Project rules

According to the specifications of the project assigned to you, the development of the project consists in:

- Creating a high-level model (C/C++ or Python),
- Writing the RTL code of the module in SystemVerilog,
- Writing one (or more) testbench(es) in SystemVerilog and testing the module with Modelsim,
- Performing the synthesis and the implementation of the module on the FPGA device 5CGXFC9D6F27C7 with Quartus (including STA),
- Writing a report.

The report will be used as a starting point for the evaluation of the project. The evaluation will also include the verification of the simulation and implementation processes and results.

Report content

1. **Project Specifications**
   An introduction to the project, including the specifications and their analysis.

2. **High-level Model**
   The description of the high-level model (C/C++ or Python)

3. **RTL Design**
   The description of the RTL design architecture, with block diagrams, schematics (functional and/or architectural), FSM diagrams, etc ...

4. **Interface Specifications and Expected Behavior**
   The description of how your module should be used in terms of how inputs should be provided (including timing) and what the corresponding output(s) would be. In practice, a kind of user manual, which may also include the waveforms at the input and/or output for several use cases (main case(s) and possible corner case(s)).

5. **Functional Verification**
   The description of how you tested your module, the testbench(es) architecture, and the test(s) you performed to verify that your module works according to the expected behavior described in Section 4.

6. **FPGA Implementation Results**
   The presentation of the steps you have performed in Quartus (including the FPGA device you have used), the corresponding results such as resource usage, maximum frequency, ... including comments on the results.
      - Remember that the maximum frequency is the result of the Static Timing Analysis (STA).

Note: to generate the schematics or block diagrams for the project report, any software tool for drawing can be used, anyway it is highly suggested to use the Microsoft tool **Visio or DrawIO**. It is included within the package of free licenses that University of Pisa offer to their students for the Microsoft products: for more information refer to http://www.sid.unipi.it/polo6/studenti/licenze-software/. To download such tool, access at https://azureforeducation.microsoft.com/devtools, download the installer and the license activation key

Project delivery

Deliver a .zip file containing:

- All the RTL design files,
- All the testbench(es) files, including test vectors (if any),
- All the high-level model files,
- Constraints file(s) for STA (if any),
- The project report (PDF or Word).

The .zip archive for the delivery must have the following name:
project_930II_<academic_year>_<surname_1>[_<surname_2>].zip

- <academic_year> must be the current academic year in the format <yyyy>_<yyyy+1>; e.g.: for the academic year 2023/2024, it must be 2023_2024
- <surname_1> [_<surname_2>] must be the list of your surname(s)

Full example(s):

- project_930II_2023_2024_crocetti.zip
- project_930II_2023_2024_crocetti_saponara.zip

For the delivery, you can create the .zip archive from the work environment that is sent you upon the project assignment: in this case, please, run the clean.py scripts inside the modelsim/ and quartus/ sub-folders before creating (and delivering) the archive. For more information refer to the work environment guide inside the doc/ subfolder: **work_env_guide.pdf**.

Send the project archive for the delivery to Pietro Nannipieri (pietro.nannipieri@unipi.it), and Prof. Sergio Saponara (sergio.saponara@unipi.it) as Carbon copy (Cc).