Course Title	Computer Organization and Architecture				
Course Code	ACOE201				
Course Type	Compulsory				
Level	Bachelor (1st Cycle)				
Year / Semester	3 (Fall)				
Teacher's Name	Costas Kyriacou, Konstantinos Tatas				
ECTS	5	Lectures / week	3	Laboratories / week	2
Course Purpose	This course aims to introduce students to computer organization and architecture with emphasis on the instruction set architecture, the CPU datapath units, the operation of a single cycle and a multi-cycle CPU, the semiconductor memory technologies, the memory hierarchy and the Input/Output system.				
Learning Outcomes	By the end of the course, the students are expected to: 1. describe the instruction execution cycle with reference to the flow of information at the register level, and analyze typical Instruction Set Architectures with respect to the machine code size, number of operands, addressing modes and branch types; 2. design the basic units of a CPU datapath, such as the ALU and the register file; 3. describe the internal structure and analyze the operation of a CPU datapath and design a simple single-cycle and a multi-cycle non-pipelined CPU; 4. Describe the internal structure of the types of semiconductor memory devices, evaluate them with respect to memory capacity, speed and power consumption, and design simple memory modules with word size and address size expansion; 5. explain how the memory wall problem affects the performance of a computer and how cache memory exploits locality to reduce the memory wall problem; 6. describe the operation and evaluate the performance of the common cache memory mapping methods, cache replacement policies and write policies; 7. justify the need for virtual memory and outline the function of virtual memory mechanisms; 8. outline and compare the mechanisms for I/O communication and data transfers;				

	9. write assembly language code segments and use computer tools to debug them and analyze their operation; 10. use EDA tools and FPGA boards to design, simulate, verify, implement and test the operation of datapath units and memory devices.		
Prerequisites	ACOE161, ASCS182	Co-requisites	None
Course Content	• **Computer Arithmetic:** Negative number representation, signed and unsigned arithmetic operations on binary and hexadecimal numbers, fractional numbers and the floating point representation. • **Introduction and Instruction Set Architectures:** Instruction cycle and flow of information at the register level. Performance issues. Instruction Set Architectures, instruction formats and instruction decoding. Relation between machine language, assembly language and high level languages. • **CPU design basics:** Datapaths, register files, ALU, shift and rotate circuits. Register transfer operations and micro-operations. Control unit implementation. Single-cycle and multi-cycle non-pipelined CPU design. • **Semiconductor Memory:** Internal structure of semiconductor memory devices, signals and basic characteristics. Types of memory devices, ROM and RAM (dynamic and static). Memory expansion and memory addressing. Memory technologies. • **Memory Hierarchy:** The memory wall problem and the locality principle. Cache memory organization and mapping. Cache replacement and write policies. Cache performance metrics. Virtual memory. • **Input/Output:** I/O interfacing and addressing. Mechanisms for I/O communication and data transfers: program controlled, direct memory access and interrupts. • **Laboratory Work:** Part 1: Assembly language programming, use of assemblers and simulators to analyze the operation of code segments. Part 2: Individual or small group experiments performed with the use of common FPGA boards. Experiments include the design and analysis of the basic units of a typical CPU such as register files, ALUs, memory devices and simple cache units.		
Teaching Methodology	The taught part of course is delivered to the students by means of lectures, conducted with the help of computer presentations. Lecture notes and presentations are available for students to use in combination with the textbook, through the university's e-learning platform. Lectures are supplemented with laboratory work. The laboratory work is made out of two parts. The first part introduces students to assembly language, where lab sessions are carried out using simulators. The second part concerns the operation of specific hardware components examined with the use of FPGA boards.		

Bibliography	Textbook: • Paterson, Hennessy, ***Computer Organization and Design: the Hardware/Software Interface***, Morgan Kaufman, 2017 Reference: • M. Mano, C. R. Kime, T. Martin, ***Logic and Computer Design Fundamentals***, Prentice Hall, 2014
Assessment	The assessment of the course includes two tests with problem solving questions, two assignments with problem solving and design questions and a final exam with problem solving and design open questions. The laboratory work assessment is based on the students' lab reports. The weights for each assessment component are: • Assignments 10% • Tests: 30% • Laboratory Work: 20% • Final Exam 40%
Language	English