Ministerio de Cultura y Educación
Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
Departamento: Informatica
Area: Area II: Sistemas de Computacion

(Programa del año 2024)
(Programa en trámite de aprobación)
(Presentado el 09/09/2024 09:59:48)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
ARQUITECTURA DEL PROCESADOR I	ING. EN COMPUT.	28/12	2024	2° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
GROSSO, ALEJANDRO LEONARDO	Prof. Responsable	P.Asoc Exc	40 Hs

III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	**Teóricas**	**Prácticas de Aula**	**Práct. de lab/ camp/ Resid/ PIP, etc.**	**Total**
Hs	1 Hs	1 Hs	4 Hs	5 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	2º Cuatrimestre

Duración			
Desde	**Hasta**	**Cantidad de Semanas**	**Cantidad de Horas**
05/08/2024	15/11/2024	15	75

IV - Fundamentación

Con esta asignatura se inicia el estudio de las arquitecturas de procesadores para los alumnos de tercer año de la carrera de Ingeniería en Computación, tomando como punto de partida los procesadores secuenciales de tipo Von Neumann. El alumno será involucrado en:
*Problemas de representación de los datos que serán manipulados por el procesador; en cómo diseñar dispositivos físicos que realicen la memorización y manipulación de los datos.
*Comprender cuál es el conjunto de manipulaciones provistas habitualmente por los procesadores.
*Cómo se controla la realización de estas manipulaciones a medida que transcurre el tiempo.
Esto conformará la base previa para el entendimiento de arquitecturas más avanzadas que intentan mejorar el rendimiento de los procesadores secuenciales.

V - Objetivos / Resultados de Aprendizaje

*Aprender a representar datos y a manipularlos usando circuitos digitales.
*Comprender cómo están diseñados los procesadores secuenciales y cómo es su ciclo de instrucción.
*Desarrollar una actitud crítica frente al diseño de distintos procesadores.
*Obtener experiencia en programación de bajo nivel. Comprender cómo interactúan los procesadores con su medio externo

VI - Contenidos

PROGRAMA ANALÍTICO Y DE EXAMEN

Unidad N° 1: Sistemas Numéricos y Aritmética

Repaso de la representación de números negativos: representación signo y magnitud, sistemas de complementos: complemento a la raíz y complemento a la raíz disminuida, en exceso. Operación de complementación. Repaso de números fraccionarios: representación en punto fijo y punto flotante: rango de representación y operaciones aritméticas. Representación de caracteres. Código BCD, ASCII y EBCDIC.

Unidad N° 2: Circuitos Digitales

Circuitos combinacionales: semisumador, sumador completo, substractor completo, multiplexor y demutiplexor, decodificador. Especificación en VHDL de los circuitos: sumador, multiplexor, demultiplexor. Circuitos secuenciales: biestable, FF-SR, FF-D, registros, contadores ascendentes y descendentes módulo 2^n, registros desplazadores a izquierda y a derecha. Máquina de Moore. Máquina de Mealy.
Especificación en VHDL de una máquina secuencial de Mealy. Implementación de una fila y una pila.

Unidad N° 3: Organización Básica de una Memoria.

Organización de una memoria RAM (Random Access Memory). Construcción de una memoria de cuatro palabras con una memoria de dos palabras. Unidades de almacenamiento. Convenciones de numeración de bits y de bytes. Definición de contenido y continente de la información.

Unidad N° 4: Organización y Funcionamiento de una Unidad Central de Procesamiento.

Ciclo de instrucción. Organización básica de una máquina de primera generación: unidad de control, unidad aritmética y lógica, registros y buses internos. Instrucciones típicas. Fase de búsqueda de instrucción. Fase de búsqueda de operando. Fase de procesamiento de los datos. Fase de almacenamiento de resultado. Definición de Camino de datos (Data path). Camino de datos de la máquina de primera generación. Diseño de la unidad de control de una máquina de primera generación. Máquinas de segunda generación. Elementos de una instrucción de la máquina: Operación, frases de dirección. Instrucciones de tres direcciones, dos direcciones, una y cero dirección. tipos de operandos y tipos de operaciones. Modos de direccionamiento: conceptos generales. Modos de direccionamiento: registro, absoluto, inmediato, registro indirecto, post-incremento, pre-decremento. Modo de direccionamiento de múltiples componentes: indexado, relativos: a una base, a la próxima instrucción y a una página. Base indexado con desplazamiento. Operaciones: Manipulación de datos (manejo de datos, lógicas, aritméticas y relacionales), secuenciamiento, entrada-salida, supervisión.

