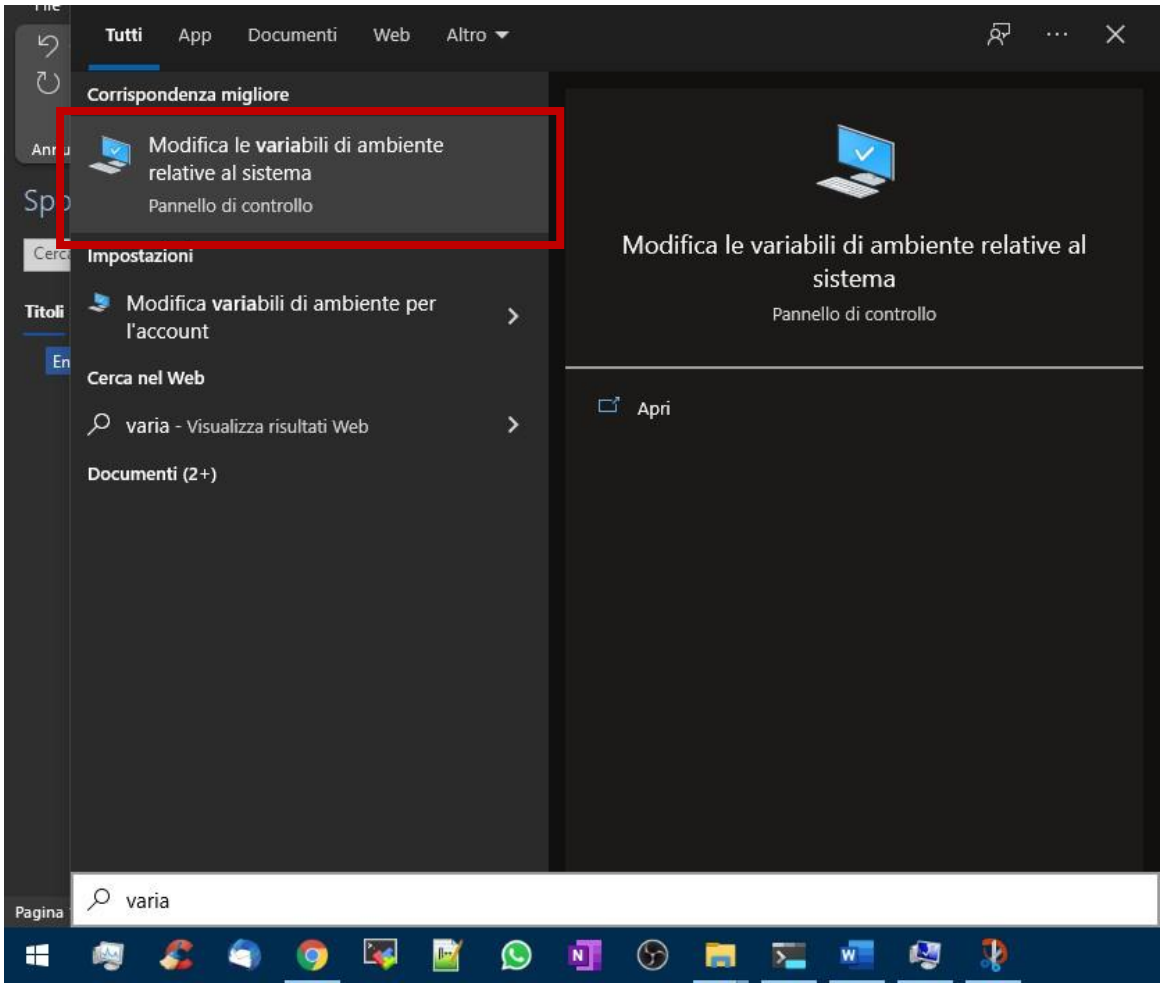
