



FACULTAD DE CIENCIAS EXACTAS, FÍSICAS y NATURALES



Universidad
Nacional
de Córdoba

Asignatura: **Arquitectura de Computadoras**

Código: 10-09815

RTF

9

Semestre: 8

Carga Horaria

96

Bloque: Tecnologías Aplicadas

Horas de Práctica

48

Departamento: Computación

Correlativas:

- Sistemas Operativos

Contenido Sintético:

- Historia y descripción general.
- Arquitectura y set de instrucciones.
- Medición del rendimiento
- Organización del procesador
- Organización y arquitectura del sistema de memoria
- Paralelismo a nivel de instrucciones
- Arquitecturas multi-núcleo, multiprocesadores
- Computadoras Vectoriales.
- Arquitecturas de sistemas distribuidos, arquitecturas alternativas
- Caso de estudio realizado con herramientas utilizadas en el diseño de arquitectura de computadoras de diseño de un procesador de arquitectura abierta, haciendo uso de Verilog.

Competencias Genéricas:

- CG1: Identificar, formular y resolver problemas de ingeniería.
- CG2: Concebir, diseñar y desarrollar proyectos de ingeniería (sistemas, componentes, productos o procesos).
- CG5: Contribuir a la generación de desarrollos tecnológicos y/o innovaciones tecnológicas.
- CG6: Desempeñarse de manera efectiva en equipos de trabajo.
- CG7: Comunicarse con efectividad.
- CG10: Actuar con espíritu emprendedor.

Aprobado por HCD:1042-HCD-2023

RES: Fecha: 27/11/2023

Competencias Específicas:

- CE2 Diseñar, desarrollar e implementar diversas Arquitecturas de Computadoras y todos los subsistemas relacionados.
 - CE2.1 Conocer profundamente los principios fundamentales de la arquitectura de computadoras, incluyendo la estructura de la CPU, la memoria, el sistema de entrada/salida y la interconexión entre los componentes principales.
 - CE2.2 Utilizar correctamente lenguajes de descripción de hardware,
 - CE2.3 Poseer habilidades en la escritura y comprensión de código HDL, así como en la simulación y verificación del diseño.
 - CE2.4 Analizar, modelar, diseñar, desarrollar y probar circuitos electrónicos digitales
 - CE2.5 Utilizar técnicas de optimización de rendimiento de las CPUs.
 - CE2.6 Analizar, modelar y diseñar microarquitecturas y pipelines.
 - CE2.7 Conocer e implementar conceptos de organización y gestión de la memoria en un sistema de computadoras.
 - CE2.8 Conocer e implementar los diferentes esquemas de interconexión utilizados en sistemas de computadoras,
 - CE2.9 Conocer e implementar los protocolos de

Presentación

Objetivos:

- Adquirir: La dimensión tecnológica necesaria para la comprensión, construcción e implementación de las estructuras internas, su organización e interrelación de computadores digitales.
- Comprender: Todas las reglas y principios tecnológicos que nos permita independizarnos de los dispositivos desarrollados en el momento, como así también una profunda práctica en el manejo de dispositivos actuales para así afianzar los conceptos teóricos y un manejo acabado de la realidad tecnológica.

LINEAMIENTOS GENERALES

La presente asignatura es una actividad curricular que pertenece al octavo semestre de la carrera de Ingeniería en Computación.

A través del cursado de la asignatura el alumno desarrollará las siguientes competencias:

- Comprensión, construcción e implementación de las estructuras internas, su organización e interrelación de computadores digitales.
- Identificar y construir estructuras internas, su organización e interrelación de computadores digitales con el fin de comprenderlos e implementarlos.
- Mediante el análisis identificar problemas de implementación en distintas arquitecturas.
- Saber identificar los distintos dominios de aplicación de las distintas arquitecturas.

- Conocer la interacción de la arquitectura con su entorno; se pondrá énfasis en la detección y prevención de situaciones problemáticas tales como la performance en pipeline, la eficiencia en threads y en procesos (paralelismo temporal y espacial).
- Buscar la eficiencia y la eficacia del sistema haciendo uso de implementaciones paralelas.
- Mediante el uso de herramientas y criterios, realizar el diseño de arquitecturas paralelas, basado en la definición de thread, procesos, e interacciones.
- Realizar la implementación de un sistema con aplicaciones paralelas

En los últimos años se han producido profundos cambios en las arquitecturas de computadoras. Las arquitecturas súper escalar, múltiples core y NUMA; comenzaron a dominar el mercado, con lo que han emergido un sin número de nuevos conceptos como predecodificación, Dynamic Branch, Prediction, instruction reordering, speculation execution of loads e implementación de procesadores CISC por un RISC core. Estas ideas han sido implementadas en las principales arquitecturas de procesadores. Más aun, existen lenguajes que explotan estos beneficios, por lo que se hará énfasis en:

- Arquitectura computacionales que facilitan y promueven la programación en paralelo
- Herramientas de depuración que facilitan las pruebas.

El dictado se orienta a capacitar al alumno para Identificar y construir modelos de sistemas donde este involucradas las arquitecturas paralelas, lograr sistemas con eficiencia y eficacia en los cuales las arquitecturas sean concordantes con los requerimientos. Todo materializando un diseño, implementación y las pruebas de sistema.

Contenidos

Unidad 1. Introducción

Tecnología y arquitectura.

Medidas de la calidad de las arquitecturas.

Medidas de performance, precio y consumo.

Factores de influencia de las distintas arquitecturas. Ejemplos.

Unidad 2. Sistema de memoria

Sistema de memoria tecnologías y costo.

Memoria principal. Jerarquía de memoria. Ejemplos.

Unidad 3. Set de Instrucciones

Representación de datos.

Tipos y precisión. Set de registros.

