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 26/03/2024 19:22:37)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
ARQUITECTURA DEL PROCESADOR	ING. INFORM.	026/12-08/15	2024	1° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
GROSSO, ALEJANDRO LEONARDO	Prof. Responsable	P.Asoc Exc	40 Hs
ARROYUELO BILLIARDI, JORGE A.	Prof. Co-Responsable	P.Adj Exc	40 Hs
ARROYUELO, MONICA DEL VALLE	Responsable de Práctico	JTP Exc	40 Hs
CABALLERO, WALTER DAMIAN	Auxiliar de Práctico	A.1ra Semi	20 Hs

III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	**Teóricas**	**Prácticas de Aula**	**Práct. de lab/ camp/ Resid/ PIP, etc.**	**Total**
Hs	2 Hs	1 Hs	3 Hs	6 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	1º Cuatrimestre

Duración			
Desde	**Hasta**	**Cantidad de Semanas**	**Cantidad de Horas**
11/03/2024	21/06/2024	15	90

IV - Fundamentación

Durante varios años en la materia se han utilizado procesadores comerciales para la enseñanza de las diferentes arquitecturas de los procesadores.
La parte positiva de tal enfoque es que permite utilizar realizaciones concretas de tales arquitecturas sobre las cuales el alumno puede programar a muy bajo nivel y experimentar de forma directa la infraestructura necesaria para poder interpretar los algoritmos que en años posteriores implementarán en los distintos lenguajes de programación de alto nivel o en herramientas aún más abstractas que sirven para especificar qué hace un algoritmo para luego sintetizar cómo el algoritmo debe interpretarse sobre una arquitectura particular.

Nuestro enfoque es proponer una arquitectura de procesador que pueda extenderse imitando la manera en que las arquitecturas han ido evolucionando históricamente. En consecuencia se le propone al alumno la utilización de una arquitectura muy simple, inspirada en la MU0 (Manchester University versión 0), que únicamente utiliza el modo de direccionamiento absoluto. Sobre esta arquitectura el alumno puede experimentar los problemas que surgen cuando intentan implementar estructuras de datos como los vectores o pilas. Posteriormente, se le propone al alumno una extensión de la arquitectura que admite el modo absoluto indirecto. Utilizando dicha arquitectura experimentan que pueden resolver, de

forma más simple, problemas que utilizan vectores o pilas y además evitar que las instrucciones del programa se modifiquen durante la ejecución del mismo.

La siguiente propuesta de extensión es la incorporación del modo indexado, que no sólo evita que las instrucciones se tengan que modificar durante la ejecución del programa, sino que además permite que el código sea re-entrante (característica muy deseable para implementar sistemas operativos).
Por último se propone una arquitectura que incorpora los modos relativo al registro contador de programa (PC) y relativo a una base almacenada en un registro base. En ésta arquitectura final el alumno puede apreciar la especialización de los registros en registros de datos y registros de dirección y el uso diferenciado que se hace de ellos en los algoritmos que implementan.

El paso siguiente es estudiar el surgimiento de las arquitecturas de registros generales donde se elimina la especialización de los registros del procesador para finalmente experiementar sobre una arquitectura LOAD/STORE como la ultilizada por la plataforma EDU-CIAA que está basada en el procesador Cortex-M4 de ARM que utiliza el LPCxpreso 4337 del frabricante NXP. En esta última arquitectura los alumnos pueden apreciar cómo el diseñador restringe el conjunto de instrucciones y los modos de direccionamientos de la máquina a un mínimo posible sugerido principalmente por el conjunto de instrucciones que más frecuentemente usan los compiladores para traducir los lenguajes de alto nivel a lenguaje de máquina.

Para las extensiones propuestas se han desarrollado los ensambladores y los simuladores para que los alumnos puedan experimentar. Los simuladores utilizan una interfaz similar a la ofrecida por el depurador de código abierto gdb para el sistema operativo Linux.

Los ensambladores se desarrolaron utilizando las herramientas de software lex y bison que permiten generar un emsamblador o compilador a partir de la gramática del lenguaje a traducir.

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

CONTENIDOS MÍNIMOS

Sistemas Numéricos. 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 de ensamble, ensamblador y programación en lenguaje de ensamble. Entrada / Salida. Interfaz con la CPU. Interfaz con los dispositivos. Interrupciones. Excepciones. Llamadas al sistema.