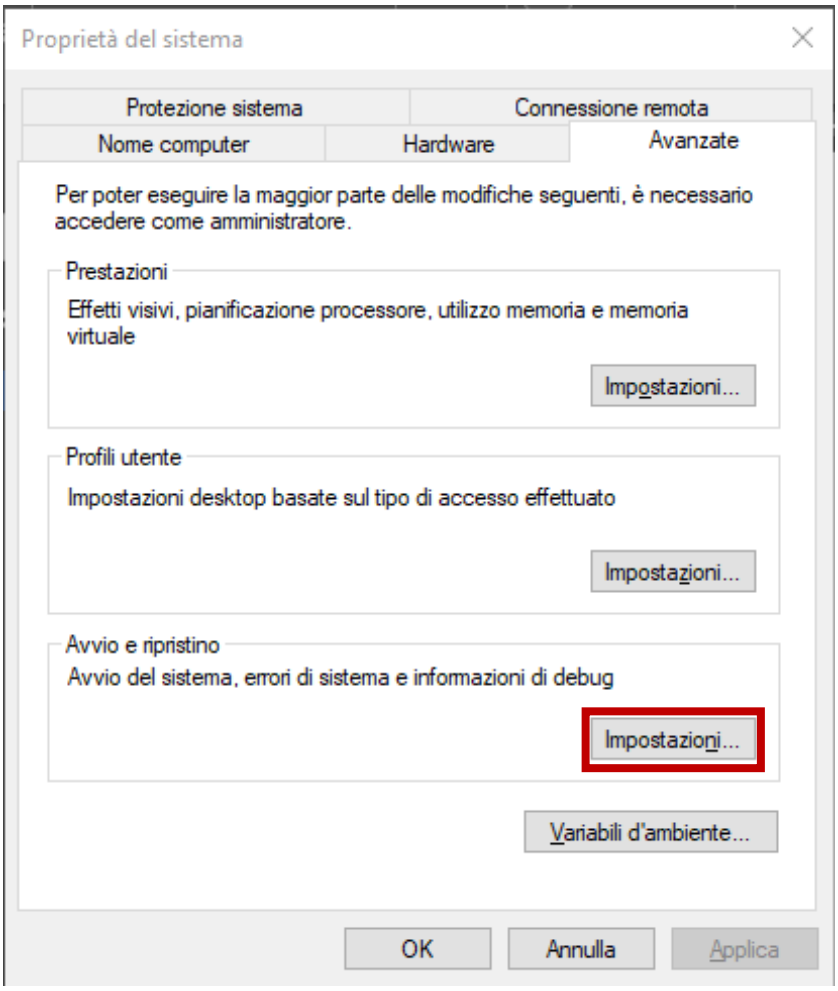
Environment setup