Unidad N° 5: Lenguaje assembly y assembler

Máquinas de tercera generación. Máquinas Load-Store.
El procesador ARM: tipos de datos, conjunto de instrucciones, modos de direccionamiento, formato de instrucción. Lenguaje assembly: conceptos generales, sintaxis y pseudo operaciones. Assembler de dos pasadas. Operaciones en tiempo de ensamble, de carga y de ejecución. Implementación en lenguaje assembly de las estructuras de control: if-then, if-then-else, case, while y repeat. Acceso en lenguaje assembly a arreglos multidimensionales.
Máquinas de registros generales. El procesador i386: tipos de datos, conjunto de instrucciones, modos de direccionamiento. Mecanismo de segmentación y paginado. Llamadas a subrutina y retornos de subrutina. Uso de la pila. Pasaje de parámetros a las subrutinas: por valor, por dirección, por valor-retorno, por retorno. Formas de pasaje de parámetros: en registros y en la pila. Ejemplos en el procesador i386 y en el procesador ARM.

Unidad N° 6: Entrada/Salida

Organización de un sistema de computadora: unidad central de procesamiento (CPU), memoria, entrada/salida (I/O) y sistema de interconexión (buses). Módulos de entrada/salida: interfaz con la CPU e interfaz con el dispositivo. Protocolos de entrada/salida. Entrada/Salida programada. Organización de la entrada/salida: dedicada y embebida en memoria. Ejemplos de la entrada/salida en el procesador ARM.

Unidad N° 7: Interrupciones. Acceso directo a memoria (ARM)

Concepto de multiprogramación. Problemas presentados por la multiprogramación. Cambios de control necesarios para soportar multiprogramación: Interrupciones, despacho, llamadas al sistema. Condiciones asíncronas y síncronas. Atención de una interrupción asíncrona. Finalización de una interrupción síncrona. Manejo típico de una interrupción. Permiso para interrumpir: prioridades y deshabilitación de interrupciones. Acceso directo a memoria. Robo de ciclo.

VII - Plan de Trabajos Prácticos

Práctico N° 1: Sistemas numéricos

Representación en sistemas de complemento. Representación en complemento a la raíz y raíz disminuida. Representación binaria en exceso. Operaciones aritméticas en los distintos sistemas de representación. Detección de desborde en los distintos sistemas de representación. Representación de números en BCD. Código ASCII. Representación y aritmética de números representados en punto fijo y flotante, rangos de representación.

Práctico N° 2: Circuitos digitales

Circuitos sumadores, restadores, multiplexores y demutiplexores, decodificadores y codificadores.
Registros. Contadores. Desplazamiento. Decodificación de direcciones de memoria. Diseño de máquinas de estado finitas. Especificaci'on y simulación de circuitos digitales combinacionales utilizando el software abierto ghdl y gtkwave.

Práctico N° 3: Programación assembly en el ARM (inicial)

Características generales de la arquitectura del ARM. Organización de la memoria. Familiarización con set de instrucciones del ARM y pseudo instrucciones. Programas en lenguaje assembly simples. Instrucciones aritméticas y lógicas. Instrucciones de control condicional.Traducción de diferentes tipos de instrucciones. Calcular los desplazamientos que conforman algunas instrucciones y que son expresados en complemento a dos.

Práctico N° 4: Programación assembly en el DLX (modos de direccionamiento)

Realizar programas en los que se usen los distintos modos de direccionamiento: inmediato, registro, indexado y relativos: a una base y a la próxima instrucción.
Implementación de las estructuras de control if-then, if-then-else, case, while y repeat.

Práctico N° 5: Programación assembly en el i386 y ARM(subrutinas y pasaje de parámetros)

Instrucciones de llamada a subrutina y de retorno de subrutina. Manejo de la pila. Implementación en lenguaje assembly de las distintas formas de pasajes de parámetros: en registro y en la pila. Pasaje por valor, por dirección, por valor retorno y por retorno. Utilización del ensamblador "as" ("arm-none-eabi-as" para el ARM), el linker "ld" ("arm-none-eabi" para el ARM) y el depurador "gdb" ("arm-none-eabu-gdb" para el ARM) bajo linux para la programación de los ejercicios de aula.

Práctico N° 6: Programación assembly en el ARM (traducción a lenguaje de máquina)

Analizar los listados generados por el ensamblador. Traducción de diferentes tipos de instrucciones. Calcular los desplazamientos que conforman algunas instrucciones y que están representados en complemento a dos.

Práctico N° 7: Entrada / salida e interrupciones (ARM)

Protocolo entre los dispositivos para hacer una entrada/salida programada. Programas que realicen entrada/salida en forma programada. Cuestionario sobre conceptos generales de interrupciones. Programas que realicen entrada/salida con interrupciones.

