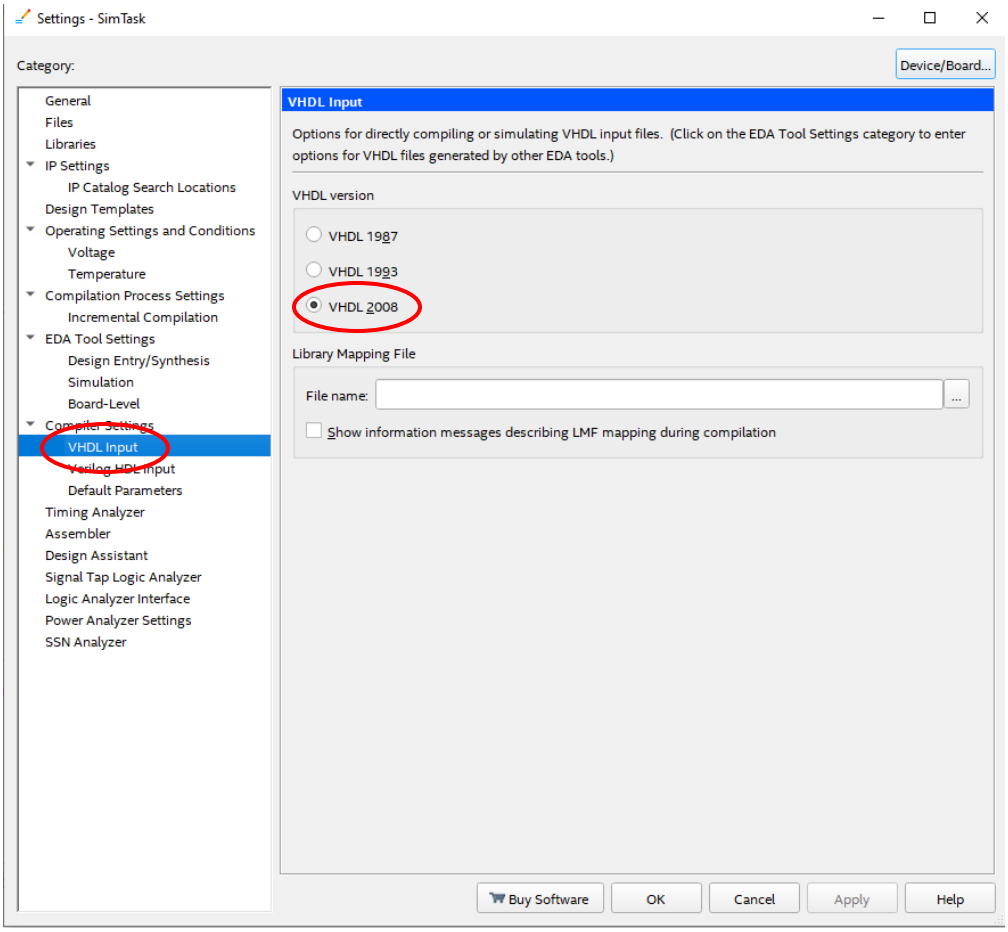
- a folder called “db”,
- a Quartus Project File, (.qpf)
- a Quartus Settings File, (.qsf) &
- a Quartus Workspace File. (.qws)

- Under the EDA tool settings, choose ModelSim-Altera, format is VHDL
- do not enable – “run gate-level
- You can also activate this dialog by double-clicking the EDA Netlist writer – Edit settings.



