UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

COURSE SYLLABUS

Computer Architecture

2526-1-E3102Q102

Aims

At the end of the course the student will have knowledge about the components of a basic computer architecture and about the basics of assembly programming, skills for designing small modifications to the internal structure of a computer and for writing simple assembly programs, and also competence in choosing the best technology, in terms of performance, for some specific computing tasks.

Upon completion of the course, students will be able to:
• understand and explain the main components of the hardware architecture of a computer (e.g., circuits) and their main functions (e.g., the programming toolchain, datapath, memory) (**Knowledge and understanding**);
• understand and develop simple programming languages in Assembly (**Knowledge and understanding**) (**Applying knowledge and understanding**);
• understand and manage exceptions and interruptions (Knowledge and understanding) (**Applying knowledge and understanding**);
• understand and manage memory classes (**Knowledge and understanding**) (**Applying knowledge and understanding**);
• change, motivate, and explain various aspects of the computer architecture (e.g., add new instructions) to implement, for example, new instructions (**Making judgements**) (**Communication skills**).

Contents

- Main components of the hardware architecture of a computer.
- Instruction set architecture.
- Programming toolchain.
- Control of the datapath.
- Exception handling.
- I/O techniques.

- Memory hierarchies: cache

Detailed program

1. Information representation in digital computers

- representation of non numeric information
- representation of positive and negative integer numbers
- fixed and floating point representation of numbers.

3. Logic circuits

- combinatorial circuits
- sequential circuits and FSMs (Finite State Machines)
- overview of relevant circuits: decoder, multiplexer, register file, ALU, etc.

5. Instruction Set Architecture

- von Neumann architecture,
- CPU, registers, ALU and memory,
- fundamental cycle of instruction execution (fetch/decode/execute),
- types and formats of MIPS instructions,
- addressing modes.

7. Assembly language

- Symbolic format of instructions,
- Software development toolchain (compiler, assembler, linker, loader, debugger, etc.),
- Pseudo-instructions and assembler directives,
- Development of simple assembly programs,
- Programming conventions (memory, register names, etc.).

9. Datapath

- Data path for each type of instruction,
- Data path control with FSM (multi-cycle implementation).
- +: pipelining and hazard handling

11. Exception handling

- Taxonomy of exceptions in MIPS32 parlance.
- Modifications to the Control Unit FSM, Cause register, etc.

13. Techniques for handling I/O

- Polling (transfers under program control),
- Interrupt,
- Direct Memory Access.

15. Memory classess : cache

- Direct mapping cache,

- Fully associative cache,
- N-way set associative cache
- +: LRU replacement.

Prerequisites

No prerequisites.

Teaching form

- 14 frontal lessons of 2 hours each held by the teacher in presence;
- 2 frontal lessons of 2 hours each held by the teacher remotely in asynchronous mode;
- 10 sessions for exercises of 2 hours each held by the teacher in presence, 50% frontal 50% interactive;
- 8 interactive laboratory lessons of 3 hours each held by the teacher in presence;

Textbook and teaching resource

- Textbook: David Patterson, John Hennessy: Computer Organization and Design, The Hardware/Software Interface. Fifth edition. Morgan Kaufmann (Elsevier)
- Other teaching material available on the elearning platform concerning lectures, practices, and laboratory, some self-evaluation tests, etc.

Semester

First semester.

Assessment method

The exam will be possible to be completed in two partial tests, to be held around mid-course and at the end of the period, or as always in a single test among those scheduled all aorund the academic year.

Each test is based only on a two-part computer-based assessment, one part blocking and one extended. A valid score is attainable only after the extended part.

The first part is built around questions with closed answers (mainly exercises).

The second part is with closed and open answers and can be taken only by students who passed the first assessment. This assessment revolves also around topics not asked in the first part, marked with '+' in the program in this syllabus.

Office hours

Send email to arrange an appointment.

Sustainable Development Goals