Tipo de instrucciones. Técnicas de direccionamiento.

Diseño de set de instrucciones. Ejemplos.

Unidad 4. Paralelismo a nivel de instrucciones (ILP).

Conceptos básicos de pipelining. Técnicas de diseño de pipeline. Control de pipeline.

Técnicas de compilador para explotar ILP. Ejemplos.

Unidad 5. Planificación dinámica

Predicción de saltos. Especificación basada en hardware.

Paralelismo a nivel de hilos. Sistemas distribuidos.

Unidad 6. Arquitecturas vectoriales

Procesador vectorial genérico. SIMD. Ejemplos.

Unidad 7. Arquitecturas de propósito general.

MIPS. RISC V. ARM-Cortex A53. Intel Core i7.

Unidad 8. Multiprocesadores

Multiprocesadores performance.

Multiprocesadores

interconexión.

Coherencia en el cache de multiprocesadores.

Algoritmos de multiprocesadores.

Simple paralelismos, técnicas de sincronización, transformación de algoritmos seriales en paralelos. Ejemplos.

Unidad 9. Arquitecturas Alternativas

Arquitecturas de flujo de datos.

Deep Neural Networks. Tensor Processor Unit. GPUs.

Metodología de enseñanza

Las clases impartidas son teóricas, prácticas y de laboratorio.

Las actividades teóricas se realizan a través de exposiciones dialogadas del docente orientadas a desarrollar en los alumnos la capacidad de saber identificar la concurrencia, la interacción de la aplicación con su entorno, el diseño y las pruebas de sistemas.

Durante el desarrollo de los Trabajos Prácticos se realizan actividades que le permiten al estudiante poner en práctica las habilidades y verificar los criterios y técnicas de modelado, diseño y prueba.

Por otra parte en las clases de Laboratorio el alumno verifica, a través de la implementación, el funcionamiento de sistemas y realizará los casos de pruebas.

Evaluación

La evaluación formativa se realizará mediante la realización de las actividades prácticas. El desarrollo de las competencias se evaluará de forma continua mediante rúbricas, construida en base a los resultados de aprendizaje propuestos. Se realizarán también evaluaciones parciales.

Condiciones para la promoción de la materia

1. Tener aprobadas las materias correlativas.
2. Asistir al 80% de las clases teóricas y prácticas.
3. Rendir y aprobados los dos parciales con 50% , el alumno podrá recuperar una vez cada parcial.
4. Presentar y aprobar los trabajos que se exijan durante el desarrollo de los trabajos prácticos.
5. Aprobar los trabajos de Laboratorio.
6. Aprobar un coloquio integrador.
7. Alcanzar el criterio de aceptación mínimo para los resultados de aprendizaje propuestos

Los alumnos que cumplan las exigencias referidas en los puntos 1 al 7 serán considerados promocionados. Los alumnos que cumplan las exigencias referidas en los puntos 1 al 5 serán considerados regulares. El resto será considerado libre.

Actividades prácticas y de laboratorio

Actividades Prácticas

1. Herramientas e implementar sistemas digitales. Bases de Verilog, Estructuras de Verilog, Concurrencia y Secuencialidad, Inferencia de elementos de memoria (voluntaria e involuntaria), Instancias de arquitectura, multi-instancias, verificación
2. Herramientas e implementación de sistemas paralelos en el tiempo. Otras estructuras y sentencias, Subprogramas: funciones y procedimientos, Tipos y Packages aritméticos, Uso eficiente de operadores aritméticos, Diseño de máquinas de estado con Verilog. Diferentes técnicas para diseñar arquitecturas mas eficientes
3. Implementación de sistemas paralelos en el tiempo. Herramientas para el diseño jerárquico, Reglas generales de diseño para optimizar la performance, Manejo de archivos en Verilog, Modelado del tiempo en Verilog, Verificación funcional de modelos Verilog, Ejemplos con Xilinx en Verilog.

Actividades de Laboratorio

1. Implementación de distintos diseños por clases
2. Implementación de un ALU
3. Implementación de un modulo de comunicacion UART
4. Implementación del pipeline simplificado de un procesador MIPS / RISC

Desagregado de competencias y resultados de aprendizaje

Al final del curso se espera que un estudiante alcance los siguientes habilidades:

- Interpreta correctamente el dominio de un problema.
- Posee las habilidades comunicacionales suficientes para realizar las preguntas necesarias para desarrollar un diseño completo ajustado a las necesidades del dominio presentes y futuras.
- Cumple en tiempo con los compromisos asumidos con su equipo de trabajo.
- Respeta las pautas de trabajo establecidas en clase para las actividades de equipo.
- Comprender el funcionamiento de los procesadores y computadoras modernas.
- Identificar y construir estructuras internas del computador, su organización e interrelación de computadores digitales.
- Identificar problemas de implementación en distintas arquitecturas.
- Identificar los distintos dominios de aplicación de las distintas arquitecturas.
- Buscar la eficiencia y la eficacia del sistema haciendo uso de implementaciones paralelas.
- Realizar el diseño de arquitecturas paralelas, basado en la definición de thread, procesos, e interacciones.

Bibliografía

- Computer Architecture A Quantitative Approach 5ta Edición, David A. Patterson, John L. Hennessy
- Computer Architecture A Quantitative Approach 6ta Edición, David A. Patterson, John L. Hennessy
- Introduction to Parallel Computing, Second Edition, By Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar
- FPGA Prototyping by Verilog Examples. Pong P. Chu.
- Computer Organization and Design RISC-V Edition: The Hardware Software Interface. 2da Edición. David A. Patterson, John L. Hennessy
- Computer Organization and Design: The Hardware Software Interface. Revised 4th Edition. David A. Patterson, John L. Hennessy