VIII - Regimen de Aprobación

Regularización

Para regularizar la materia el alumno deberá cumplir con los siguientes requisitos:
Asistir al 80% de las clases teóricas y prácticas.
Aprobar dos exámenes parciales. Cada examen parcial tendrá dos recuperaciones.

Examen Final

Los alumnos regulares deberán rendir un examen final (que podrá ser oral o escrito) que consistirá en preguntas sobre los temas desarrollados durante el dictado de la materia.

Alumnos libres

Los alumnos que desean rendir libre la materia se deberán poner en contacto con la cátedra a los efectos de realizar un práctico, el cual contendrá ejercicios similares a los desarrollados en los prácticos durante el dictado de la materia. Aprobando éste trabajo práctico el alumno tendrá derecho a rendir un examen oral con iguales características que el de los alumnos regulares.

IX - Bibliografía Básica

[1] WILLIAM STALLINGS. Computer Organization and Architecture: Designing for Performance. ED. Pearson Prentice Hall. [2010].
[2] NIKLAUS WIRTH. Digital Circuit Design. An Introductory Textbook. Springer [1995].
[3] HAYES J. Computer Architecture and Organization. ED. MC. GRAW HILL [1988].
[4] JOHN F. WAKERLY. Microcomputer Architecture and Programming. Willey [1981].
[5] C. WILLIAM GEAR. Computer Organization and Programing. ED. MC. GRAW HILL [1980].
[6] WILLIAM STALLINGS. Computer Organization and Architecture. ED. MACMILLAN [1990].
[7] WILLIAM STALLINGS. Computer Organization and Architecture fourth edition. ED. PRENTICE HALL [1996].
[8] JOHN L. HENNESSY & DAVID PATTERSON. Computer Architecture: A Quantitative Approach. 2nd Edition. ED. MORGAN AND KAUFMANN [1990].
[9] JEAN-PIERRE MEINADIER. Estructura y Funcionamiento de los Computadores Digitales. Editorial AC, Madrid [1980].
[10] DAVID MAXINEZ; JESSICA ALCALÁ. VHDL El arte de programar sistemas digitales. Editorial CECSA, Mexico [2002].
[11] JHON CRAWFORD. Architecture of the Intel 80386. Readings in Computer Architecture, pag. 157-162. Edited by M. D. Hill, n. P. Jouppi and G. S. Sohi. ISBN 1-55860-539-8. [2000]
[12] Dandamudi Sivarama P.. Introduction to Assembly Language Programming, for Pentium and RISC Processors. Springer. ISBN 0-387-20636-1. [2004].
[13] Dandamudi Sivarama P.. Guide to Assembly Language Programming in Linux. Springer. ISBN 0-387-25897-3. [2005].

X - Bibliografia Complementaria

[1] GERRIT A. BLAAUW-FREDERICK P. BROOKS, Jr. Computer Architecture. Concepts and Evolution. ED. ADDISON-WESLEY. [1997].
[2] PETER J. ASHENDEN. The Designer's Guide to VHDL (Second Edition). ED. Morgan Kaufmann Publishers [2002].
[3] BEHROOZ PARHAMI Computer Arithmetic. Oxford University Press; 2da edición [2010].

XI - Resumen de Objetivos

*Aprender a representar datos y manipularlos usando circuitos digitales.
*Comprender cómo están diseñados los procesadores secuenciales y cómo es su ciclo de instrucción.
*Desarrollar una actitud crítica frente al diseño de distintos procesadores.
*Obtener experiencia en programación de bajo nivel.Comprender cómo interactúan los procesadores con su medio externo.

XII - Resumen del Programa

Representación y aritmética de números enteros y fraccionarios. Circuitos Digitales. Circuitos combinacionales básicos. Circuitos secueciales, Máquinas de estados finitas. Organización Básica de una Computadora. Unidad central de procesamiento (CPU). Memoria. Entrada / Salida. Organización interna de una CPU. Ciclo de instrucción. Conjunto de instrucciones de un procesador. Tipos de arquitecturas secuenciales. Lenguaje assembly, assembler y lenguaje de programación. Entrada / Salida. Interfaz con la CPU. Interfaz con los dispositivos. Interrupciones. Excepciones. Llamadas al sistema. Acceso directo a memoria.

XIII - Imprevistos

Comunicarse con la cátedra.
Correo electrónico: agrosso@unsl.edu.ar
Arquitectura del Procesador I.
Departamento de Informática.
Of. 25. Bloque II. 1er. Piso.
Facultad de Cs. Fisico, Matemáticas y Naturales.
Universidad Nacional de San Luis.
Ejercito de los Andes 950. CP 5700.

XIV - Otros

ELEVACIÓN y APROBACIÓN DE ESTE PROGRAMA	
	Profesor Responsable
Firma:	
Aclaración:	
Fecha:	