In order to make working the **scripts** inside the folders quartus/ and modelsim/, **Python** is required (both versions 2.x and 3.x will work), with packages **os**, **sys**, **shutil** and **glob**; in addition, you have to set some environment variables, according to your OS.

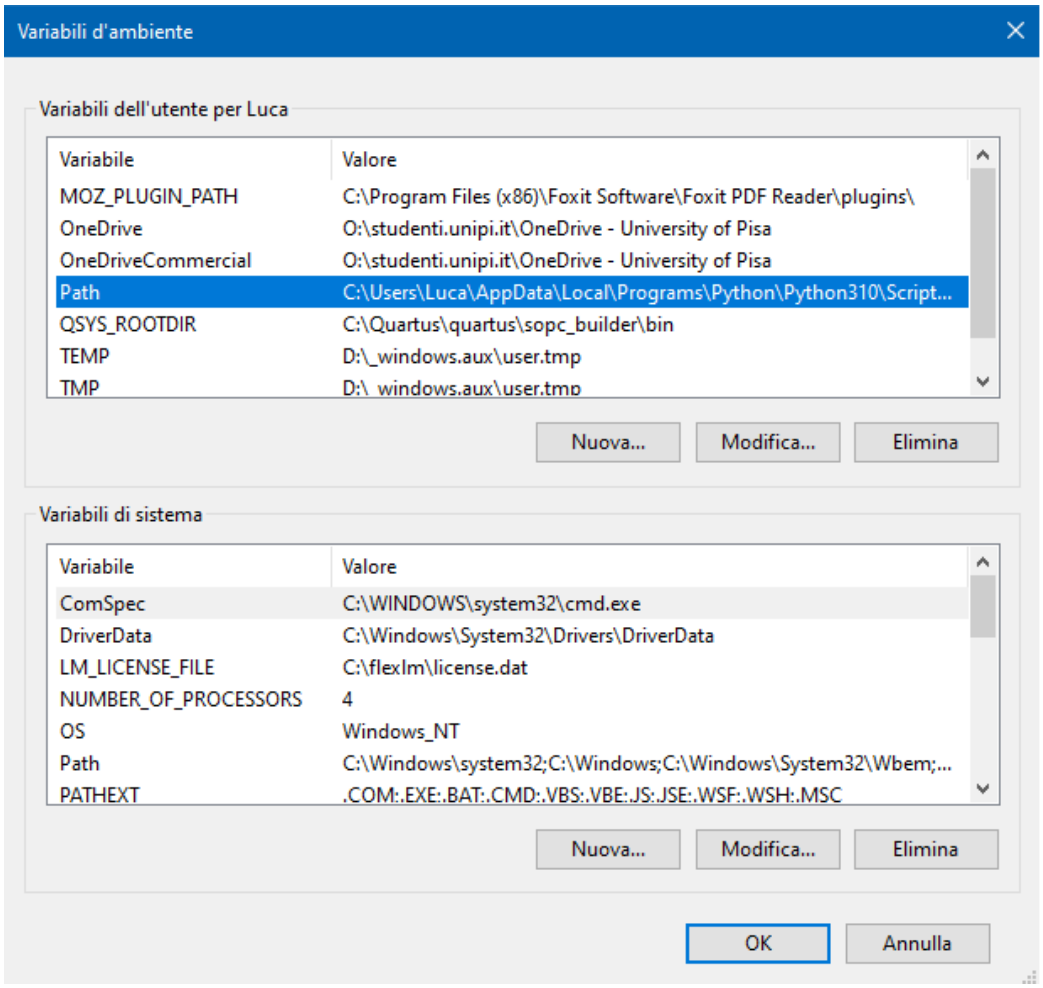
- **Windows OS**
  - Press the super-key on the keyboard (the one with the Windows logo) or the Start button (with the mouse) and type on the keyboard searching 'environment variables' ['variabili di ambiente']: 'envir' ['variab'] should be enough to make compare the entry '**Modify System environment variables**' ['**Modifica le variabili di ambiente relative al sistema**']; open it by clicking on it;



