<table>
<tr><td>Course Title</td><td colspan="5">Computer Organization & Architecture</td></tr>
<tr><td>Course Code</td><td colspan="5">ECE210</td></tr>
<tr><td>Course Type</td><td colspan="5">Compulsory</td></tr>
<tr><td>Level</td><td colspan="5">Bachelor (1st Cycle)</td></tr>
<tr><td>Year / Semester</td><td colspan="5">2nd Year / 3rd Semester</td></tr>
<tr><td>Teacher’s Name</td><td colspan="5">TBA</td></tr>
<tr><td>ECTS</td><td>6</td><td>Lectures / week</td><td>3 hours / 14 weeks</td><td>Laboratories / week</td><td>N/A</td></tr>
<tr><td>Course Purpose and Objectives</td><td colspan="5">The objective of this course is to provide a comprehensive introduction to the fundamental concepts of computer organization and computer architecture, and to equip students with the necessary skills which will allow them to understand the operation of a modern Central Processing Unit (CPU) and measure its performance. Additionally, the course will allow students to understand the trade-offs during the design of a modern computer system. The course follows an embedded laboratory approach, where students are required to utilize CPU simulation tools during the implementation of lectures.</td></tr>
<tr><td>Learning Outcomes</td><td colspan="5">Upon successful completion of this course, students should be able to:
• Describe the concepts of computer organization and computer architecture and be able to differentiate between them
• Describe the Von Neumann architecture and explain the operation of the fetch-decode-execute cycle
• Explain how hardware parts interact in a typical CPU in order to execute a computer instruction
• Describe the major characteristics of Instruction Set Architectures (ISA), and utilize effectively a typical ISA in order to program the operation of a CPU
• Explain how computer memory is organized hierarchically in a modern computer system, and illustrate the implementation methods of cache memory and virtual memory
• Describe and explain the operation of the basic types of I/O architectures used in modern computer systems</td></tr>
<tr><td>Prerequisites</td><td colspan="2">ECE200</td><td colspan="2">Co-requisites</td><td>None</td></tr>
<tr><td>Course Content</td><td colspan="5"><u>Overview and History:</u> Basic concepts of computer architecture and computer organization, differentiation between the concepts of computer architecture and computer organization, main developments and contributors in the history of computer systems.</td></tr>
</table>

	The Computer System: Definition of a computer system, the Central Processing Unit (CPU), main parts of a CPU (Arithmetic Logic Unit, Control Unit, Registers), System Bus, Clock, Main Memory, description and explanation of the Von Neumann architecture as the basis for the implementation of computer systems, limitations of the Von Neumann architecture, the fetch-decode-execute cycle, the Von Neumann bottleneck, representation of a computer as a hierarchical set of computing levels, recent developments in the implementation of computer systems (including Harvard architectures, parallel computers, multicore architectures, embedded systems). Computer Organization: Arithmetic Logic Unit implementation, Control Unit implementation, hardware implementation of the Control Unit's decoding circuit, software (microprogrammed) implementation of the Control Unit's decoding circuit, trade-offs between hardware and microprogrammed implementation of the Control Unit's decoding circuit. Instruction Set Architectures (ISAs): Basic characteristics and functions of ISAs, design parameters of instructions (including instruction length, variable vs fixed-length instructions, addressing modes, number of operands, and endianness), internal storage (including accumulator, stack, and register-based architectures), types of instructions, Complex Instruction Set Architectures (CISC), Reduced Instruction Set Architectures (RISC), critical comparison between CISC and RISC architectures, modern implementations of CISC and RISC architectures. Assemblers and Assembly Language: Definition of assemblers, operation of assemblers, assembly language, source files and object files, labels, comments, mnemonics, assembly languages versus high-level programming languages, developing small-scale assembly programs for specific CPUs. Memory Organization: Basic memory concepts, hierarchical organisation of computer memory, access time, cost, and capacity trade-offs, types of computer memory, Random Access Memory (RAM), Read-Only Memory (ROM), cache memory, cache mapping schemes, cache policies, virtual memory, implementation of virtual memory using paging and segmentation techniques, secondary storage systems (including magnetic, optical, and solid-state systems). Input / Output (I/O): Definition and design considerations of the I/O subsystem in a modern computer system, I/O control methods (including programmed I/O, interrupt-driven I/O, Direct Memory Access (DMA) and channel-attached I/O), basic interrupt service routines, operation of the I/O bus
Teaching Methodology	Face-to-Face

Bibliography	“*The Essentials of Computer Organization and Architecture*”, by Linda Null and Julia Lobur “*Computer Organization and Architecture*” by William Stallings “*Structured Computer Organization*” by Andrew Tanenbaum “*Computer Architecture: A Quantitative Approach*” by John L. Hennessy “*Digital Design and Computer Architecture*” by David Harris and Sarah Harris
Assessment	Examinations: 70% Assignments/Lab: 20% Class Participation and Attendance: 10% 100%
Language	English