

CS-476

**Embedded system design**

Kluter Ties Jan Henderikus

| Cursus                                  | Sem.     | Type |
|---|----------|------|
| Computer science                        | MA2, MA4 | Opt. |
| Cybersecurity                           | MA2, MA4 | Opt. |
| Electrical and Electronical Engineering | MA2, MA4 | Opt. |
| Robotics                                | MA2, MA4 | Opt. |
| SC master EPFL                          | MA2, MA4 | Opt. |

|                            |                     |
|----------------------------|---------------------|
| Language of teaching       | English             |
| Credits                    | 6                   |
| Session                    | Summer              |
| Semester                   | Spring              |
| Exam                       | During the semester |
| Workload                   | 180h                |
| Weeks                      | 14                  |
| <b>Hours</b>               | <b>4 weekly</b>     |
| Courses                    | 2 weekly            |
| Lab                        | 2 weekly            |
| <b>Number of positions</b> |                     |

**Summary**

Hardware-software co-design is a well known concept in embedded system design. It is also a concept required in designing FPGA-accelerators in data-centers. This course teaches how to transform algorithms in smart hardware-software solutions.

**Content**

High-level architectures:

- FIFO's, LIFO's, ring-buffers, and ping-pong buffers.
- FSM-D (finite state machine data-path) structures.
- Stream processing.

Acceleration methods:

- Custom instruction set extensions.
- Hardware accelerators
- Compiler optimizations.

Implementation methods:

- Hardware-software co-design.
- Timing closure.
- Virtual prototyping.
- Bare metal versus usage of an RTOS.

**Learning Prerequisites****Required courses**

- 

**Recommended courses**

- Architecture-aware programming
- CS-200 Computer Architecture

### Important concepts to start the course

- C/C++ programming skills
- Verilog description skills

### Learning Outcomes

By the end of the course, the student must be able to:

- Design buffers to account for different read and write behaviors
- Understand the concept of FSM-D's and use this concept to design accelerators
- Understand the concept of stream processing and be able to implement a stream processor
- Design and optimize an embedded system on FPGA given a set of prerequisites

### Teaching methods

- First 9 weeks : theory with small projects and reports
- Last 5 weeks : mini project in groups of 2 students with final demonstration and presentation

### Expected student activities

- Perform in groups of two students small projects to put the theory into practice
- Optimize a real world problem in a final project, explain the choices made, and present the results

### Assessment methods

- Lab reports : 50%
- Final mini-project : 50%

### Supervision

|               |                             |
|---------------|-----------------------------|
| Office hours  | No                          |
| Assistant.e.s | Yes                         |
| Others        | Electronic forum and Moodle |

### Resources

#### Moodle Link

- <https://go.epfl.ch/CS-476>