

## 1. COURSE

CS221. Computer Systems Architecture (Mandatory)

## 2. GENERAL INFORMATION

2.1 Course	:	CS221. Computer Systems Architecture
2.2 Semester	:	3 <sup>er</sup> Semestre.
2.3 Credits	:	3
2.4 Horas	:	2 HT; 2 HP;
2.5 Duration of the period	:	16 weeks
2.6 Type of course	:	Mandatory
2.7 Learning modality	:	Face to face
2.8 Prerequisites	:	CS112. Programming I. (2 <sup>nd</sup> Sem) CS112. Programming I. (2 <sup>nd</sup> Sem)

## 3. PROFESSORS

Meetings after coordination with the professor

## 4. INTRODUCTION TO THE COURSE

A computer scientist must have a solid knowledge of the organization and design principles of diverse computer systems, by understanding the limitations of modern systems they could propose next-gen paradigms. This course teaches the basics and principles of Computer Architecture. This class addresses digital logic design, basics of Computer Architecture and processor design (Instruction Set architecture, microarchitecture, out-of-order execution, branch prediction), execution paradigms (superscalar, dataflow, VLIW, SIMD, GPUs, systolic, multithreading) and memory system organization.

## 5. GOALS

- Provide a first approach in Computer Architecture.
- Study the design and evolution of computer architectures, which lead to modern approaches and implementations in computing systems.
- Provide fine-grained details of computer hardware, and its relation with software execution.
- Implement a simple microprocessor using Verilog language.

## 6. COMPETENCES

- 1) Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions. (**Usage**)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (**Assessment**)

