

Universidad Católica San Pablo (UCSP)
Escuela Profesional de
Ciencia de la Computación
SILABO



CS221. Arquitectura de Computadores (Obligatorio)

1. Información general

1.1 Escuela	:	Ciencia de la Computación
1.2 Curso	:	CS221. Arquitectura de Computadores
1.3 Semestre	:	3 ^{er} Semestre.
1.4 Prerrequisitos	:	CS1D2. Estructuras Discretas II. (2 ^{do} Sem)
1.5 Condición	:	Obligatorio
1.6 Modalidad de aprendizaje	:	Presencial
1.7 horas	:	2 HT; 2 HL;
1.8 Créditos	:	3

2. Profesores

3. Fundamentación del curso

Un profesional en Ciencia de la Computación debe tener un conocimiento sólido de la organización y los principios de diseño de diversos sistemas de computación, al comprender las limitaciones de los sistemas modernos serán capaces de proponer nuevos paradigmas en la próxima generación. Este curso enseña los fundamentos y principios de la arquitectura de computadoras. Esta clase incluye diseño de lógica digital, conceptos básicos de arquitectura de computadora y diseño de procesador (*Instruction Set Architecture*, microarquitectura, ejecución fuera de orden, predicción de *branches*), paradigmas de ejecución (superescalar, flujo de datos, VLIW, SIMD, GPU, sistólica, multiproceso) y organización del sistema de memoria.

4. Resumen

1. Lógica digital y sistemas digitales 2. Representación de datos a nivel máquina 3. Organización de la Máquina a Nivel Ensamblador 4. Organización funcional 5. Organización y Arquitectura del Sistema de Memoria 6. Interfaz y comunicación 7. Multiprocesamiento y arquitecturas alternativas 8. Mejoras de rendimiento

5. Objetivos Generales

- Proporcionar un primer enfoque en Arquitectura de Computadoras.
- Estudiar el diseño y la evolución de las arquitecturas de computador, que llevaron a las implementaciones de los sistemas modernos.
- Proporcionar un estudio profundo del hardware y su relación con la ejecución del software.
- Implementar un microprocesador simple usando el lenguaje Verilog.

6. Contribución a los resultados (*Outcomes*)

Esta disciplina contribuye al logro de los siguientes resultados de la carrera:

- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Evaluar**)

7. Contenido

UNIDAD 1: Lógica digital y sistemas digitales (18)	
Competencias: b	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • Revisión e historia de la Arquitectura de Computadores. • Lógica combinacional y secuencial/<i>field programmable gate arrays</i> como bloque fundamental de construcción lógico combinacional secuencial. • Modelos de representación(abstracción) • Herramientas de diseño asistidas por computadora que procesan hardware y representaciones arquitecturales. • Registrar transferencia notación / Hardware language descriptivo (Verilog/VHDL) • Restriccion física (Retrasos de Entrada, fan-in, fan-out, energía/potencia) 	<ul style="list-style-type: none"> • 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 [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 combinacionales 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, sistetizar, 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]
<p>Lecturas: D. Harris and S. Harris (2012), Patt and Patel (2005), Patterson and Hennessy (2004), J.Ashenden (2007), Hennessy and Patterson (2006), Parhami (2005), Stalings (2010), P.Chu (2006)</p>	

UNIDAD 2: Representación de datos a nivel máquina (8)	
Competencias:	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • Bits, Bytes y Words. • Representacion 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. 	<ul style="list-style-type: none"> • 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: D. Harris and S. Harris (2012), Patt and Patel (2005), Patterson and Hennessy (2004), J.Ashenden (2007), Hennessy and Patterson (2006), Parhami (2005), Stalings (2010), P.Chu (2006)	

UNIDAD 3: Organización de la Máquina a Nivel Ensamblador (8)	
Competencias: b	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • Organización Básica de la Máquina de Von Neumann. • Unidad de Control. • <i>Instruction sets</i> y tipos (manipulación de información, control, I/O) • Assembler y 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. 	<ul style="list-style-type: none"> • 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]
<p>Lecturas: D. Harris and S. Harris (2012), Patt and Patel (2005), Patterson and Hennessy (2004), J.Ashenden (2007), Hennessy and Patterson (2006), Parhami (2005), Stalings (2010), P.Chu (2006)</p>	

