

Programa del curso CE-4301

Arquitectura de Computadores I

Área Académica de Ingeniería en Computadores

Programa de Licenciatura en Ingeniería en Computadores.

I parte: Aspectos relativos al plan de estudios

1 Datos generales

Nombre del curso:	Arquitectura de Computadores I
Código:	CE-4301
Tipo de curso:	Teórico-Práctico
Electivo o no:	Obligatorio
Nº de créditos:	4
Nº horas de clase por semana:	4
Nº horas extraclase por semana:	8
% de las áreas curriculares:	Ciencias de Ingeniería (65%) - Diseño Ingeniería (35%).
Ubicación en el plan de estudios:	Curso del 7º semestre del Programa de Licenciatura en Ingeniería en Computadores.
Requisitos:	CE-3201 –Taller de Diseño Digital
Correquisitos:	Ninguno
El curso es requisito de:	CE-4302 Arquitectura de Computadores II
Asistencia:	Libre
Suficiencia:	Sí
Posibilidad de reconocimiento:	Sí
Vigencia del programa:	II Semestre de 2017

2 Descripción general

“Knowledge is power. Knowledge shared is power multiplied”

Robert Noyce

El diseño de computadores y aplicaciones basadas en computador requiere un conocimiento detallado de las capacidades y limitaciones de las herramientas de hardware, como de software. El curso de Arquitectura de Computadores I, pretende abordar el aspecto del perfil académico-profesional relacionado con el diseño de sistemas computacionales, desde una perspectiva conceptual que involucra elementos de abstracción para el diseño de procesadores y computadores integrados en chip, como es el caso del CPU, interfaz de memoria y dispositivos de entrada-salida. Se parte del conocimiento previo obtenido en los cursos “Diseño de Sistemas Digitales” (EL-3310) y “Taller de Diseño Digital” (CE-3201) y se profundiza en las estructuras que hacen posible el análisis, la implementación y la evaluación de un set de instrucciones particular, tomando en cuenta la arquitectura de la memoria, interfaces y protocolos de comunicación, técnicas de procesamiento paralelo, así como otra serie de características propias del diseño de computadores de aplicación general y específica, sentando las bases necesarias para el curso “Arquitectura de Computadores II” (CE-4302).

La metodología del curso busca además reforzar ciertas habilidades no técnicas como el trabajo en equipo y la creatividad, así como los valores de responsabilidad, respeto y tolerancia. El desarrollo de estas habilidades y valores permitirá una ejecución óptima de los diferentes proyectos de manera individual y grupal. Adicionalmente se espera reforzar la creación de documentación técnica como artículos científicos, informes y bitácoras de trabajo.

En este curso, se presenten desarrollar, a nivel intermedio (M), los atributos de: análisis de problemas (AP), ética y equidad (EE) y el uso de herramientas de ingeniería (HI), según estipulados por la agencia CEAB, correspondientes a los establecidos de acuerdo con el proceso de mapeo de atributos, de la carrera de Ingeniería en Computadores.

Se atenderá caso a caso cualquier necesidad educativa especial, con apoyo del Departamento de Orientación y Psicología y Trabajo Social y Salud de acuerdo a la particularidad de cada caso.

3 Objetivos

Objetivo(s) del curso	Atributo(s) correspondiente(s)	Nivel de desarrollo de cada atributo que se planea alcanzar: Inicial - I, intermedio - M o avanzado - A
1. Al completar el curso el estudiante será capaz de desarrollar un conjunto de herramientas de diseño de microprocesadores avanzados, por medio del estudio comprensivo de su estructura, interfaz y programación, con la finalidad de proponer soluciones a problemas de ingeniería, así como del desarrollo de habilidades de trabajo individual y en equipo, y la elaboración de documentación técnica, de manera individual y colaborativa, ordenada y concisa, haciendo uso de principios y valores como responsabilidad, respeto y tolerancia.	Análisis de problemas (AP)	M
	Uso de herramientas de ingeniería (HE)	M
	Ética y Equidad (EE)	M
2. Durante el curso, el estudiante desarrollará habilidades para el diseño y optimización de la arquitectura de un computador para un propósito determinado, basado en la comprensión de las características que determinan la arquitectura de un computador mediante el estudio de los bloques que posee un microprocesador de propósito general.	Análisis de Problemas (AP)	M
3. Durante el curso, el estudiante aplicará estrategias de optimización de tiempos de acceso y volumen de almacenamiento como criterios de diseño y optimización, basado en la comprensión de mecanismos de	Análisis de Problemas (AP)	M

