EPFL

Lab. On HW-SW Digital Systems Codesign
EE-390(a)

Course Outline

Prof. David Atienza
Dr. Denisa Constantinescu, Dr. Miguel Peón-Quirós
Mr. Rubén Rodríguez-Álvarez, Ms. Stasa Kostic, Mr. Karan Pathak

Teaching staff

- Prof. David Atienza (david.atienza@epfl.ch) ELG-130
- Dr. Miguel Peón Quirós (miguel.peon@epfl.ch) INJ-237
- Dr. Denisa Constantinescu (denisa.constantinescu@epfl.ch) ELG-137

- Teaching assistants → Lab management
  - Mr. Rubén Rodríguez Álvarez (ruben.rodriguezalvarez@epfl.ch) ELG-135
  - Ms. Stasa Kostic (stasa.kostic@epfl.ch) ELG-121
  - Mr. Karan Pathak (ruben.rodriguezalvarez@epfl.ch) ELG-121

- All materials of the course in Moodle – EE390(a)
  - https://moodle.epfl.ch/course/view.php?id=16115

- Office hours
  - Please send an e-mail to request a slot

Course objectives and outline

- Objectives:
  - Learn the anatomy of complex SoCs, comprising microprocessor(s), peripherals and accelerators
    - Assembling the HW and programming the SW
  - Learn the basics of HW/SW partitioning and how to create different types of accelerators
  - Identify trade-offs in system performance, energy efficiency and use of resources
  - Learn to integrate custom HW with a Linux operating system
  - Learn how to profit from high-level synthesis tools to improve your productivity designing HW

- Course dynamic
  - Guided examples
    - Getting familiar with the concepts and tools
  - Develop your own exercises

Lab topics and schedule (may be adjusted)

1. Concept of HW-SW co-design for SoCs
   - Processor access to peripherals and buses
   - Types of peripherals
   - Flow of computation in a SoC with accelerators
   - Zynq-7000 SoC
2. Co-design with HLS for HW description
   - Design flow with HLS
   - Integration of SW and HW in Linux
   - Performance & energy characterization
   - Local memory. Loops. Latency and throughput
   - Design of a basic accelerator for convolutions
3. Design exploration & optimization using HLS
   - Synthesis flow. HLS scheduler.
   - Loop optimizations.
   - Integration of convolution accelerator in a CNN
4. Array optimizations in HLS
   - Types of storage resources
   - Array partitioning & reshaping
5. Cache coherence & virtual memory
   - Allocating memory for DMA
   - Impact of caching on CNN layers computed in SW
6. Interrupts and Linux device drivers
7. Dynamic job scheduling across multiple accelerators
8. Memory hierarchies
   - Caches and buses
9. Final guided project
   - Last four weeks of the semester

Organization and grading

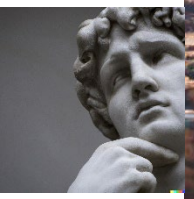
- All classes and lab sessions held on-campus (ELG-022) on Thursdays (14h15-17h00)
- FPGA boards:
  - The FPGA boards can be borrowed to work at home (distributed on the first day of class)
  - The Xilinx software used in the course can be used in the lab computers and also on your own laptop/PC
    - Connecting to EPFL's VPN may be necessary to use certain features
    - *Please, do not use the class time to set up your laptops! Ask for office hours if you need help*
  - Have your board ready in the lab for each lecture!
  - We can provide locks to use the lockers in the lab room (with a CHF 20.- deposit)
  - Return the board on the last day of class

- Exercises and grading – Continuous evaluation format:
  - Weekly exercises (not graded)
  - Midterm exercise (individual evaluation): **30% of the final grade**
    - Must be delivered at most two weeks after the corresponding class
    - Delivered in GitLab and a report answering questions
    - The TAs will ask you questions to assess your understanding
  - Final guided projects (in groups of two students): **70 % of the final grade**
    - Delivered in GitLab and presented in class **during the last session of the semester**

Delivery of projects: Use of GIT and TCL scripts

- All exercises and projects will be delivered through a git repository
  - Git must be used for daily development, as a way to track your individual work
  - Do not simply push your code at the end!

- During the course, we will learn how to use TCL scripts to reconstruct Xilinx projects
  - To avoid having a vast number of files in the repositories
  - Make smaller and portable projects between computers and members of the team for the team projects part



Questions?

Prof. David Atienza

EPFL – Embedded Systems Laboratory
david.atienza@epfl.ch