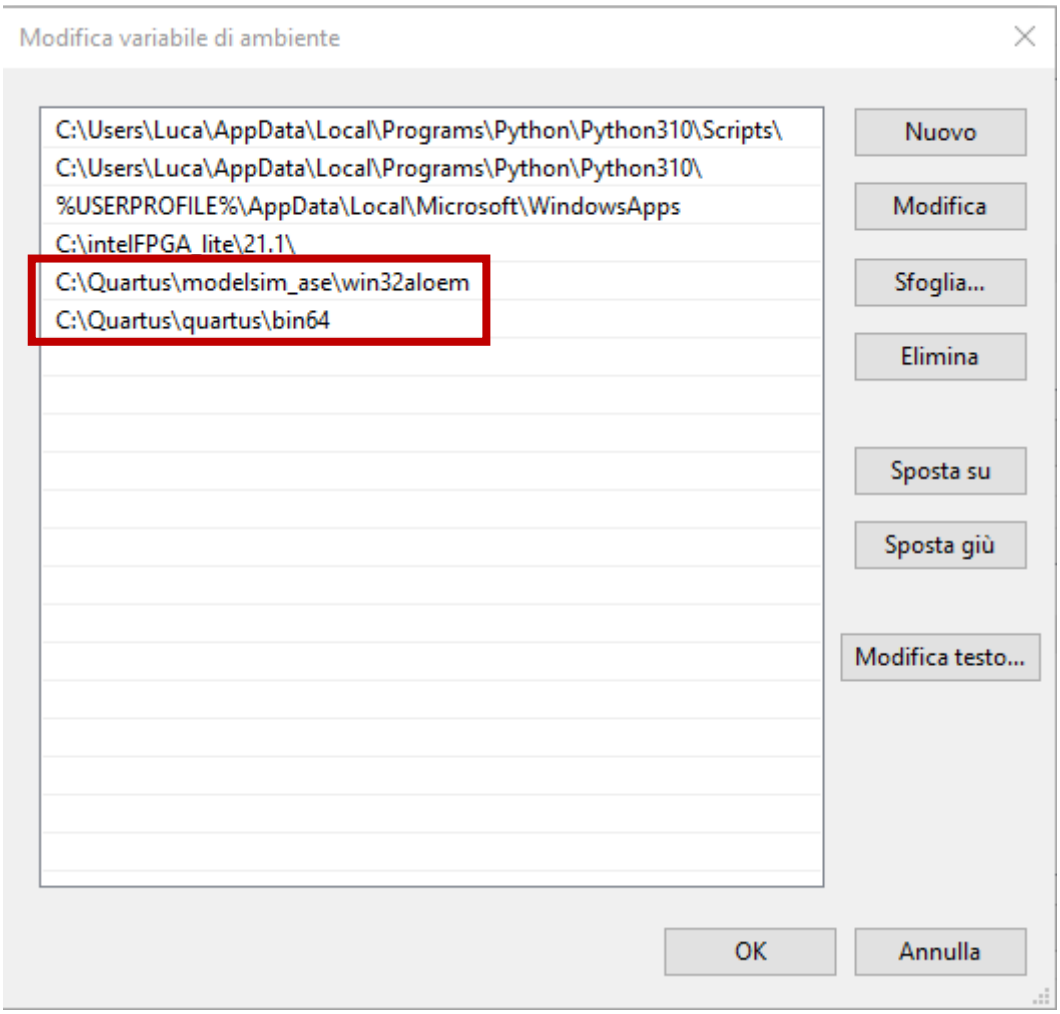
- Once open, click on the button '**Environment variables**' ['**Variabili d'ambiente**'] on the right corner at the bottom of the window.



- In the section '**User's variables for …**' ['**Variabili dell'utente per …**'], top èart, double click on the variable '**Path**'



- If not already present, add the path to the Quartus executables folder (i.e., the one containing quartus.exe, quartus_sh.exe, …) and to the Modelsim executables folder (i.e., the one containing vsim.exe), then click on '**OK**' button



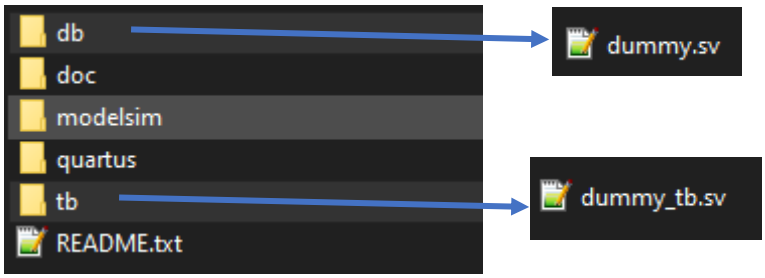
- To check everything is fine, open a Command Prompt terminal and try to type and run **quartus** and **vsim**

- **Linux OS**
  - Usually, the paths to Quartus and Modelsim executables should be automatically added as environment variables, anyway, if not so, individuate them and then add as environment variable executing the command:
    - **export PATH=$PATH:<quartus_path>**
    - **export PATH=$PATH:<modelsim_path>**

  - To verify everything is ok, open a terminal a try to type **quartus** and **vsim**: they should be autocompleted by pressing TAB key and the respective programs should be launched.

