University of Engineering and Technology
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS2201. Computer Architecture
2. **Credits:** 3
3. **Hours of theory and Lab:** 2 HT; 2 HL;
4. **Professor(s)**

   Meetings after coordination with the professor
5. **Bibliography**

[Den05] Peter J. Denning. "The locality principle". In: *Commun. ACM* 48.7 (July 2005), pp. 19–24. ISSN: 0001-0782. DOI: 10.1145/1070838.1070856. URL: http://doi.acm.org/10.1145/1070838.1070856.

[Don06] J. Dongarra. "Trends in high performance computing: a historical overview and examination of future developments". In: *Circuits and Devices Magazine, IEEE* 22.1 (2006), pp. 22–27. ISSN: 8755-3996. DOI: 10.1109/MCD.2006.1598076.

[EA05] Hesham El-Rewini and Mostafa Abd-El-Barr. *Advanced Computer Architecture and Parallel Processing*. Hoboken, NJ: John Wiley & Sons, 2005. ISBN: 0-471-46740-5.

[HP06] J. L. Hennessy and D. A. Patterson. *Computer Architecture: A Quantitative Approach*. 4th. San Mateo, CA: Morgan Kaufman, 2006.

[Joh91] M. Johnson. *Superscalar microprocessor design*. Prentice Hall series in innovative technology. Prentice Hall, 1991. ISBN: 9780138756345.

[Par02] Behrooz Parhami. *Introduction to parallel processing: algorithms and architectures*. Plenum series in computer science. Plenum Press, 2002. ISBN: 9780306459702.

[Par05] Behrooz Parhami. *Computer Architecture: From Microprocessors to Supercomputers*. New York: Oxford Univ. Press, 2005. ISBN: ISBN 0-19-515455-X.

[PH04] D. A. Patterson and J. L. Hennessy. *Computer Organization and Design: The Hardware/Software Interface*. 3rd ed. San Mateo, CA: Morgan Kaufman, 2004.

[Sta10] William Stalings. *Computer Organization and Architecture: Designing for Performance*. 8th. Upper Saddle River, NJ: Prentice Hall, 2010.

6. **Information about the course**

   (a) **Brief description about the course** It is necessary that the professional in Computer Science has a solid knowledge of the organization and operation of the various computer systems in which the programming environment is installed. This will also know how to establish the scope and limits of the applications that are developed according to the platform being used.

   The following topics will be addressed: basic digital logic components in a computer system, design of instruction sets, microarchitecture of the processor and execution in pipelining, organization of memory: cache and virtual memory, protection and sharing, system I / O and interrupts, super-scalar architectures and out-of-order execution, vector computers, multithreading architectures, symmetric multiprocessors, memory and synchronization models, integrated systems and parallel computers.

   (b) **Prerrequisites:** CS1D02. Discrete Structures II. (2^{nd} Sem)

   (c) **Type of Course:** Mandatory

   (d) **Modality:** Face to face

7. Specific goals of the Course

- This course is intended to provide the student with a solid foundation in the evolution of computer architectures and the factors that influenced the design of hardware and software in today's computer systems.
- Ensure understanding of what hardware is itself and how it interacts with hardware and software in a current computing system.
- To deal with the following topics: basic digital logic components in a computer system, design of instruction sets, microarchitecture of the processor and execution in pipelining, organization of memory: cache and virtual memory, protection and sharing, system I / O and interrupts, super-scalar architectures and out-of-order execution, vector computers, multithreading architectures, symmetric multiprocessors, memory and synchronization models, integrated systems and parallel computers.

8. Contribution to Outcomes

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

g) The broad education necessary to understand the impact of computing solutions in a global, economic, environmental, and societal context. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Assessment**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

g) The broad education necessary to understand the impact of computing solutions in a global, economic, environmental, and societal context. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Assessment**)

9. Competences (IEEE)

C4. An understanding of computer hardware from a software perspective, for example, use of the processor, memory, disk drives, display, etc.⇒ **Outcome i**

C8. Understanding of what current technologies can and cannot accomplish. ⇒ **Outcome b,i,g**

C9. Understanding of computing's limitations, including the difference between what computing is inherently incapable of doing vs. what may be accomplished via future science and technology.⇒ **Outcome b,g**

C4. An understanding of computer hardware from a software perspective, for example, use of the processor, memory, disk drives, display, etc.⇒ **Outcome i**

C8. Understanding of what current technologies can and cannot accomplish. ⇒ **Outcome b,i,g**