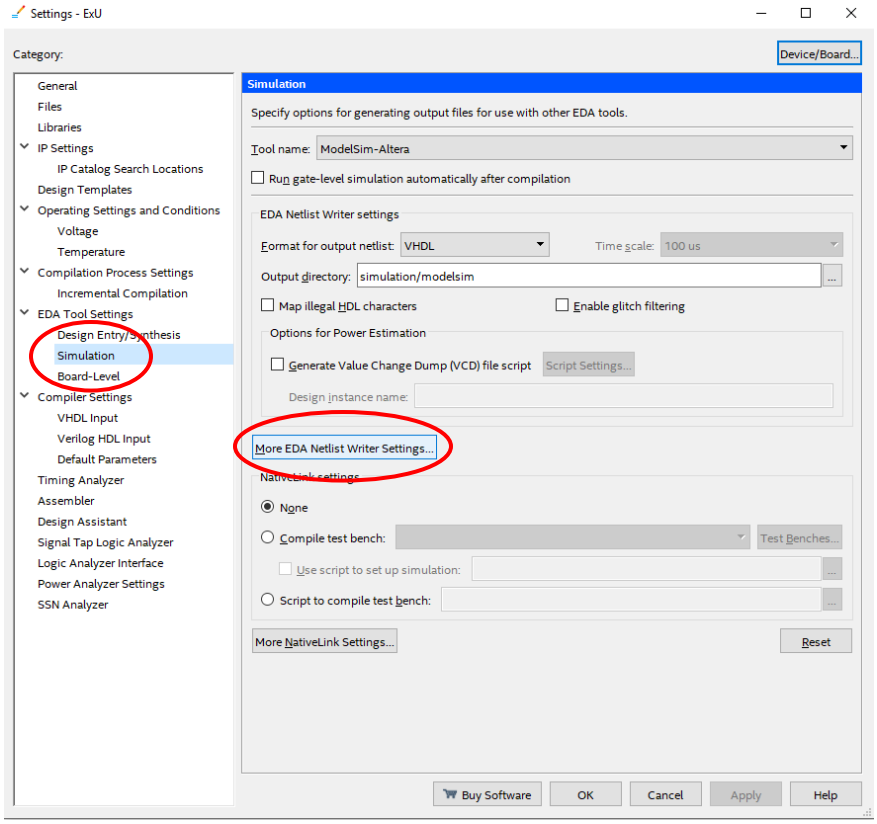
At any time you can choose/verify the **version of VHDL** being used by the current project.
bring up the project settings, Assignments->Settings. (Or ctrl-shift-E)

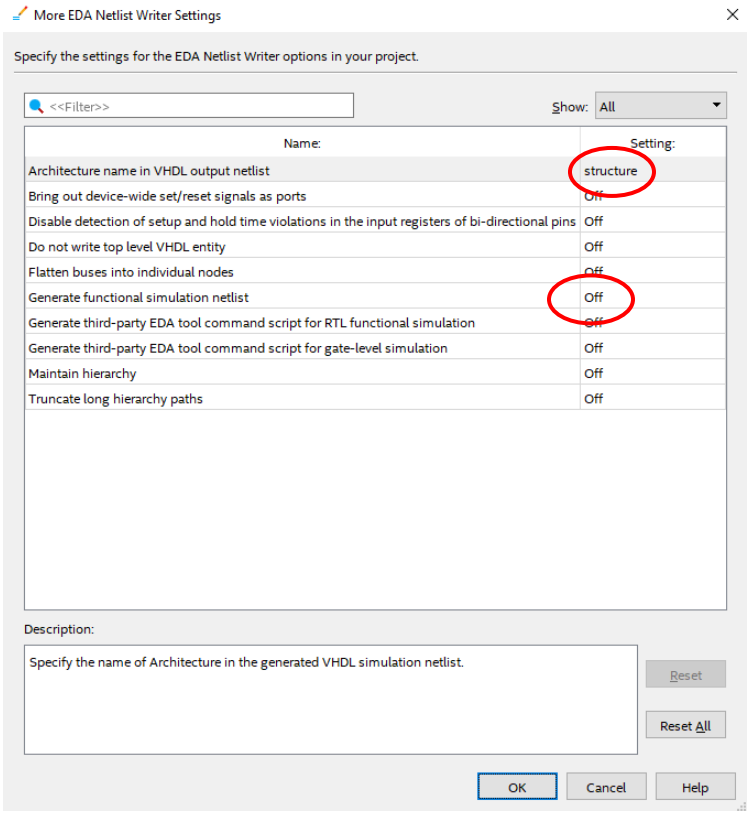




Select EDA tool Settings – Simulation

Click – More EDA Netlist Writer Settings – turn **OFF** "Generate functional simulation netlist"





Close Quartus.

From now on always start Quartus by double-clicking on the project icon.