

CS-470

Advanced Computer Architecture

Introduction and Practical Info

Paolo Ienne
<paolo.ienne@epfl.ch>

Goals

1. Explore general-purpose computer architecture **beyond undergraduate courses** (CS-200 at EPFL)
2. Analyse particularities of **application-specific computing systems** (e.g., embedded processors and dedicated hardware generated from high-level languages)
3. Give some rudiments of **hardware security**

Understand well **some very fundamental ideas**
that have proved **ubiquitous** in the advances of computing systems

Moodle

- Moodle of this course:

<https://moodle.epfl.ch/course/view.php?id=15017>

- Everything is available from there:
 - Schedule, slides, videos, labs, homework sets, forum, etc.
- Schedule changes will be mentioned in the lectures, posted on Moodle, and announced by e-mail

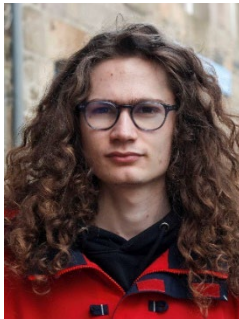
Schedule, Evaluation, TAs, and AEs

Lectures: every week, 3 hours

Labs: ten sessions, 2 hours

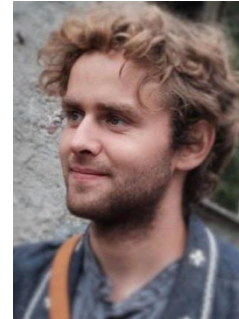
Homework: four sets, 2–3 weeks each to complete

Exam: written, about ten questions



Louis Coulon

<louis.coulon@epfl.ch>



Simon Marti

<simon.marti@epfl.ch>



Mohamed Shahawy

<mohamed.shahawy@epfl.ch>



Julien De Castelnau

<julien.decastelnau@epfl.ch>

CS-470 – Advanced Computer Architecture
SIN, SSC, and SEL MSc
Computer Engineering Specialization
Spring 2025

| Week Date | | | Time | Room | Lectures | Labs | Homework Sets |
|-----------|-----|----------|-----------|---------|---|---|--------------------|
| 1 | Wed | 19.02.25 | | | No course | | |
| 2 | Wed | 26.02.25 | | | No course | | |
| 3 | Wed | 05.03.25 | 13h - 16h | INM200 | Introduction Exploiting ILP Dynamically — Pipelining | | |
| 4 | Wed | 12.03.25 | 13h - 16h | INM200 | Exploiting ILP Dynamically — Dynamic Scheduling | | |
| 5 | Wed | 19.03.25 | 13h - 16h | INM200 | Exploiting ILP Dynamically — Renaming Registers Exploiting ILP Dynamically — Prediction and Speculation | | Set #1 distributed |
| | Thu | 20.03.25 | 11h - 13h | BC07-08 | | Lab #1a: MIPS R10000 | |
| 6 | Wed | 26.03.25 | 13h - 16h | INM200 | Exploiting ILP Dynamically — Prediction and Speculation | | |
| | Thu | 27.03.25 | 11h - 13h | BC07-08 | | Lab #1b: Homework Set #1 support | |
| 7 | Wed | 02.04.25 | 13h - 16h | INM200 | Exploiting ILP Dynamically — Simultaneous Multithreading Exploiting ILP Statically — VLIW Architectures | | |
| | Thu | 03.04.25 | 11h - 13h | BC07-08 | | Lab #1c: Homework Set #1 support | |
| 8 | Wed | 09.04.25 | 13h - 16h | INM200 | Exploiting ILP Statically — VLIW Architectures Exploiting ILP Statically — Compiler Techniques for VLIWs and IA-64 | | Set #2 distributed |
| | Thu | 10.04.25 | 11h - 13h | BC07-08 | | Lab #2a: Intel Itanium | |
| | Tue | 15.04.25 | | | | | Set #1 due |
| 9 | Wed | 16.04.25 | 13h - 16h | INM200 | Exploiting ILP Statically — Compiler Techniques for VLIWs and IA-64 Dynamic Binary Translation | | |
| | Thu | 17.04.25 | 11h - 13h | BC07-08 | | Lab #2b: Homework Set #2 support | |
| 10 | Wed | 30.04.25 | 13h - 16h | INM200 | Application-Specific Computing — Instruction Set Extensions and High Level Synthesis | | |
| | Thu | 01.05.25 | 11h - 13h | BC07-08 | | Lab #2c: Homework Set #2 support | |
| | Tue | 06.05.25 | | | | | Set #2 due |
| 11 | Wed | 07.05.25 | 13h - 16h | INM200 | Application-Specific Computing — Instruction Set Extensions and High Level Synthesis Application-Specific Computing — Challenges of High Level Synthesis | | Set #3 distributed |
| | Thu | 08.05.25 | 11h - 13h | BC07-08 | | Lab #3a: Xilinx VivadoHLS design optimization | |
| 12 | Wed | 14.05.25 | 13h - 16h | INM200 | Hardware Security — Microarchitectural Side-Channel Attacks | | |
| | Thu | 15.05.25 | 11h - 13h | BC07-08 | | Lab #3b: Homework Set #3 support | |
| | Tue | 20.05.25 | | | | | Set #3 due |
| 13 | Wed | 21.05.25 | 13h - 16h | INM200 | Hardware Security — Trusted Execution Environments | | Set #4 distributed |
| | Thu | 22.05.25 | 11h - 13h | BC07-08 | | Lab #4a: Cache Attacks | |
| 14 | Wed | 28.05.25 | 13h - 16h | INM200 | Hardware Security — Physical Side-Channel Attacks | | |
| | Thu | 29.05.25 | 11h - 13h | BC07-08 | | Lab #4b: Homework Set #4 support | |
| | Mon | 09.06.25 | | | | | Set #4 due |