C9. Understanding of computing's limitations, including the difference between what computing is inherently incapable of doing vs. what may be accomplished via future science and technology.⇒ **Outcome b,g**

10. List of topics

1. Digital logic and digital systems
2. Machine level representation of data
3. Assembly level machine organization
4. Functional organization
5. Memory system organization and architecture
6. Interfacing and communication
7. Multiprocessing and alternative architectures
8. Performance enhancements

11. Methodology and Evaluation

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:
The final grade NF is obtained through of:

$NF = 0.10 * R + 0.30 * P + 0.20 * L + 0.20 * EP + 0.20 * EF$

Where:

R : Lecturas

P : Proyecto

L : Laboratorio

EP : Examen Parcial

EF : Examen Final

To pass the course you must obtain 11 or more in the final grade NF.

12. Content

Unit 1: Digital logic and digital systems (18)	
Competences Expected: C8	
Learning Outcomes	**Topics**
• Describe the progression of computer technology components from vacuum tubes to VLSI, from mainframe computer architectures to the organization of warehouse-scale computers [Familiarity] • Comprehend the trend of modern computer architectures towards multi-core and that parallelism is inherent in all hardware systems [Usage] • Explain the implications of the "power wall" in terms of further processor performance improvements and the drive towards harnessing parallelism [Usage] • Articulate that there are many equivalent representations of computer functionality, including logical expressions and gates, and be able to use mathematical expressions to describe the functions of simple combinational and sequential circuits [Familiarity] • Design the basic building blocks of a computer: arithmetic-logic unit (gate-level), registers (gate-level), central processing unit (register transfer-level), memory (register transfer-level) [Usage] • Use CAD tools for capture, synthesis, and simulation to evaluate simple building blocks (eg, arithmetic-logic unit, registers, movement between registers) of a simple computer design [Familiarity] • Evaluate the functional and timing diagram behavior of a simple processor implemented at the logic circuit level [Assessment]	• Overview and history of computer architecture • Combinational vs. sequential logic/Field programmable gate arrays as a fundamental combinational + sequential logic building block • Multiple representations/layers of interpretation (hardware is just another layer) • Computer-aided design tools that process hardware and architectural representations • Register transfer notation/Hardware Description Language (Verilog/VHDL) • Physical constraints (gate delays, fan-in, fan-out, energy/power)
Readings : [Par05], [PH04]	

Unit 2: Machine level representation of data (8)	
Competences Expected: C9	
Learning Outcomes	**Topics**
• Explain why everything is data, including instructions, in computers [Assessment] • Explain the reasons for using alternative formats to represent numerical data [Familiarity] • Describe how negative integers are stored in sign-magnitude and twos-complement representations [Usage] • Explain how fixed-length number representations affect accuracy and precision [Usage] • Describe the internal representation of non-numeric data, such as characters, strings, records, and arrays [Usage] • Convert numerical data from one format to another [Usage]	• Bits, bytes, and words • Numeric data representation and number bases • Fixed- and floating-point systems • Signed and twos-complement representations • Representation of non-numeric data (character codes, graphical data) • Representation of records and arrays
Readings : [Par05], [Sta10]	

Unit 3: Assembly level machine organization (8)	
Competences Expected: C4,CS3	
Learning Outcomes	**Topics**
• Explain the organization of the classical von Neumann machine and its major functional units [Familiarity] • Describe how an instruction is executed in a classical von Neumann machine, with extensions for threads, multiprocessor synchronization, and SIMD execution [Familiarity] • Describe instruction level parallelism and hazards, and how they are managed in typical processor pipelines [Familiarity] • Summarize how instructions are represented at both the machine level and in the context of a symbolic assembler [Familiarity] • Demonstrate how to map between high-level language patterns into assembly/machine language notations [Usage] • Explain different instruction formats, such as addresses per instruction and variable length vs fixed length formats [Usage] • Explain how subroutine calls are handled at the assembly level [Usage] • Explain the basic concepts of interrupts and I/O operations [Familiarity] • Write simple assembly language program segments [Usage] • Show how fundamental high-level programming constructs are implemented at the machine-language level [Usage]	• Basic organization of the von Neumann machine • Control unit; instruction fetch, decode, and execution • Instruction sets and types (data manipulation, control, I/O) • Assembly/machine language programming • Instruction formats • Addressing modes • Subroutine call and return mechanisms • I/O and interrupts • Heap vs. Static vs. Stack vs. Code segments
Readings : [Par05], [PH04], [HP06]	

