National Tsing Hua University
Department of Electrical Engineering
EE4292 IC Design Laboratory (積體電路設計實驗)
Fall 2021

Term Project (30%)

Assigned on Dec 2, 2021
Due by Jan 20, 2022

Objectives:

You will need to use what you learned in this course to 1) implement a target algorithm/application into RTL codes and 2) do a basic P&R flow on it to derive its layout and chip performance. The topic can be any digital implementation (see details below) which has sufficient complexity (assessed subjectively by the teacher and objectively by area, e.g. gate count > 20K). The proposal and final report should cover discussions on four aspects: functionality, specification, implementation, and verification.

Project Examples:

1. **Enhancement to the CPU in labs** (default difficulty: low-medium)
   Try to enhance the CPU in non-trivial ways. For example, you can handle data hazard completely by hardware, instead of by the workaround NOP insertion. You can also add multiplication and make it five-stage pipelined. Other kinds of complex instructions can also be added. For this topic, you will have to give a solid test plan to verify your design.

2. **2048-point Fast Four Transform** (default difficulty: low-medium)
   Try to implement the well-known FFT into digital circuits. You can implement more than one kind of architectures and compare them in terms of area, speed, and power. An advanced alternative is a reconfigurable processor for 512-/1024-/2048-/4096-point FFT.

3. **Floating-point ALU** (default difficulty: low-medium)
   Try to implement floating-point arithmetic operations. You can refer to the format of IEEE 754-1985 (http://en.wikipedia.org/wiki/IEEE_754-1985). An advanced alternative is to implement some usual analytic functions based on this ALU, such as sin, cos, log, *etc*.

4. **Matrix Multiply Unit** (default difficulty: low-medium)

Try to implement an efficient hardware accelerator for matrix multiplication which is a key operation in deep learning. You may refer to the matrix multiply unit in Google TPU (https://cloud.google.com/blog/big-data/2017/05/an-in-depth-look-at-googles-first-tensor-processin g-unit-tpu).

5. **Advanced CNN Accelerator** (default difficulty: medium-high)

Try to implement a hardware accelerator for advanced CNN models. You may discuss the trade-off between hardware complexity and recognition accuracy.

Teaming:

One team can have **two to three** members. And the grading will depend on the contribution of each member.

Project Schedule:

Dec 2, 2021:	Project announcement
Dec 23, 2021:	Project proposal (topic and team member identified; check with TA 林楷平)
Dec 24, 2021:	Project difficulty assigned
Jan 6, 2022:	Interim presentation (front-end design: software (if any), RTL and testbench)
Jan 20, 2022:	Final presentation; submission of Verilog codes and project report (no late submission is allowed)

Grading Rule (100%):

1. Completeness (25%)
    - Basic flow: RTL code, testbench, RTL simulation, synthesis, P&R, post-layout simulation
2. Difficulty (25%)
    - High: 20-25%; Medium: 10-20%; Low: 5-10%
    - If floorplan utilization ≥ 90% and P&R timing ≤ 1.1*synthesis timing, you can get an extra 2-point bonus for the semester grade.
3. Presentation (25%)
    - Interim: 10%
    - Final: 15% (10% by teacher & TA, 5% by students' cross-evaluation)
4. Documentation (25%)
    - Proposal (3%)
    - Final report (22%)

Deliverable (per team basis):

1. Project proposal briefly describing the following: (see the template for reference)
    - Team members
    - Your topic*
2. Interim presentation slides (in four minutes) including:
    - Functionality and specification of your project
    - Current status of your project on testbench and RTL design

3. Final presentation slides (in five minutes)
   [Note: Focus on novelty and contributions. Leave details to the final report.]
4. Project report summarizing: (in 5 pages, either single- or double-column)
   - Your result*
   - Contribution of each member
5. Verilog code and CAD tool scripts

*Note - Four aspects should be covered:

Aspect	Proposal	Final Report
Functionality	Initial ideas of your design (what it should be)	Implemented ideas (what it really is)
Specification (Area/Speed/Power)	Target performance (how well it should be)	Chip performance (how well it really is)
Implementation	Initial plan (what you will do)	Achieved items (what you really have done)
Verification	Test plan (how to verify functionality and specification)	Verification results