School of Technology and Management
Course Unit Syllabus

Level of qualification:
First cycle (EQF level 6) - Bachelor
Study cycle:
Computer Science

Course Unit: **9119107 – Computer Architecture**

Year 1 **Semester 2** **ISCED Code: 0714** **ECTS: 6.0**

Tipe of Course Unit: Compulsory **Delivery Mode:** Face-to-face **Language of Instruction:** Portuguese

COURSE COORDINATOR: João Carlos da Silva Martins

HOURS OF WORK

TOTAL HOURS	Contact Hours								Hours in autonomous work
	Theory	Theory and practice	Practical and laboratory work	Field work	Seminar	Internship	Tutorial guidance	Other	
150		**30**	**45**						**75**

Prerequisites (if applicable): It is desirable that students have, at least, frequented the Digital Systems course.

LEARNING OUTCOMES (knowledge, skills and competence)

1. Identify the levels of abstraction relevant to the study of microprocessors' architecture.

Main content (MC): 1, 2.

2. Implement computer programs directly in assembly language using procedures.

MC: 3, 4, 5 and 6. Secondary content (SC): 7

3. Compute the result of floating point operations using the IEEE754 representation.

MC: 7 and 8.

4. Compute the result of arithmetic operations using the processor datapath.

MC: 7 and 9. SC: 11.

5. Implement assembly programs that use the floating point unit.

MC: 8 and 9. SC: 10.

6. Describe the processor components and the steps involved in the execution of the different types of instructions: ALU, load/store and branch.

MC: 10. CS: 11.

7. Classify and solve the data and control hazards in a parallel architecture.

MC: 12. CS: 11.

8. Compute the impact of the memory organization in the computer performance.

MC: 13 and 14. SC:15.

CONTENTS

1. Abstractions and technologies in computer architecture
2. Performance measure: execution time, power consumption and number of tasks
3. Instruction set architecture
4. Internal representation of instructions and numeric values
5. Hardware support for arithmetic, data transfer, logic and flux control instructions

6. Hardware support for procedures and addressing modes
7. Computer arithmetic: sum, subtraction, multiply and division
8. Floating point operations (standard IEEE734)
9. Hardware support for floating point operations
10. Processor control and datapath
11. Parallel architectures paradigms
12. Hazards and resolution techniques in parallel architectures
13. Memory types: cache (SRAM), principal (DRAM) and virtual memory
14. Memory systems and hierarchies
15. Storage and input/output devices

DEMONSTRATION OF THE CONTENTS COHERENCE WITH THE COURSE UNIT'S LEARNING OUTCOMES

This course aims to provide the students with the fundamentals needed to understand the organization and architecture of a computational system, in the aspects related to the design and internal functional behavior. Emphasis is given to the knowledge of computer architecture and operation and control mechanisms. In the practical side, exercises are solved aiming a better assimilation of course contents. Several program examples and practical work use assembly language programming.

TEACHING METHODOLOGIES

Syllabus presentation and discussion of particular examples in computer architecture. In addition, the course has also a practical component where laboratories and practical activities will be held . These activities consist in solving exercises, both in an accompanied manner and autonomously, by students. The problems to solve are intended to illustrate the various constituent systems in computer architecture and its impoact on the achitecture performance. Another important practical component will be the development of programming projects in assembly language.

DEMONSTRATION OF THE COHERENCE BETWEEN THE TEACHING METHODOLOGIES AND THE LEARNING OUTCOMES

The theoretical-practical classes aimed at a first to make an introduction to the themes related to computer architecture using, whenever possible, practical examples to show its relevance. The theoretical explanation of each theme is completed with a practical component aimed at strengthening the concepts presented and then extend its practical application to new situations. It will also be provided several programming examples in assembly to be used by students in their programming projects. At the end, the students are intended to be able to understand the operation of any architecture and use the tools necessary to approach a new architecture and develop programs in low-level language for it.

EVALUATION METHODS

Students will be assessed through intermediate evaluation tests and by development and discussion of a programming project in assembly language. There will also be a written final exam, with questions aligned with the practical problems solved in classes, and requirement to write or interpret assembly code segments.

MAIN BIBLIOGRAPHY

• David Patterson, John Hennessy "Computer Organization and Design: the hardware/software interface," 5th edition, Morgan Kaufmann, 2013

• Sarah Harris, David Harris, "Digital Design and Computer Architecture: ARM Edition," Morgan Kaufmann, 2015

• M. Morris Mano, Charles Kime, "Logic and Computer Design Fundamentals," 4 edition, Pearson, 2007

• Herbert Taub, "Digital Circuits and Microprocessors," McGraw-Hill, 1994.

Year of implementation: 2015/2016 | Date of approval by the Technical-Scientific Board: 2016-01-06