COMPUTER ARCHITECTURE

COURSE OUTLINE

Responsible: Nikolaos Petrakis

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	Department of Electronic Engineering		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	0806.4.001.0	**SEMESTER**	1st
COURSE TITLE	Computer Architecture		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
	2	5
Total	**2**	**5**

COURSE TYPE general background, special background, specialised general knowledge, skills development	
PREREQUISITE COURSES	Highly recommended to have enough knowledge of "Structured Programming" and "Digital Systems Design" .
LANGUAGE OF INSTRUCTION and EXAMINATIONS	English
OFFERED TO ERASMUS STUDENTS	Yes (in English) — Winter Semester
COURSE WEBSITE (URL)	https://eclass.chania.teicrete.gr/courses/

2. LEARNING OUTCOMES

Learning outcomes

Familiarity with the internal structure and basic operations of a computer as well as gaining knowledge in the organization and design of the hardware and software that make up a typical computing system. Emphasis will be placed on the lower levels, the level of digital logic and the design of the central processing unit.

Programming in machine language and in symbolic language (assembly).

Understand processor organization, memory, datapath and input / output structures.

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

• Explain the purpose of CPU, I / O subsystems, and various storage subsystems.

• Understand the Instruction Set Architecture (ISA) of a machine, its design and implementation.

• Distinguish computers based on their set of instructions.

• Describe the modern methodology for evaluating and comparing processor performance.

• Describe how to internally represent integer and real (floating point) numbers (IEEE 754) and perform conversions according to the standard.

• Describe the basic ways of addressing and give examples of instructions that use them.

• Describe the technique of partially overlapping operations and its expected benefits.

• Know the low level programming rules and execute code including defining and calling procedures, leaf-

procedures, but also non-leaf procedures using the stack correctly.

• Understand the relationship between hardware and software and the relationship between low-level programming and high-level programming.

• Understand the implementation of the control unit either as a classical sequential circuit or with the technique of microprogramming.

• To know the basic principles that govern the organization of modern processors, and some modern research trends in the field of computer architecture.

• Use the MIPS emulator of the MIPS processor for programming at the machine language level.

General Competences

Search, analysis and synthesis of data and information, using the necessary technologies

Decision making

Autonomous work

Teamwork

Project design and management

Exercise criticism and self-criticism

Promoting free, creative, and inductive thinking

3. SYLLABUS

Compulsory course for students in the field of computer organization and computer architecture.

Reference to historical data on the evolution of computers and categories of computer systems.

RISCs and CISCs.

The internal structure of a modern thirty-two-bit processor (MIPS32) is gradually revealed through the study of its instruction set. Also, reference is made to issues of design of computer systems with parallel processing (MIMD, SIMD).

Categories of computer applications and their characteristics.

Structure and basic operations of a typical computer. Study of the instruction repertoire.

Machine language - representation of instructions on the computer.

Symbolic language (assembly language). High level programming language support.

Hardware support for procedures (leaf procedures and non-leaf procedures).

Addressing modes. Signed / unsigned integer representation.

Arithmetic and logic unit and arithmetic and logic operations.

Representation of real (floating point) numbers (IEEE 754) and operations with them.

Computer evaluation and understanding of performance.

Address and data paths and datapath design.

Control unit and timings. Microprogram development.

Increase efficiency by pipelining.

Main memory. Auxiliary memory. Cache memory. Virtual Memory. Memory technology.

Content Addressable Memories (CAM).

Input / Output Units.

Using the various tools (SPIM or MARS) introduced in the course, students should explore in depth several aspects of computer architecture and / or organization to achieve a more complete understanding.

4. TEACHING and LEARNING METHODS - EVALUATION	
DELIVERY Face-to-face, Distance learning, etc.	Face to face theoretical teaching. Laboratory training in groups of students (maximum 20 students per group). Practice exercises in small groups of students.
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	Use of slide show software Use of an Integrated Development Environment (IDE) like MARS 4.5, which is a very easy-to-use MIPS assembler, developed at the University of Missouri. Communication with students through an asynchronous distance learning platform.
TEACHING METHODS The manner and methods of teaching are described in detail.	**Activity** | **Semester workload** **Course total** |
STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	I. Written final exam (WFE) (70%) - Problem solving / calculations - Comparative evaluation of theory elements II. Laboratory test (LT) (15%) - Laboratory work / technical reports / measurements in small groups III. Examination in practice exercises (PE) (15%) - Individual practice tasks The grade of the course (WFE * 0.7 + LT * 0.15 + PE * 0.15) must be at least five (5.0). The grade of each of I, II, III must be at least three (3.0). The assessment criteria are accessible to students from the course website and are announced in the first course.

5. ATTACHED BIBLIOGRAPHY

- Suggested textbooks:

- D. A. Patterson, J. L. Hennessy, *Computer Organization and Design – The Hardware / Software Interface*, 5th ed., Morgan Kaufman Publishers, 2014.
- J. L. Hennessy, D. A. Patterson, *Computer Architecture: A Quantitative Approach*, 6th ed., Morgan Kaufman Publishers, 2018.