Unidad N° 1: Sistemas Numéricos y Aritmética

Sistemas numéricos posicionales. Sistema binario, hexadecimal y octal. Cambios de base B a 10. Cambios de base 10 a base b. Conversión de b=r^k a base r y viceversa. Aritmética de números sin signo no fraccionarios. 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. Números fraccionarios: representación

en punto fijo y punto flotante: rango de representación y operaciones aritméticas. Representación de caracteres. Código ASCII.

Unidad N° 2: Circuitos Digitales

Señales analógicas versus señales digitales. Circuitos digitales. Álgebra de Boole. Propiedades del álgebra de Boole. Formalización de circuito digital. Síntesis de circuitos digitales. Compuertas Booleanas: AND, OR, NOT, XOR, NAND, NOR. Representación de circuitos digitales con circuitos Booleanos. Simplificación de circuitos. Utilización de un número restringido de operadores Booleanos. Circuitos combinacionales: semisumador, sumador completo, substractor completo, multiplexor y demutiplexor, decodificador. Circuitos secuenciales: biestable, FF-SR, FF-D, cerrojos (latches) versus registros, contadores ascendentes y descendentes módulo 2^n, registros desplazadores a izquierda y a derecha. Máquina de Moore. Máquina de Mealy. Grafo de estados. Definición formal. Su uso para formalizar un circuito secuencial.

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 (CPU), lenguaje de ensamble y de máquina.

Tipos de arquitecturas secuenciales. Ciclo de instrucción. Organización básica de una máquina con una arquitectura secuencial de primera generación: unidad de control, unidad aritmética y lógica, registros y buses internos, modo de direccionamiento absoluto directo e indirecto. 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 (datapath). Camino de datos de la máquina de primera generación. Diseño de la unidad de control de una máquina de primera generación.

Lenguaje de ensamble: conceptos generales, sintaxis y pseudo operaciones. Ensamblador de dos pasadas. Operaciones en tiempo de ensamble, de carga y de ejecución. Traducción a lenguaje de máquina para la arquitectura de primera generación.

Unidad N° 5: Evolución de las arquitecutas secuenciales. Modos de direccionamientos, lenguaje de ensamble y ensamblador

Tipos de Arquitectura secuencial: máquinas de segunda generación y tercera generación.

Modos de direccionamiento: conceptos generales. Modos: 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.

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.

Operaciones: Manipulación de datos (manejo de datos, lógicas, aritméticas y relacionales), secuenciamiento, entrada-salida, supervisión.

Implementación en lenguaje de ensamble de las estructuras de control: if-then, if-then-else, case, while y repeat.

Llamadas a subrutina y retornos de subrutina. Uso de la pila. Pasaje de parámetros a las subrutinas: por valor, por dirección. Formas de pasaje de parámetros: en registros y en la pila. Ejemplos en la arquitectura M0-extendida.

Unidad N° 6: Introducción a las arquitecturas LOAD/STORE. Entrada/Salida

Tipo de Arquitectura secuencial Load Store. El procesador ARM: tipos de datos, conjunto de instrucciones, modos de direccionamiento.

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 e inmersa en el espacio de direcciones de memoria. Ejemplos de entrada/salida programada en la arquitectura M0 (lectura de un teclado de membrana 4x4 teclas y escritura en un display de 4 dígitos de 8 segmentos). Programación de entrada/salida en lenguaje C sobre la plataforma EDU-CIAA. Análisis del código generado por el compilador para el ARM.

Unidad N° 7: Interrupciones y DMA (Direct Memory Access)

Concepto de multiprogramación. Problemas presentados por la multiprogramación. Cambios de control necesarios para soportar multiprogramación: Interrupciones, despacho y llamadas al sistema. Excepciones. 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. Ejemplos en lenguaje C sobre la platafomra EDU-CIAA.