  - If you want to permanently add such variables (in case you had to manually add them), append the commands 'export …' to your **.bashrc** file inside your home folder (/home/<user name>: the .bashrc is a hidden file).

Environment quick guide

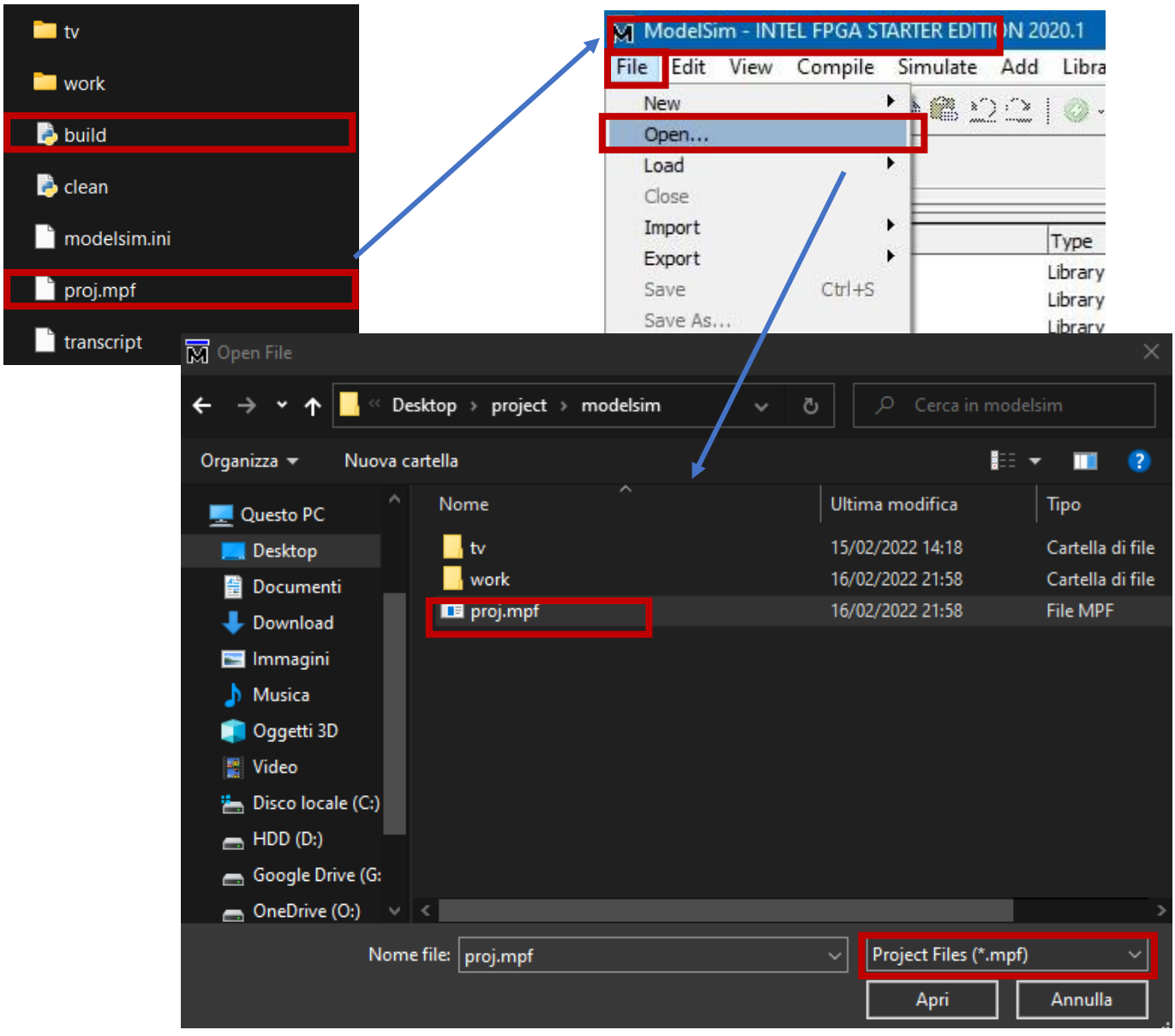


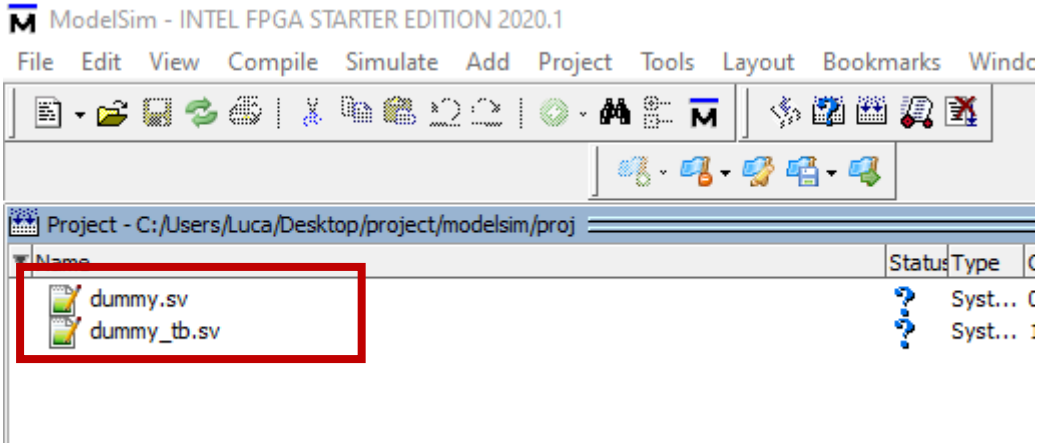
Assuming to have the design file **dummy.sv** inside folder **db/** and the testbench **dummy_tb.sv** inside the folder **tb/** (they also may be blank files), move inside folder **modelsim/** and run the script **build.py** with python:

python build.py

The command can be used on both Windows OS and Linux OS, using, respectively, the command prompt (WIN+R > cmd) or the terminal.