Unit 4: Functional organization (8)	
Competences Expected: C9	
Learning Outcomes	**Topics**
• Compare alternative implementation of datapaths [Assessment] • Discuss the concept of control points and the generation of control signals using hardwired or microprogrammed implementations [Familiarity] • Explain basic instruction level parallelism using pipelining and the major hazards that may occur [Usage] • Design and implement a complete processor, including datapath and control [Usage] • Determine, for a given processor and memory system implementation, the average cycles per instruction [Assessment]	• Implementation of simple datapaths, including instruction pipelining, hazard detection and resolution • Control unit: hardwired realization vs. microprogrammed realization • Instruction pipelining • Introduction to instruction-level parallelism (ILP)
Readings : [Par05], [HP06]	

Unit 5: Memory system organization and architecture (8)	
Competences Expected: CS3	
Learning Outcomes	**Topics**
• Identify the main types of memory technology (eg, SRAM, DRAM, Flash, magnetic disk) and their relative cost and performance [Familiarity] • Explain the effect of memory latency on running time [Familiarity] • Describe how the use of memory hierarchy (cache, virtual memory) is used to reduce the effective memory latency [Usage] • Describe the principles of memory management [Usage] • Explain the workings of a system with virtual memory management [Usage] • Compute Average Memory Access Time under a variety of cache and memory configurations and mixes of instruction and data references [Assessment]	• Storage systems and their technology • Memory hierarchy: importance of temporal and spatial locality • Main memory organization and operations • Latency, cycle time, bandwidth, and interleaving • Cache memories (address mapping, block size, replacement and store policy) • Multiprocessor cache consistency/Using the memory system for inter-core synchronization/atomic memory operations • Virtual memory (page table, TLB) • Fault handling and reliability • Error coding, data compression, and data integrity
Readings : [Par05], [PH04], [Den05]	

Unit 6: Interfacing and communication (8)	
Competences Expected: C4,C9,CS3	
Learning Outcomes	**Topics**
• Explain how interrupts are used to implement I/O control and data transfers [Familiarity] • Identify various types of buses in a computer system [Familiarity] • Describe data access from a magnetic disk drive [Usage] • Compare common network organizations, such as ethernet/bus, ring, switched vs routed [Assessment] • Identify the cross-layer interfaces needed for multimedia access and presentation, from image fetch from remote storage, through transport over a communications network, to staging into local memory, and final presentation to a graphical display [Familiarity] • Describe the advantages and limitations of RAID architectures [Familiarity]	• I/O fundamentals: handshaking, buffering, programmed I/O, interrupt-driven I/O • Interrupt structures: vectored and prioritized, interrupt acknowledgment • External storage, physical organization, and drives • Buses: bus protocols, arbitration, direct-memory access (DMA) • Introduction to networks: communications networks as another layer of remote access • Multimedia support • RAID architectures
Readings : [Par05], [Sta10]	

Unit 7: Multiprocessing and alternative architectures (8)	
Competences Expected: C9	
Learning Outcomes	**Topics**
• Discuss the concept of parallel processing beyond the classical von Neumann model [Assessment] • Describe alternative parallel architectures such as SIMD and MIMD [Familiarity] • Explain the concept of interconnection networks and characterize different approaches [Usage] • Discuss the special concerns that multiprocessing systems present with respect to memory management and describe how these are addressed [Familiarity] • Describe the differences between memory backplane, processor memory interconnect, and remote memory via networks, their implications for access latency and impact on program performance [Assessment]	• Power Law • Example SIMD and MIMD instruction sets and architectures • Interconnection networks (hypercube, shuffle-exchange, mesh, crossbar) • Shared multiprocessor memory systems and memory consistency • Multiprocessor cache coherence
Readings : [Par05], [Par02], [EA05]	

Unit 8: Performance enhancements (8)	
Competences Expected: C8,C9	
Learning Outcomes	**Topics**
• Describe superscalar architectures and their advantages [Familiarity] • Explain the concept of branch prediction and its utility [Usage] • Characterize the costs and benefits of prefetching [Assessment] • Explain speculative execution and identify the conditions that justify it [Assessment] • Discuss the performance advantages that multithreading offered in an architecture along with the factors that make it difficult to derive maximum benefits from this approach [Assessment] • Describe the relevance of scalability to performance [Assessment]	• Superscalar architecture • Branch prediction, Speculative execution, Out-of-order execution • Prefetching • Vector processors and GPUs • Hardware support for multithreading • Scalability • Alternative architectures, such as VLIW/EPIC, and Accelerators and other kinds of Special-Purpose Processors
Readings : [Par05], [Par02], [PH04], [Don06], [Joh91]	