COURSE OUTLINE

1. GENERAL

SCHOOL	School of Engineering, University of Patras		
DEPARTMENT	Computer Engineering and Informatics		
LEVEL OF COURSE	Undergraduate, Core		
COURSE CODE	CEID_23Y203	**SEMESTER OF STUDIES**	Fall, 3rd
COURSE TITLE	Computer Architecture		

INDEPENDENT TEACHING ACTIVITIES *if credits are awarded for separate components of the course, e.g. lectures, laboratory exercises, etc. If the credits are awarded for thewhole of the course, give the weekly teaching hours and the total credits*	**TEACHING HOURS PER WEEK**	**ECTS CREDITS**
Lectures	4	4
Total	**4**	**4**
Add rows if necessary. The teaching organization and methods used are described in detail in (d).		

COURSE TYPE *general background, special background, specialized general knowledge, skills development*	General Background Specialized General Knowledge
PREREQUISITE COURSES:	Recommended prerequisite knowledge: • Fundamental Principles of Computer System Organization and Operation (CEID_22Y104) • Logic Design (CEID_23Y107)
TEACHING AND ASSESSMENT LANGUAGE:	Greek
THE COURSE IS OFFERED TO ERASMUS STUDENTS	No
COURSE WEBPAGE (URL)	https://eclass.upatras.gr/courses/CEID1492/

2. LEARNING OUTCOMES

Learning outcomes

The course learning outcomes, specific knowledge, skills, and competences of an appropriate level that the students will acquire with the successful completion of the course are described.

Consult Appendix A

- *Description of the level of learning outcomes for each qualifications cycle, according to the Qualifications Framework of the European Higher Education Area*
- *Descriptors for Levels 6, 7 & 8 of the European Qualifications Framework for Lifelong Learning and Appendix B*
- *Guidelines for writing Learning Outcomes*

Upon successful completion of the course, students will be able to:

1. Identify and describe the basic organization of a computer system and its main structural components (processor, memory, input/output).
2. Analyze and compare the performance of computing systems using appropriate metrics.
3. Develop assembly language programs for the RISC-V architecture, demonstrating an understanding of low-level hardware operations.
4. Use and exploit computer architecture simulators (such as RARS) to understand and monitor program execution and processor behavior.

5. Design and describe the operation of a simple central processing unit (CPU) using a single-cycle datapath architecture, with emphasis on the datapath and control unit.
6. Explain the operation and benefits of pipelining, as well as the main techniques for handling hazards.
7. Understand the fundamental principles of memory hierarchy, including cache memories and their impact on system performance.

General Abilities

Taking into consideration the general competences that the degree-holder must acquire (as these appear in the Diploma Supplement and appear below), at which of the following does the course aim?

Search for, analysis and synthesis of data and information, with the use of the necessary technology	*Project planning and management*
Adapting to new situations	*Respect for difference and multiculturalism*
Decision-making	*Respect for the natural environment*
Working independently	*Showing social, professional and ethical responsibility and sensitivity to gender issues*
Team work	*Criticism and self-criticism*
Working in an international environment	*Production of free, creative and inductive thinking*
Working in an interdisciplinary environment	
Production of new research ideas	*Others…*

Decision-making
Working independently
Team work
Production of new research ideas
Production of free, creative and inductive thinking

3. COURSE CONTENT

The student is first introduced to the fundamental concepts of computer organization and technology, along with the contemporary challenges faced by computer architecture. The course then focuses on the concept of Instruction Set Architecture (ISA) as a critical interface between hardware and software. Instruction is based on the RISC-V architecture, a modern, open-source, and modular Reduced Instruction Set Computer (RISC) architecture, which serves as a powerful educational tool for understanding the foundational principles of computer systems. Within this context, students develop basic programming skills in RISC-V assembly language and learn how high-level compiled code is translated into machine instructions. This is followed by an in-depth exploration of performance evaluation metrics and methods, as well as an analysis of the key factors affecting system efficiency.

Subsequently, the course focuses on computer organization and processor design for implementing an instruction set architecture – initially without pipelining techniques. Students study and become familiar with the datapath, the control unit, and their relationship to RISC-V instructions, drawing on knowledge from the courses “Logic Design” and “Fundamental Principles of Computer System Organization and Operation”. The course concludes with an introduction to pipelining, emphasizing both its performance benefits and the complications it introduces.

Course Topics:

- Introduction to computer architecture and technology
- Performance evaluation metrics
- RISC-V Instruction Set Architecture
- Assembly language and machine code
- From high-level code to hardware execution
- Computer arithmetic: representations, operations, and implementation
- Single-cycle processor design: datapath and control unit
- Introduction to pipelining and its impact on performance
- Introduction to the design and operation of cache memories

4. TEACHING AND LEARNING METHODS – ASSESSMENT

TEACHING METHOD *Face-to-face, Distance learning, etc.*	Face-to-face
USE OF INFORMATION AND COMMUNICATION TECHNOLOGIES *Use of ICT in teaching, laboratory education, communication with students*	Powerpoint Slides eClass
TEACHING ORGANIZATION *The manner and methods of teaching are described in detail.* *Lectures, seminars, laboratory practice, fieldwork, study and analysis of bibliography, tutorials, placements, clinical practice, art workshop, interactive teaching, educational visits, project, essay writing, artistic creativity, etc.* *The student's study hours for each learning activity are given, as well as the hours of non-directed study according to the principles of the ECTS*	***Activity*** / ***Semester Workload*** Lectures: 52 Study/Homeworks: 42 Midterms: 3 Final Exams: 3 ***Total number of hours for the Course (25 hours of work-load per ECTS credit)***: ***100***
STUDENT ASSESSMENT *Description of the evaluation procedure* *Language of evaluation, methods of evaluation, summative or conclusive, multiple-choice questionnaires, short-answer questions, open-ended questions, problem-solving, written work, essay/report, oral examination, public presentation, laboratory work, clinical examination of patient, art interpretation, other* *Specifically defined evaluation criteria are given and if and where they are accessible to students.*	Assessment is conducted in Greek and includes assignments, a midterm progress assessment, and a final written examination. The evaluation of theoretical understanding is primarily based on the final exam, which consists of a combination of multiple-choice questions, short-answer questions, essay-type questions, and problem-solving exercises. After the publication of results, students have the opportunity to review their graded exam, understand their mistakes, and discuss the grading criteria.

5. RECOMMENDED LITERATURE

Suggested bibliography:

- "Computer Organization and Design, RISC-V Edition", D. A. Patterson, J. L. Hennessy, 2nd Edition, Morgan Kaufmann, 2020.
- "Digital Design and Computer Architecture, RISC-V Edition", S. L. Harris, D. Harris, Morgan Kaufmann, 2021.
- "Computer Organization and Design, MIPS Edition", D. A. Patterson, J. L. Hennessy, 6th Edition, Morgan Kaufmann, 2020.
- " Computer Organization and Architecture", 11th edition, William Stallings, Published by Pearson, 2019.

Related academic journals:

- IEEE Micro
- IEEE Transactions on Computers
- IEEE Transactions on Dependable and Secure Computing
- ACM Transactions on Architecture and Code Optimization