This will create a **Modelsim project** (**.mpf**) including all the **SystemVerilog files** inside the folder **db/** and **tb/**. Then launch Modelsim and open the .mpf file.





Now you can edit the HDL files, by using the Modelsim text editor, or an external text editor, compile them and simulate, according to how much shown at during the lectures.

In case of usage of test vectors, they shall be stored inside the folder **modelsim/tv/** and invoked within the testbench(es) using the relative path **./tv/<test vectors file>**. E.g.: **$fopen("./tv/stim.txt","r");**

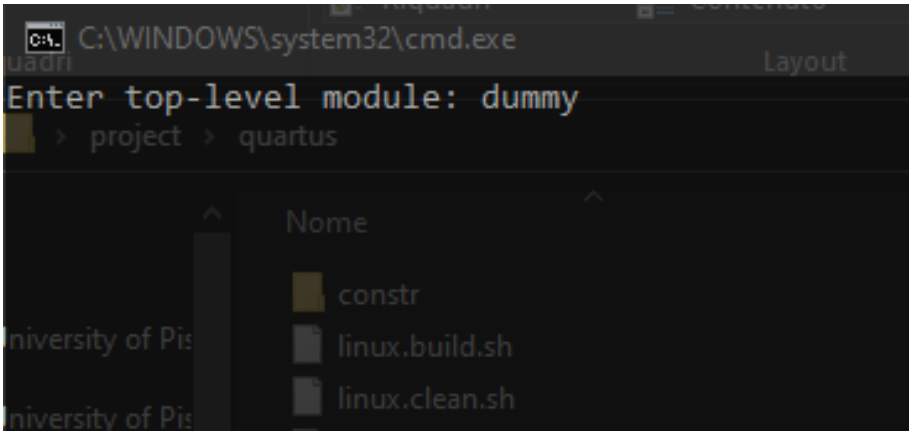
Similarly, for Quartus and synthesis move inside the folder **quartus/** and run the script **build.py** with python:

python build.py

The command can be used on both Windows OS and Linux OS, using, respectively, the command prompt (WIN+R > cmd) or the terminal.