UNIDAD 4: Organización funcional (8)	
Competencias: b	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • Implementación de <i>datapath</i>, incluyendo un <i>pipeline</i> de instrucciones, detección de <i>hazards</i> y la resolución. • Control de unidades: Microprogramada. • Instrucción (Pipelining) • Introducción al paralelismo al nivel de instrucción (PNI) 	<ul style="list-style-type: none"> • 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: D. Harris and S. Harris (2012), Patt and Patel (2005), Patterson and Hennessy (2004), J.Ashenden (2007), Hennessy and Patterson (2006), Parhami (2005), Stalings (2010), P.Chu (2006)	

UNIDAD 5: Organización y Arquitectura del Sistema de Memoria (8)	
Competencias: b	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • 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 <i>inter-leading</i>. • 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. 	<ul style="list-style-type: none"> • Identificar las principales tecnologías de memoria (Por ejemplo: SRAM, DRAM, Flash, Disco Magnético) y su relación costo beneficio [Familiarizarse] • Explique el efecto de latencia de memoria en tiempo de ejecución [Familiarizarse] • Describir como el uso de jerarquía de memoria (caché, memoria virtual) es aplicado para reducir el atraso efectivo en la memoria [Usar] • Describir los principios de la administración de memoria [Usar] • Explique el funcionamiento de un sistema con gestión de memoria virtual [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: D. Harris and S. Harris (2012), Patt and Patel (2005), Patterson and Hennessy (2004), J.Ashenden (2007), Hennessy and Patterson (2006), Parhami (2005), Stalings (2010), P.Chu (2006)	

UNIDAD 6: Interfaz y comunicación (8)	
Competencias: b,i	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • 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. 	<ul style="list-style-type: none"> • 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: D. Harris and S. Harris (2012), Patt and Patel (2005), Patterson and Hennessy (2004), J.Ashenden (2007), Hennessy and Patterson (2006), Parhami (2005), Stalings (2010), P.Chu (2006)	

UNIDAD 7: Multiprocesamiento y arquitecturas alternativas (8)	
Competencias: i	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • <i>Power Law</i>. • Ejemplos de <i>sets</i> 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. 	<ul style="list-style-type: none"> • 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: D. Harris and S. Harris (2012), Patt and Patel (2005), Patterson and Hennessy (2004), J.Ashenden (2007), Hennessy and Patterson (2006), Parhami (2005), Stalings (2010), P.Chu (2006)	

UNIDAD 8: Mejoras de rendimiento (8)	
Competencias: i	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • 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. 	<ul style="list-style-type: none"> • 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: D. Harris and S. Harris (2012), Patt and Patel (2005), Patterson and Hennessy (2004), J.Ashenden (2007), Hennessy and Patterson (2006), Parhami (2005), Stalings (2010), P.Chu (2006)	

8. Metodología
<p>El profesor del curso presentará clases teóricas de los temas señalados en el programa propiciando la intervención de los alumnos.</p> <p>El profesor del curso presentará demostraciones para fundamentar clases teóricas.</p> <p>El profesor y los alumnos realizarán prácticas</p> <p>Los alumnos deberán asistir a clase habiendo leído lo que el profesor va a presentar. De esta manera se facilitará la comprensión y los estudiantes estarán en mejores condiciones de hacer consultas en clase.</p>

9. Evaluar
<p>Evaluación Continua 1 : 20 %</p> <p>Examen parcial : 30 %</p> <p>Evaluación Continua 2 : 20 %</p> <p>Examen final : 30 %</p>

References

- Harris, David and Sarah Harris (2012). *Digital Design and Computer Architecture*. 2nd. Morgan Kaufmann. ISBN: 978-0123944245.
- Hennessy, J. L. and D. A. Patterson (2006). *Computer Architecture: A Quantitative Approach*. 4th. Morgan Kaufman: San Mateo, CA.
- J.Ashenden, Peter (2007). *Digital Design (Verilog): An Embedded Systems Approach Using Verilog*. Morgan Kaufmann. ISBN: 978-0123695277.
- P.Chu, Pong (2006). *RTL Hardware Design Using VHDL*. 1st. Wiley-Interscience.
- Parhami, Behrooz (2005). *Computer Architecture: From Microprocessors to Supercomputers*. Oxford Univ. Press: New York. ISBN: ISBN 0-19-515455-X.
- Patt, Yale N and Sanjay J Patel (2005). *Introduction to Computing Systems*. 2nd. McGraw Hill.

- Patterson, D. A. and J. L. Hennessy (2004). *Computer Organization and Design: The Hardware/Software Interface*. 3rd ed. Morgan Kaufman: San Mateo, CA.
- Stalings, William (2010). *Computer Organization and Architecture: Designing for Performance*. 8th. Prentice Hall: Upper Saddle River, NJ.