Course Title	Advanced Computer Architecture				
Course Code	ACOE301				
Course Type	Compulsory				
Level	BSc (Level 1)				
Year / Semester	3rd Year/ 5th Semester				
Teacher's Name	Dr. Costas Kyriacou, Dr. Konstantinos Tatas				
ECTS	6	Lectures/week	3	Laboratories/week	1
Course Purpose	This course builds on the knowledge acquired through the course ACOE201 (Computer Organization and Architecture). Its purpose is to provide students with an in-depth knowledge on advanced concepts in computer architecture such as pipelining, superscalar processors and memory hierarchy (caches and virtual memory organization).				
Learning outcomes	1. Evaluate and compare the performance of computer systems through the use of common metrics and benchmarks. 2. Analyse the ISA of the main processors with respect to performance and complexity. 3. Describe the operation of integer and floating arithmetic circuits used in RISC processors. 4. Describe the operation of a pipelined microarchitecture and justify the need of optimizations like forwarding, loop unrolling, and branch prediction. 5. Describe the operation of superscalar microarchitectures and justify the need of optimizations like out-of-order execution, register renaming and speculative execution. 6. Justify the need for memory hierarchy, describe the features of current cache systems and virtual memory.				
Prerequisites:	ACOE201	Co-requisites:	None		
Course contents:	• **Performance Metrics:** Measuring performance and metrics. Improve performance, clock cycles, CPI, instructions count, MIPS, MOPS, MFLOPs. Benchmarks. Amdahl's Law. • **Instruction Set Architecture (ISA):** Specifications, classes, registers, memory addressing and addressing modes. In-depth analysis of the MIPS and the x86 ISAs. Reference to the ARM and RISC V ISAs. • **ALU Design:** Full ALU design of the MIPS processor. Multiplication and division algorithms in hardware. Floating point units. • **Pipelining:** Single-cycle, Multi-cycle versus Pipeline. Structural, data and control hazards. Staling and forwarding. Loop unrolling, branch prediction and speculative execution. Exceptions. The MIPS R3000 pipeline and design of a pipelined processor.				

	• **Superscalar processors:** Superscalar pipelines. Data hazards in superscalar pipelines. Out-of-order execution, register renaming. Multithreading and execution of threads in superscalar processors. • **Memory Hierarchy:** Locality and memory hierarchy. Advanced cache memory issues: set associative caches, multilevel caches, write policies, replacement policies, Virtual memory. Protection. Translation Lookaside Buffer (TLB). • **Assembly Language:** Laboratory sessions to enhance the knowledge of students in programming with assembly language.
Teaching Methodology	The course is structured in lectures that are conducted with the help of both computer presentations and traditional means. Practical examples and exercises are included in the lectures to enhance the material learning process. Often short post-lecture quizzes are used to assess the level of student understanding and provide feedback. Student questions are addressed during the lecture, or privately after the lecture or during office hours. Lecture notes are available through the web for students to use in combination with the textbooks.
Bibliography	Textbooks: • J. Hennessy, D. Paterson, ***Computer Architecture: A Quantitative Approach***, 6th edition, Morgan Kaufman, 2017 • D. Paterson, J. Hennessy, ***Computer Organization and Design: the Hardware/Software Interface***, Morgan Kaufman, 2015
Assessment	Students are assessed continuously and their knowledge is checked through tests with their assessment weight, date and time being set at the beginning of the semester via the course outline. Furthermore, individual and group design assignments are used to develop practical engineering skills. • Assignments 20% • Tests: 20% • Labs 20% • Final Exam: 40%
Language	English