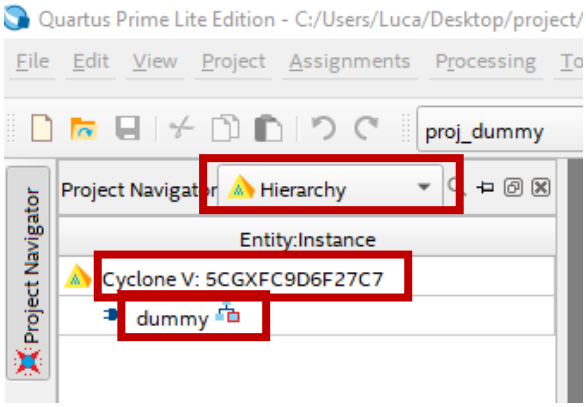
This will create a Quartus project file (**.qpf**) including all the **SystemVerilog design files** inside **db/** and the eventual design **constraint files** (**.sdc**) stored inside **quartus/constr/**: even if not edited yet, for instance you can create blank constraint file inside the dedicated folder, so that it is included within the Quartus project and modify its content later. Quartus project file automatically will already include the FPGA device chosen during lectures and the explanation of the tools.

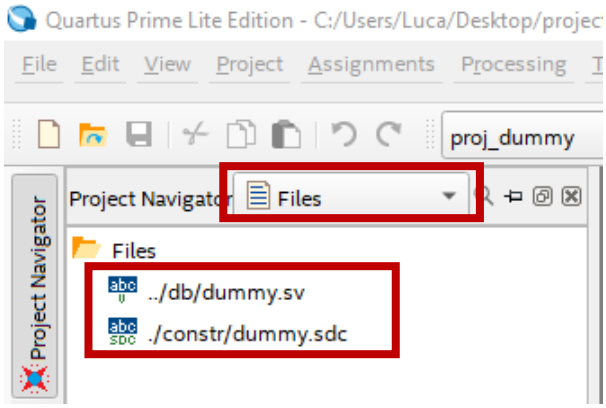
Before creating the Quartus project, the script will prompt for the name of the top-level entity/module: **type it** and then press **ENTER** button



To open Quartus project, just double click on .qpf file.

Now you can use Quartus as shown during lectures and perform synthesis and fitting (and eventually STA).





In case you want to restart from scratch, just run the script **clean.py** with python:

python clean.py

This applies to both folders **quartus/** and **modelsim/**; the command can be used on both Windows OS and Linux OS, using, respectively, the command prompt (WIN+R > cmd) or the terminal.

!!! Warning !!! the script **clean.py** clears the content of folder quartus/ (or modelsim/) deleting all files but the original ones, i.e.:

- quartus/
  - build.py
  - clean.py
  - quartus.build
  - constr/ [the whole folder and its content is kept unmodified]
- modelsim/
  - build.py
  - clean.py
  - tv/ [the whole folder and its content is kept unmodified]

In addition, also all other environment folders, i.e. db/, tb/, and doc/, and their content are not affected by the clean operation trough scripts.

Note: before delivering the project for the exam, **please run the clean scripts inside folders quartus/ and modelsim/**, so that Modelsim and Quartus projects files are not included within the delivery, but they are included all and only the files relevant to the project, i.e.

- **design file(s)** inside db/ (db/ is not affected by clean scripts)
- **testbench(es) file(s)** inside tb/ (tb/ is not affected by clean scripts)
- **project report** inside doc/ (doc/ is not affected by clean scripts)
- **testvector(s) file(s)** inside modelsim/tv/ (modelsim/tv/ is not affected by clean scripts)
- **constraint(s) file(s)** inside quartus/constr/ (quartus/constr/ is not affected by clean scripts)