



Universidad Nacional San Cristobal de Huamanga (UNSCH)

Programa Profesional de
Ciencia de la Computación
Sílabo 2024-II

1. CURSO

CS221. Computer Systems Architecture (Mandatory)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS221. Computer Systems Architecture
2.2 Semestre	:	3 rd Semester.
2.3 Créditos	:	3
2.4 horas	:	2 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Mandatory
2.7 Modalidad de aprendizaje	:	Face to face
2.8 Prerrequisitos	:	CS1D2. Discrete Structures II. (2 nd Sem) CS1D2. Discrete Structures II. (2 nd Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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. OBJETIVOS

- 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. RESULTADOS DEL ESTUDIANTE

- 1) Analyze a complex computing problem and 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. TEMAS

Unidad 1: Lógica digital y sistemas digitales (18)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje (<i>Learning Outcomes</i>)
• Revisión e historia de la Arquitectura de Computadores. • Lógica combinatorial vs. secuencial/Arreglos de puertas de campo programables como bloque fundamental de construcción lógico combinatorial-secuencial. • Múltiples representaciones / Capas de interpretación (El hardware es solo otra capa) • Herramientas de diseño asistidas por computadora que procesan hardware y representaciones arquitecturales. • Registrar transferencia notación / Hardware language descriptivo (Verilog/VHDL) • Restricción física (Retrasos de Entrada, fan-in, fan-out, energía/poder)	• Describir el avance paulatino de los componentes de la tecnología de computación, desde los tubos de vacío hasta VLSI, desde las arquitecturas mainframe a las arquitecturas en escala warehouse [Familiarizarse] • 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 [Usar] • Explicar las implicancias de los límites de potencia para mejoras adicionales en el rendimiento de los procesadores y también en el aprovechamiento del paralelismo [Usar] • 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 combinatoriales y secuenciales sencillos [Familiarizarse] • 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) [Usar] • Usar herramientas CAD para capturar, sistematizar, y simular bloques de construcción (como ALUs, registros, movimiento entre registros) de un computador simple [Familiarizarse] • Evaluar el comportamiento de un diagrama de tiempos y funcional de un procesador simple implementado a nivel de circuitos lógicos [Evaluar]
Lecturas : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unidad 2: Representación de datos a nivel máquina (8)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje (<i>Learning Outcomes</i>)
• Bits, Bytes y Words. • Representación de datos numérica y bases numéricas. • Sistemas de punto flotante y punto fijo. • Representaciones con signo y complemento a 2. • Representación de información no numérica (códigos de caracteres, información gráfica) • Representación de registros y arreglos.	• Explicar porqué en computación todo es datos, inclusive las instrucciones [Evaluar] • Explicar las razones de usar formatos alternativos para representar datos numéricos [Familiarizarse] • Describir cómo los enteros negativos se almacenan con representaciones de bit de signo y complemento a 2 [Usar] • Explicar cómo las representaciones de tamaño fijo afectan en la exactitud y la precisión [Usar] • Describir la representación interna de datos no numéricos como caracteres, cadenas, registros y arreglos [Usar] • Convertir datos numéricos de un formato a otro [Usar]
Lecturas : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unidad 3: Organización de la Máquina a Nivel Ensamblador (8)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje (<i>Learning Outcomes</i>)
• Organización Básica de la Máquina de Von Neumann. • Unidad de Control. • Paquetes de instrucciones y tipos (manipulación de información, control, I/O) • Assembler / Programación en Lenguaje de Máquina. • Formato de instrucciones. • Modos de direccionamiento. • Llamada a subrutinas y mecanismos de retorno. • I/O e Interrupciones. • Montículo (Heap) vs. Estático vs. Pila vs. Segmentos de código.	• Explicar la organización de la maquina clásica de von Neumann y sus principales unidades funcionales [Familiarizarse] • Describir cómo se ejecuta una instrucción en una máquina de von Neumann con extensión para hebras, sincronización multiproceso y ejecución SIMD (máquina vectorial) [Familiarizarse] • Describir el paralelismo a nivel de instrucciones y sus peligros, y cómo es esto tratado en pipelines de proceso típicos [Familiarizarse] • Resumir cómo se representan las instrucciones, tanto a nivel de máquina bajo el contexto de un ensamblador simbólico [Familiarizarse] • Demostrar cómo se mapean los patrones de lenguajes de alto nivel en notaciones en lenguaje ensamblador o en código máquina [Usar] • Explicar los diferentes formatos de instrucciones, así como el direccionamiento por instrucción, y comparar formatos de tamaño fijo y variable [Usar] • Explicar como las llamadas a subrutinas son manejadas a nivel de ensamblador [Usar] • Explicar los conceptos básicos de interrupciones y operaciones de entrada y salida (I/O) [Familiarizarse] • Escribir segmentos de programa simples en lenguaje ensamblador [Usar] • Ilustrar cómo los bloques constructores fundamentales en lenguajes de alto nivel son implementados a nivel de lenguaje máquina [Usar]
Lecturas : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unidad 4: Organización funcional (8)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje (<i>Learning Outcomes</i>)