VII - Plan de Trabajos Prácticos

Práctico N° 1: Sistemas numéricos

Representación de números en binario. Bases 2^k. Cambios de la base B a b. Cambios de base utilizando aritmética en la base B. Cambios de base utilizando artitmética en la base b. Representación en binario con bit de signo. 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

Compuertas Booleanas. Expresión Booleana. Tablas de verdad. Circuitos Booleanos. Expresiones Booleanas usando una sola conectiva. Simplificación de circuitos. Construccioón de: Circuitos sumadores, restadores, multiplexores y demutiplexores, decodificadores y codificadores. Biestables, flip-flop R-S, flip-flop D. Registros. Contadores. Desplazamiento. Decodificación de direcciones de memoria. Diseño de máquinas de estado finitas.

Práctico N° 3: Programación assembly en M0 (inicial)

Uso de la M0 para comprender la semántica de las instrucciones más usuales de un procesador. Instrucciones aritméticas y lógicas. Instrucciones de control condicional.Traducción de diferentes tipos de instrucciones a lenguaje de máquina de la M0.

Realizar la suma de los elementos de un vector de dimensión n usando la M0. Implementar las operaciones de una pila usando modo absoluto indirecto sobre la M0.

Práctico N° 4: Programación en lenguaje de ensamble de la M0 extendida.

Implementación en la M0 extendida de las estructuras de control if-then, if-then-else, case, while y repeat.
Uso del modo indexado y del modo registro indirecto para procesar vectores e implementar pilas.

Incorporación del modo base a la M0-extendida. Su uso en estructuras vinculadas. Definición de árboles binarios en forma estática mediante directivas al ensamblador. Recorridos sobre la estructura usando modo base.
Uso del modo relativo al PC y su utilidad para reubicar programas en la memoria de la máquina sin necesidad de usar un programa cargador que reubique el programa al momento de su carga en un origen distinto al que se usó cuando se ensambló dicho programa.
Calcular los desplazamientos que conforman algunas instrucciones y que están representados en complemento a dos.

Práctico N° 5:

Instrucciones de llamada a subrutina y de retorno de subrutina. Manejo de la pila. Implementación en lenguaje de ensamble del pasaje de parámetros en registros y en la pila. Pasaje por valor y por dirección. Uso de la pila para implementar subrutinas recursivas.

Práctico N° 6: Entrada-salida progrmada en la M0-extendida.

Realización de un programa que lea dos teclas desde un teclado de membrana de 4x4 teclas sume sus valores y muestre el resultado de la suma en hexadecimal usando un display de 4 dígitos con 8 segmentos c/u.

Implementación de los protocolos de E/S en forma progrmada en la M0-extentida.
Analizar los listados generados por el compilador C para la plataforma EDU_CIAA.
Entrada/Salida programada utilizando la EDU-CIAA para reproducir los tonos DTMF (Dual Tone Multi Frequency) mediante el conversor digital-analógico del LPCxpreso 4337.

Práctico N° 7: interrupciones y DMA (ARM)

Realizar el programa del práctico anterior utilizando 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 (de aula y/o laboratorio), 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] JHON F. WAKERLEY. Microcomputer Architecture and Programming. Ed. JOHN WILEY AND SONS [1981].
[3] NIKLAUS WIRTH. Digital Circuit Design. An Introductory Textbook. Springer [1995].
[4] HAYES J. Computer Architecture and Organization. ED. MC. GRAW HILL [1988].
[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] JOSEPHT YIU. The Definitive Guide to ARM Cortex-M3 and Cortex-M4 Processors. Third Edition. ISBN 13: 978-0-12-408082-9. [2014].
[11] GERRIT A. BLAAUW-FREDERICK P. BROOKS, Jr. Computer Architecture. Concepts and Evolution. ED. ADDISON-WESLEY. [1997].

X - Bibliografia Complementaria

[1] PETER J. ASHENDEN. The Designer's Guide to VHDL (Second Edition). ED. Morgan Kaufmann Publishers [2002].
[2] BEHROOZ PARHAMI Computer Arithmetic. Oxford University Press; 2da edición [2010].
[3] DAVID MAXINEZ; JESSICA ALCALÁ. VHDL El arte de programar sistemas digitales. Editorial CECSA, Mexico [2002].

XI - Resumen de Objetivos

*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.

XII - Resumen del Programa

Sistemas Numéricos. 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 de ensamble, ensamblador y programación en lenguaje de ensamble. 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. (arqui.unsl@gmail.com) (agrosso@email.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:	