Evaluation

- **70%** of the grade from the written exam
- **30%** from Homework Sets #1–#4
 - Set #1: 10%, performed **individually**
 - Set #2: 10%, can be performed **in groups of two**
 - Sets #3 and #4: 5% each, performed **individually**
 - **Beware**: All sets will be checked for **plagiarism**!
- Labs not graded but highly recommended

Slides

- Available on **Moodle**
- All available now; you will receive notice if there are significant changes
- Please avoid printing in colour (slides should be readable in B/W)—and possibly avoid printing at all!
- Past **videos** are available from Moodle

Labs and Homework Sets

- **Four groups** on four main themes of the course
- **Labs** are usually **introductory to** the corresponding homework, hence optional but **strongly recommended**
- Homework sets will be available on **Moodle** on **scheduled dates**, deadlines are **firm**
- Homework sets are all in the form of **programming assignments** (often with a choice of programming language) but with strict command line interface and I/O format

Grading of Homework Sets

- For sets #1 and #2, you will get a few input files with the corresponding **correct output**
- Distributed I/O pairs are **not meant to be comprehensive** but give a sample of the aspects tested—thorough testing is your responsibility
- Homework sets are **graded semiautomatically** by applying different undisclosed I/O pairs, some qualitatively similar to those distributed and some possibly quite different or larger in scale
- Sets #3 and #4 are **graded manually** by checking qualitative correctness

Textbook(s)

- One **main reference** for most parts
 - Hennessy & Patterson, *Computer Architecture—A Quantitative Approach* (**AQA**), 6th ed., MK, 2019 (or any recent previous versions)
 - Excellent book but not all subjects are fully developed there—on the other hand there is **much** more in it and it is an excellent reference
- Several **papers on Moodle**, mostly “classics”
 - Only a few essential for the labs and homework sets (see Moodle)
 - Many are simply interesting key papers closely related to the course (“optional”)
 - Some may be necessary to understand what is on the slides
- Another **good reference**, very different, with many details
 - Šilc, Robič, & Ungerer, *Processor Architecture*, Springer, 1999
- **Good reader**, interesting complement
 - Hill, Jouppi, & Sohi (eds.), *Readings in Computer Architecture*, MK, 2000

On Your Mark!... Get Set!... Go!

- Any info missing? Ask now...