administración de memoria utilizados en el diseño de computadores		
4. Al finalizar el curso, el estudiante estará en la capacidad de reconocer y comparar cuantitativamente las estrategias y algoritmos comúnmente utilizados en la industria de microprocesadores en cuanto al paralelismo de procesamiento a nivel de instrucción y a nivel de datos.	Uso de herramientas de ingeniería (HE)	M
	Análisis de Problemas (AP)	M
5. Al finalizar el curso, el estudiante estará en la capacidad de comparar las estrategias de control de flujo utilizadas en el diseño de microprocesadores para la optimización de tiempo de ejecución de instrucciones de un programa computacional.	Análisis de Problemas (AP)	M
6. Al finalizar el curso, el estudiante estará en la capacidad de comparar las características de las arquitecturas Load/Store contra las arquitecturas Registro/Memoria.	-	-
7. Durante el curso, el estudiante podrá discutir las estrategias de intercomunicación entre los distintos componentes de un computador, mediante una evaluación comparativa de las tecnologías utilizadas a nivel industrial.	Uso de herramientas de ingeniería (HE)	M

4 Contenidos

Introducción al diseño de computadores (2 semanas)

- Historia de los computadores
- Tendencias industriales
- Organización de Computadores vrs Arquitectura de computadores
- Ley de Amdahl
- Taxonomía de Flynn
- Métricas de desempeño

Diseño del set de instrucciones (2 semanas)

- CISC
- Von Newmann
- Harvard
- CISC vs RISC
- Arquitectura Load/Store
- Arquitectura Registro-Memoria
- Métricas de desempeño

Pipelining (3 semanas)

- Estrategias de segmentación y pipelining
- Riesgos estructurales
- Riesgo de dependencia de datos
- Riesgos de control

Otras técnicas de paralelismo (2 semanas)

- Arquitectura superescalares
- Predicción de saltos
- Ejecución fuera de orden
- Estructuras de unidades funcionales
- Calendarización estática y dinámica
- Introducción a VLIW

Diseño de la jerarquía de memoria (3 semanas)

- Jerarquía de memoria en un computador
- Memoria principal
- Memoria caché
- Coherencia de caché

- Caché avanzado
- Modo real y modo protegido
- Memoria virtual y paginación
- Traducción de direcciones

Interconexión de componentes en un computador (2 semanas)

- Estrategias de atención a dispositivos.
- Protocolos de comunicación de componentes.
- Casos de estudio: I2C, SPI, UART, PCIe.

Arquitecturas de aplicación específica (2 semanas)

- Diseño de computadores basados en microcontrolador.
- Diseño de computadores basados en DSP.
- Introducción a los sistemas empotrados de alto desempeño.
- Estudio de casos.

II parte: Aspectos operativos

5 Metodología de enseñanza y aprendizaje

En este curso, se plantean estrategias de aprendizaje activo, individual, social crítico y significativo, por medio de resolución y análisis de problemas, desarrollo de proyectos de diseño y optimización, tareas de investigación, entre otras técnicas.

Los contenidos del curso serán desarrollados por medio de clases magistrales teórico-prácticas, realizadas por el profesor. Durante las clases, el profesor presentará material teórico y demostraciones; los estudiantes realizarán ejercicios individuales y grupales, empleando herramientas de hardware y de software para tal fin. En periodo extraclase, el estudiante deberá desarrollar tareas teóricas y prácticas, de manera individual y grupal. Asimismo, el estudiante deberá desarrollar los proyectos en forma grupal y colaborativa. El contenido práctico se desarrolla paralelamente con el contenido teórico.

Los requerimientos de los proyectos, así como materiales adicionales serán ubicados en el TEC Digital. Cualquier dispositivo y material extra podrá ser dado en préstamo de acuerdo a los proyectos a desarrollar. Cada estudiante

y grupo de trabajo requerirá acceso a un computador de escritorio o portátil, con una distribución de GNU Linux, preferiblemente.

Otros instrumentos necesarios para el aprendizaje serán: laboratorio de computadoras, proyector, tarjetas de desarrollo FPGA, herramientas de simulación y perfilado como Altera Quartus, Simple Scalar, valgrind, oprofile, MARS MIPS ARMSim#, entre otras.