## 7. TOPICS

Unit 1: Lógica digital y sistemas digitales (18)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Revisión e historia de la Arquitectura de Computadores.</li> <li>• Lógica combinacional y secuencial/<i>field programmable gate arrays</i> como bloque fundamental de construcción lógico combinacional secuencial.</li> <li>• Modelos de representación(abstracción)</li> <li>• Herramientas de diseño asistidas por computadora que procesan hardware y representaciones arquitecturales.</li> <li>• Registrar transferencia notación / Hardware language descriptivo (Verilog/VHDL)</li> <li>• Restriccion física (Retrasos de Entrada, fan-in, fan-out, energía/potencia)</li> </ul>	<ul style="list-style-type: none"> <li>• Describir el avance de la tecnología de dispositivos, desde los tubos de vacío hasta VLSI, desde las arquitecturas mainframe a las arquitecturas en escala warehouse [Familiarity]</li> <li>• Comprender que la tendencia de las arquitecturas modernas de computadores es hacia núcleos múltiples y que el paralelismo es inherente en todos los sistemas de hardware [Usage]</li> <li>• Explicar las implicancias de los límites de potencia para mejoras adicionales en el rendimiento de los procesadores y también en el aprovechamiento del paralelismo [Usage]</li> <li>• Relacionar las varias representaciones equivalentes de la funcionalidad de un computador, incluyendo expresiones y puertas lógicas, y ser capaces de utilizar expresiones matemáticas para describir las funciones de circuitos combinacionales y secuenciales sencillos [Familiarity]</li> <li>• Diseñar los componentes básicos de construcción de un computador: unidad aritmético lógica (a nivel de puertas lógicas), unidad central de procesamiento (a nivel de registros de transferencia), memoria (a nivel de registros de transferencia) [Usage]</li> <li>• Usar herramientas CAD para capturar, sistetizar, y simular bloques de construcción (como ALUs, registros, movimiento entre registros) de un computador simple [Familiarity]</li> <li>• Evaluar el comportamiento de un diagrama de tiempos y funcional de un procesador simple implementado a nivel de circuitos lógicos [Assessment]</li> </ul>
<b>Readings :</b> [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unit 2: Representación de datos a nivel máquina (8)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Bits, Bytes y Words.</li> <li>• Representacion de datos numérica y bases numéricas.</li> <li>• Sistemas de punto flotante y punto fijo.</li> <li>• Representaciones con signo y complemento a 2.</li> <li>• Representación de información no numérica (códigos de caracteres, información gráfica)</li> <li>• Representación de registros y arreglos.</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar porqué en computación todo es datos, inclusive las instrucciones [Assessment]</li> <li>• Explicar las razones de usar formatos alternativos para representar datos numéricos [Familiarity]</li> <li>• Describir cómo los enteros negativos se almacenan con representaciones de bit de signo y complemento a 2 [Usage]</li> <li>• Explicar cómo las representaciones de tamaño fijo afectan en la exactitud y la precisión [Usage]</li> <li>• Describir la representación interna de datos no numéricos como caracteres, cadenas, registros y arreglos [Usage]</li> <li>• Convertir datos numéricos de un formato a otro [Usage]</li> </ul>
<b>Readings :</b> [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unit 3: Organización de la Máquina a Nivel Ensamblador (8)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Organización Básica de la Máquina de Von Neumann.</li> <li>• Unidad de Control.</li> <li>• <i>Instruction sets</i> y tipos (manipulación de información, control, I/O)</li> <li>• Assembler y Programación en Lenguaje de Máquina.</li> <li>• Formato de instrucciones.</li> <li>• Modos de direccionamiento.</li> <li>• Llamada a subrutinas y mecanismos de retorno.</li> <li>• I/O e Interrupciones.</li> <li>• Montículo (Heap) vs. Estático vs. Pila vs. Segmentos de código.</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar la organización de la maquina clásica de von Neumann y sus principales unidades funcionales [Familiarity]</li> <li>• Describir cómo se ejecuta una instrucción en una máquina de von Neumann con extensión para hebras, sincronización multiproceso y ejecución SIMD (máquina vectorial) [Familiarity]</li> <li>• Describir el paralelismo a nivel de instrucciones y sus peligros, y cómo es esto tratado en pipelines de proceso típicos [Familiarity]</li> <li>• Resumir cómo se representan las instrucciones, tanto a nivel de máquina bajo el contexto de un ensamblador simbólico [Familiarity]</li> <li>• Demostrar cómo se mapean los patrones de lenguajes de alto nivel en notaciones en lenguaje ensamblador o en código máquina [Usage]</li> <li>• Explicar los diferentes formatos de instrucciones, así como el direccionamiento por instrucción, y comparar formatos de tamaño fijo y variable [Usage]</li> <li>• Explicar como las llamadas a subrutinas son manejadas a nivel de ensamblador [Usage]</li> <li>• Explicar los conceptos básicos de interrupciones y operaciones de entrada y salida (I/O) [Familiarity]</li> <li>• Escribir segmentos de programa simples en lenguaje ensamblador [Usage]</li> <li>• Ilustrar cómo los bloques constructores fundamentales en lenguajes de alto nivel son implementados a nivel de lenguaje máquina [Usage]</li> </ul>
<b>Readings :</b> [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unit 4: Organización funcional (8)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Implementación de <i>datapath</i>, incluyendo un <i>pipeline</i> de instrucciones, detección de <i>hazards</i> y la resolución.</li> <li>• Control de unidades: Microprogramada.</li> <li>• Instrucción (Pipelining)</li> <li>• Introducción al paralelismo al nivel de instrucción (PNI)</li> </ul>	<ul style="list-style-type: none"> <li>• Comparar implementaciones alternativas de ruta de datos [Assessment]</li> <li>• Discutir el concepto de puntos de control y la generación de señales de control usando implementaciones a nivel de circuito o microprogramadas [Familiarity]</li> <li>• Explicar el paralelismo a nivel de instrucciones básicas usando pipelining y los mayores riesgos que pueden ocurrir [Usage]</li> <li>• Diseñar e implementar un procesador completo, incluyendo ruta de datos y control [Usage]</li> <li>• Calcular la cantidad promedio de ciclos por instrucción de una implementación con procesador y sistema de memoria determinados [Assessment]</li> </ul>
<b>Readings :</b> [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unit 5: Organización y Arquitectura del Sistema de Memoria (8)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Sistemas de Almacenamiento y su Tecnología.</li> <li>• Jerarquía de Memoria: importancia de la localización temporal y espacial.</li> <li>• Organización y Operaciones de la Memoria Principal.</li> <li>• Latencia, ciclos de tiempo, ancho de banda e <i>inter-leading</i>.</li> <li>• Memorias caché (Mapeo de direcciones, Tamaño de bloques, Reemplazo y Políticas de almacenamiento)</li> <li>• Multiprocesador coherencia cache / Usando el sistema de memoria para las operaciones de sincronización de memoria / atómica inter-core.</li> <li>• Memoria virtual (tabla de página, TLB)</li> <li>• Manejo de Errores y confiabilidad.</li> <li>• Error de codificación, compresión de datos y la integridad de datos.</li> </ul>	<ul style="list-style-type: none"> <li>• Identificar las principales tecnologías de memoria (Por ejemplo: SRAM, DRAM, Flash, Disco Magnético) y su relación costo beneficio [Familiarity]</li> <li>• Explique el efecto de latencia de memoria en tiempo de ejecución [Familiarity]</li> <li>• Describir como el uso de jerarquía de memoria (caché, memoria virtual) es aplicado para reducir el atraso efectivo en la memoria [Usage]</li> <li>• Describir los principios de la administración de memoria [Usage]</li> <li>• Explique el funcionamiento de un sistema con gestión de memoria virtual [Usage]</li> <li>• Calcule el tiempo de acceso promedio a memoria bajo varias configuraciones de caché y memoria y para diversas combinaciones de instrucciones y referencias a datos [Assessment]</li> </ul>
<b>Readings :</b> [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unit 6: Interfaz y comunicación (8)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Fundamentos de I/O: Handshaking, Bbuffering, I/O programadas, interrupciones dirigidas de I/O.</li> <li>• Interrumpir estructuras: interrumpir reconocimiento, vectorizado y priorizado.</li> <li>• Almacenamiento externo, organización física y discos.</li> <li>• Buses: Protocolos de bus, arbitraje, acceso directo a memoria (DMA).</li> <li>• Introducción a Redes: comunicación de redes como otra capa de acceso remoto.</li> <li>• Soporte Multimedia.</li> <li>• Arquitecturas RAID.</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar como las interrupciones son aplicadas para implementar control de entrada-salida y transferencia de datos [Familiarity]</li> <li>• Identificar diversos tipos de buses en un sistema computacional [Familiarity]</li> <li>• Describir el acceso a datos desde una unidad de disco magnético [Usage]</li> <li>• Comparar organizaciones de red conocidas como organizaciones en bus/Ethernet, en anillo y organizaciones conmutadas versus ruteadas [Assessment]</li> <li>• Identificar las interfaces entre capas necesarios para el acceso y presentación multimedia, desde la captura de la imagen en almacenamiento remoto, a través del transporte por una red de comunicaciones, hasta la puesta en la memoria local y la presentación final en una pantalla gráfica [Familiarity]</li> <li>• Describir las ventajas y limitaciones de las arquitecturas RAID [Familiarity]</li> </ul>
<b>Readings :</b> [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unit 7: Multiprocesamiento y arquitecturas alternativas (8)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• <i>Power Law</i>.</li> <li>• Ejemplos de <i>sets</i> de instrucciones y arquitecturas SIMD y MIMD.</li> <li>• Redes de interconexión (Hypercube, Shuffle-exchange, Mesh, Crossbar)</li> <li>• Sistemas de memoria de multiprocesador compartido y consistencia de memoria.</li> <li>• Coherencia de cache multiprocesador.</li> </ul>	<ul style="list-style-type: none"> <li>• Discutir el concepto de procesamiento paralelo mas allá del clásico modelo de von Neumann [Assessment]</li> <li>• Describir diferentes arquitecturas paralelas como SIMD y MIMD [Familiarity]</li> <li>• Explicar el concepto de redes de interconexión y mostrar diferentes enfoques [Usage]</li> <li>• Discutir los principales cuidados en los sistemas de multiprocesamiento presentes con respecto a la gestión de memoria y describir como son tratados [Familiarity]</li> <li>• Describir las diferencias entre conectores eléctricos en paralelo backplane, interconexión memoria procesador y memoria remota via red, sus implicaciones para la latencia de acceso y el impacto en el rendimiento de un programa [Assessment]</li> </ul>
<b>Readings :</b> [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unit 8: Mejoras de rendimiento (8)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Arquitectura superescalar.</li> <li>• Predicción de ramificación, Ejecución especulativa, Ejecución fuera de orden.</li> <li>• Prefetching.</li> <li>• Procesadores vectoriales y GPU's</li> <li>• Soporte de hardware para multiprocesamiento.</li> <li>• Escalabilidad.</li> <li>• Arquitecturas alternativas, como VLIW / EPIC y aceleradores y otros tipos de procesadores de propósito especial.</li> </ul>	<ul style="list-style-type: none"> <li>• Describir las arquitecturas superescalares y sus ventajas [Familiarity]</li> <li>• Explicar el concepto de predicción de bifurcaciones y su utilidad [Usage]</li> <li>• Caracterizar los costos y beneficios de la precarga prefetching [Assessment]</li> <li>• Explicar la ejecución especulativa e identifique las condiciones que la justifican [Assessment]</li> <li>• Discutir las ventajas de rendimiento ofrecida en una arquitectura de multihebras junto con los factores que hacen difícil dar el máximo beneficio de estas [Assessment]</li> <li>• Describir la importancia de la escalabilidad en el rendimiento [Assessment]</li> </ul>
<b>Readings :</b> [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

