

P51 - Submission Guidelines

Dr Noa Zilberman

Lent, 2017/18

Last updates: February 19, 2018

This document provides instructions for the submission of the practical assignment. The document covers both the list of items that need to be submitted and a description of each item.

Submission details:

Date: Tuesday, 24/4/2018.

Time: 12:00

Submission website: Moodle

Submission type: Individual.

1 Individual Submission

While projects are being done in pairs, the project submission is individual. Each student must submit the project on Moodle. While code submission and various components of the submission are shared, the project documentation is individual (with some shared contents). Refer to Section 4 for specific details.

2 Submission Contents

The following items must be included in the submission:

- Project code base (shared).
- Documentation (individual).
- Performance profile (shared).
- Project evaluation (shared).

Each of the above items should be uploaded separately (i.e. you are required to upload exactly 4 files).

3 Project code base

The entire code base of the project should be submitted as a single (compressed) file. This includes all files under *NextFPGA-SUME-live* folder. To prepare the folder for submission, make sure to remove all redundant files:

```
cd $SUME_FOLDER
make clean
```

Make sure to include in your submission the FPGA programming file (*reference_switch.bit*). The same code base can be submitted by all team members.

4 Documentation

A document describing your project is required. This document should be submitted in pdf format.

The submitted documentation is individual. A few sections may be shared, as described below.

The document is expected to take between 5 and 10 pages. Figures, tables and citations are not counted toward page length. There is no word count restriction, but exceptionally long documents (e.g., 40 pages of text) will not be considered favourably. Both one and two column page formats are acceptable.

Please use a font size of at least **10pt**.

4.1 Structure

While not mandatory, we suggest the following format for the documentation:

1. Executive summary (up to 1 page, shared) - summarizing the objectives of the project, the selected architecture, project achievements, special highlights or major issues encountered.
2. Project objectives (shared) - a description of the objectives of the project. This section is not expected to be long.
3. Project Architecture (shared) - a detailed description of the architecture of the project. It is expected to include a high level block diagram of the architecture, and a discussion of design choices and their implications on functionality and performance.
4. Assignments (shared) -a table (which may be accompanied with text) explaining how the development effort was divided between team members. It is expected that each member of the team will take complete ownership of at least several modules in the design, but some parts of the project, (e.g., architecture, integration, evaluation) may be shared. It is allowed not to have exactly the same number of modules assigned to each member of the team, as some modules require

significantly more work. The assessment will take into account the complexity of changing / implementing each module.

5. Design and implementation (**individual**) - a detailed description of the implementation / changes of each module owned by the author, including a description of design limitations and challenges you had to face. At least a paragraph should be dedicated to each module. A module is not expected to take more than 1 page, though exceptions are allowed.
6. Performance profile (shared) - text accompanying the performance profile spreadsheet and explaining its results. It is expected to include a graph visualizing the results for different packet sizes. You can reflect here on limitations faced in previous sections. The length of this section may vary according to your results - good (and simple) outcomes will likely to take only a few paragraphs of text, while you may opt to extend if your results are less favourable or non-intuitive.
7. Project status (shared) - describe the status of the project at the time of submission. While the project is expected to be fully working, if it is not, please describe it here. Describe what was achieved, (e.g., “fully working data path”) and what was not (e.g., “10G port fails to handle packets bigger than 512B”). This section will enable assessing the following section.
8. Evaluation (shared) - summarizing the functional and performance evaluation tests and results. The performance results are expected to be presented as graphs, though performance highlights (e.g., minimum latency, maximum bandwidth) are likely to be discussed in the text. You can refer here to specific files included under the project evaluation folder, but please refrain from simply pointing to these files, e.g., “we evaluated the latency, and the results are detailed in file X”. Instead say “As file X shows, the minimum latency for packet size Y is A, and the maximum latency for the same packet size is B”.
9. Discussion and conclusion (**individual**) - this section can be used to reflect on your project, highlight points that you consider important to the assessment, and to summarise your work. It is not expected to be long.

5 Performance Profile

The performance profile of the project should be submitted as a spreadsheet.

Allowed file formats are: csv, xlsx, ods.

Do not submit the performance profile in pdf format.

The same performance profile can be submitted by all team members.

The performance profile should show, for each packet size (64B to 1518B [on the wire]), the projected performance in each module (IP core).

In the unfortunate case where not all modules were implemented in time for the submission, also include the performance profile based the actual project implementation (which is likely to be based on the performance profile of the modules in the Reference-Switch). This should be included within the same file, and not as an additional file.

6 Project Evaluation

The entire evaluation environment of the project should be submitted as a single (compressed) file. This includes all files under *OSNT-SUME-live/p51/* folder.

It is expected that each project will have 2 types of evaluation environments:

- A functional validation environment, included in the submitted code based (under *NetFPGA-SUME-live*). This should include existing or new tests (located under `$NF_DESIGN_DIR/test/`) and detailed in the documentation.
- A performance evaluation environment, described in this section.

The performance evaluation environment is supposed to provide information on the latency and throughput of the designed project.

If you only ran the tests provided by the course's team, include the *OSNT-SUME-live/p51/* folder "as is", including all log files. Make sure that the log files contain only your results, and not the results of previous users of the setup. If you also created your own tests, include a README file describing the setup, any installation and configuration requirements, as well as all result log files.

Note: At the time this text is written all projects aim to achieve low latency, and the test environment is optimized for that. Changes at a later stage may reflect on this section.

The same evaluation results can be submitted by all team members.

7 Attribution

Although you are encouraged to discuss ideas with others, your programs are to be completed independently and must be your own, original work, or the work of you and your partners. Whenever you obtain significant outside help (from other students, the instructors, etc.) you should acknowledge this in your program write-up, e.g. The idea for how to implement the scheduler came from a discussion with Alice and Bob. You can never get in trouble for plagiarism if the help is properly credited.

The Computer Laboratory uses screening systems to compare student submissions. Such systems are highly effective, and we use it to identify submissions that need to be scrutinized further by course staff. Using systems of this type is common practice at many universities, and it has proved to be an effective deterrent to improper collaboration.

Programming is something you learn by doing. If you copy someone else's work, you can expect the following:

- You will not learn what the assignment was meant to teach you.
- Your copied work will be brought to the attention of the relevant administration.

(This text was adapted from the Cambridge P33 course website, which blatantly plagiarised it from Stanford CS344 web page)

8 Dissemination

We encourage all students to disseminate their work. We encourage students who developed successful projects and interesting architectures to submit their work as a paper to a conference in this field. We otherwise intend to turn the outputs of this course into a technical report that can inform other students, reflect on good (and bad) design choices, and provide architectural ideas. All contributing students will be authors on this technical report and all copyrights will be maintained.

If you wish to opt-out from the Technical Report, please notify the course's team. You should also highlight this in your submission.

We encourage students to contribute (working) IP cores and projects to the NetFPGA community. For details, refer to <https://github.com/NetFPGA/NetFPGA-SUME-public/wiki/Contributing-Code> and to the course's team.