• Implementación de rutas de datos simples, incluyendo la canalización de instrucciones, detección de riesgos y la resolución. • Control de unidades: Realización Cableada vs Realización Microprogramada. • Instrucción (Pipelining) • Introducción al paralelismo al nivel de instrucción (PNI)	• Comparar implementaciones alternativas de ruta de datos [Evaluar] • Discutir el concepto de puntos de control y la generación de señales de control usando implementaciones a nivel de circuito o microprogramadas [Familiarizarse] • Explicar el paralelismo a nivel de instrucciones básicas usando pipelining y los mayores riesgos que pueden ocurrir [Usar] • Diseñar e implementar un procesador completo, incluyendo ruta de datos y control [Usar] • Calcular la cantidad promedio de ciclos por instrucción de una implementación con procesador y sistema de memoria determinados [Evaluar]
Lecturas : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unidad 5: Organización y Arquitectura del Sistema de Memoria (8)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje (<i>Learning Outcomes</i>)
• Sistemas de Almacenamiento y su Tecnología. • Jerarquía de Memoria: importancia de la localización temporal y espacial. • Organización y Operaciones de la Memoria Principal. • Latencia, ciclos de tiempo, ancho de banda e intercalación. • Memorias caché (Mapeo de direcciones, Tamaño de bloques, Reemplazo y Políticas de almacenamiento) • Multiprocesador coherencia cache / Usando el sistema de memoria para las operaciones de sincronización de memoria / atómica inter-core. • Memoria virtual (tabla de página, TLB) • Manejo de Errores y confiabilidad. • Error de codificación, compresión de datos y la integridad de datos.	• Identifique las principales tecnologías de memoria (Por ejemplo: SRAM, DRAM, Flash, Disco Magnético) y su relación costo beneficio [Familiarizarse] • Explique el efecto del retardo de la memoria en tiempo de ejecución [Familiarizarse] • Describa como el uso de jerarquía de memoria (caché, memoria virtual) es aplicado para reducir el retardo efectivo en la memoria [Usar] • Describa como el uso de jerarquía de memoria (caché, memoria virtual) es aplicado para reducir el retardo efectivo en la memoria [Usar] • Explique el efecto del retardo de la memoria en tiempo de ejecución [Usar] • 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 [Evaluar]
Lecturas : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unidad 6: Interfaz y comunicación (8)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje (<i>Learning Outcomes</i>)
• Fundamentos de I/O: Handshaking, Bbuffering, I/O programadas, interrupciones dirigidas de I/O. • Interrumpir estructuras: interrumpir reconocimiento, vectorizado y priorizado. • Almacenamiento externo, organización física y discos. • Buses: Protocolos de bus, arbitraje, acceso directo a memoria (DMA). • Introducción a Redes: comunicación de redes como otra capa de acceso remoto. • Soporte Multimedia. • Arquitecturas RAID.	• Explicar como las interrupciones son aplicadas para implementar control de entrada-salida y transferencia de datos [Familiarizarse] • Identificar diversos tipos de buses en un sistema computacional [Familiarizarse] • Describir el acceso a datos desde una unidad de disco magnético [Usar] • Comparar organizaciones de red conocidas como organizaciones en bus/Ethernet, en anillo y organizaciones conmutadas versus ruteadas [Evaluar] • 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 [Familiarizarse] • Describir las ventajas y limitaciones de las arquitecturas RAID [Familiarizarse]
Lecturas : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unidad 7: Multiprocesamiento y arquitecturas alternativas (8)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje (<i>Learning Outcomes</i>)
• Ley potencial. • Ejemplos de juego de instrucciones y arquitecturas SIMD y MIMD. • Redes de interconexión (Hypercube, Shuffle-exchange, Mesh, Crossbar) • Sistemas de memoria de multiprocesador compartido y consistencia de memoria. • Coherencia de cache multiprocesador.	• Discutir el concepto de procesamiento paralelo mas allá del clásico modelo de von Neumann [Evaluar] • Describir diferentes arquitecturas paralelas como SIMD y MIMD [Familiarizarse] • Explicar el concepto de redes de interconexión y mostrar diferentes enfoques [Usar] • Discutir los principales cuidados en los sistemas de multiprocesamiento presentes con respecto a la gestión de memoria y describir como son tratados [Familiarizarse] • Describir las diferencias entre conectores electricos 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 [Evaluar]
Lecturas : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unidad 8: Mejoras de rendimiento (8)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje (<i>Learning Outcomes</i>)
• Arquitectura superescalar. • Predicción de ramificación, Ejecución especulativa, Ejecución fuera de orden. • Prefetching. • Procesadores vectoriales y GPU's • Soporte de hardware para multiprocesamiento. • Escalabilidad. • Arquitecturas alternativas, como VLIW / EPIC y aceleradores y otros tipos de procesadores de propósito especial.	• Describir las arquitecturas superescalares y sus ventajas [Familiarizarse] • Explicar el concepto de predicción de bifurcaciones y su utilidad [Usar] • Caracterizar los costos y beneficios de la precarga prefetching [Evaluar] • Explicar la ejecución especulativa e identifique las condiciones que la justifican [Evaluar] • Discutir las ventajas de rendimiento ofrecida en una arquitectura de multihebras junto con los factores que hacen difícil dar el máximo beneficio de estas [Evaluar] • Describir la importancia de la escalabilidad en el rendimiento [Evaluar]
Lecturas : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

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