6 Evaluación

La evaluación del curso se basará en exámenes cortos (quices) individuales, tareas y exposiciones, individuales y en grupo, exámenes parciales individuales y proyectos grupales. El estudiante deberá desarrollar dos proyectos de diseño, en grupo, que podrán ser evaluados mediante la presentación funcional, un reporte escrito, una comprobación individual, una bitácora y presentación ante el grupo.

Se desarrollarán al menos 5 comprobaciones (quices) teóricos o prácticos, junto con tareas y exposiciones que complementen los conceptos desarrollados en la clase. Los quices se anunciarán con una clase de anticipación y cubrirán la materia vista en clase, así como lecturas complementarias asignadas por el profesor. Los puntos en este rubro se distribuirán equitativamente dependiendo del número de trabajos por ítem.

Dada la naturaleza práctica del curso, no se podrá optar por un examen de reposición.

En resumen, la evaluación se compone de:

Rubro	Valor (%)	Tiempo estimado entrega
I examen parcial	15%	Semanas 8
II examen parcial (2 partes)	30% (15%, 15%)	Semana 16 y 17
2 proyectos de diseño grupal	45% (25%, 20%)	Semana 7 y semana 15
Trabajos extra clase: tareas, lecturas complementarias, exposiciones y quices	10%	1 por semana de c/u

7 Bibliografía

Obligatoria:

- Hennesy, J. And Patterson, David. **Computer Architecture: A Quantitative Approach**. 5th Edition. Elsevier – Morgan Kaufmann. 2012.
- Amdahl. G.M. **Validity of the single-processor approach to achieving large scale computing capabilities**. In AFIPS Conference Proceedings, vol. 30 (Atlantic City, N.J.. Apr. 18-20). AFIPS Press, Reston. Va., 1967. pp. 483-485 (paper).
- Flynn, M. J., & Rudd, K. W. (1996). **Parallel architectures**. ACM Computing Surveys (CSUR), 28(1), 67-70 (paper).

Complementaria:

- Patterson,D.;Hennessy,J. **Computer Organization and Design: The Hardware/Software Interface**. 5th Edition. Elsevier - Morgan Kaufmann. 2013
- Shen, J and Lipasti, M; **Modern Processor Design: Fundamentals of Superscalar Processors**. 1st Edition. Tata McGraw-Hill. 2004.
- Stallings, W. **Computer Organization and Architecture Designing for Performance**. 8th Edition. Pretince Hall, 2010.
- Irvine, Kip; **Assembly Language for Intel-Based Computers**. 5th Edition. Pearson Education. 2007.
- Brey, B. **Microprocesadores Intel**. 8a Edición. Pearson. 2009.
- Abel, P. **Lenguaje ensamblador y programación para PC IBM y compatibles**. 3a Edición. Pearson. 1995.
- Rafiquzzaman, Mohamed. **Microprocessors and microcomputer based Systems design**. CRC press. 1990.
- Intel, **iAPX 86, 88, 186 and 188 User's Manual and Programmer's Reference**. Intel Corporation, Santa Clara, California, 1986.
- Tullsen, D., S. Eggers, and H. Levy. **Simultaneous Multithreading: Maximizing On-Chip Parallelism**. In The Proceedings of the 22rd Annual International Symposium on Computer Architecture, June 1995. (paper)

8 Profesor

Ing. Fabián Zamora Ramírez, Lic.

- Licenciado en Ingeniería en Computadores, TEC (2016)
- Ha trabajado por más de 4 años en la empresa Bounce Imaging, donde se ha desarrollado en múltiples áreas de la Ingeniería en Computadores incluidas la optimización de programación de sistemas embebidos, desarrollo de aplicaciones de escritorio y móviles enfocadas en la visualización omnidireccional de panoramas y transmisión de datos utilizando protocolos específicos.
- Como parte del equipo de Bounce Imaging participó en el “Launch Festival 2016”, el cuál se llevó a cabo en Silicon Valley (San Francisco, Estados Unidos) y fue ponente en el curso “Advances in Imaging” impartido en el Instituto Tecnológico de Massachusetts (MIT, 2017 via Skype).
- Es miembro fundador de la iniciativa GoTouch del Centro de Investigaciones en Computación del Tecnológico de Costa Rica y ha brindado diversos talleres sobre programación (Android, Java, Python) tanto a estudiantes como a egresados.
- Correo electrónico: fzamora@itcr.ac.cr
- Usuario de Skype: fabianz66
- Consulta presencial: Viernes 9:30am – 11:30am
- Lugar de consulta: Oficina 410 del edificio K1.
- Consulta virtual: Foro en TecDigital, correo electrónico, Skype, según disponibilidad