## 8. WORKPLAN

### 8.1 Methodology

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

### 8.2 Theory Sessions

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

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BASIC BIBLIOGRAPHY

- [HH12] David Harris and Sarah Harris. *Digital Design and Computer Architecture*. 2nd. Morgan Kaufmann, 2012. ISBN: 978-0123944245.
- [HP06] J. L. Hennessy and D. A. Patterson. *Computer Architecture: A Quantitative Approach*. 4th. San Mateo, CA: Morgan Kaufman, 2006.
- [JAs07] Peter J. Ashenden. *Digital Design (Verilog): An Embedded Systems Approach Using Verilog*. Morgan Kaufmann, 2007. ISBN: 978-0123695277.
- [Par05] Behrooz Parhami. *Computer Architecture: From Microprocessors to Supercomputers*. New York: Oxford Univ. Press, 2005. ISBN: ISBN 0-19-515455-X.
- [PCh06] Pong P. Chu. *RTL Hardware Design Using VHDL*. 1st. Wiley-Interscience, 2006.
- [PH04] D. A. Patterson and J. L. Hennessy. *Computer Organization and Design: The Hardware/Software Interface*. 3rd ed. San Mateo, CA: Morgan Kaufman, 2004.
- [PP05] Yale N. Patt and Sanjay J. Patel. *Introduction to Computing Systems*. 2nd. McGraw Hill, 2005.
- [Sta10] William Stalings. *Computer Organization and Architecture: Designing for Performance*. 8th. Upper Saddle River, NJ: Prentice Hall, 2010.