



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 2023)

### I - Oferta Académica

Materia	Carrera	Plan	Año	Período
ARQUITECTURA DEL PROCESADOR I	LIC.CS.COMP.	32/12	2023	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	3 Hs	1 Hs	4 Hs	8 Hs

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

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
13/03/2023	24/06/2023	15	120

### IV - Fundamentación

Con esta asignatura se inicia el estudio de las arquitecturas de procesadores para los alumnos de segundo año de la carrera Lic. en Cs. de la Computación, tomando como punto de partida los procesadores secuenciales de tipo Von Neumann. El alumno estudiará y tendrá que resolver:

\*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. Dicha temática es objeto de la asignatura Arquitectura de Procesadores II.

### 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 secuenciales, 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 programación en lenguaje assembly. Entrada / Salida. Interfaz con la CPU. Interfaz con los dispositivos. Interrupciones. Excepciones. Llamadas al sistema. Acceso directo a memoria.**

### Unidad N° 1: Sistemas Numéricos y Aritmética

**Sistemas numéricos posicionales. Sistema binario, hexadecimal y octal. Cambios de base B a b. Cambios de base usando aritmética en base b. Cambios de base usando aritmética en 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 BCD, ASCII y EBCDIC.**

### Unidad N° 2: Circuitos Digitales

**Señales analógicas, 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 demultiplexor, 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.

**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. 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. Tipos de Arquitectura secuencial: 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**

**El procesador i386: 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. Tipos de Arquitectura secuencial: máquinas de tercera generación de registros generales y Load Store. El procesador i386 y el ARM: 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, en área asociada a la subrutina, en área asociada al llamador, en la pila. Ejemplos en el procesador i386.

#### **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 e inmersa en el espacio de direcciones de memoria. Ejemplos de la entrada/salida en el procesador i386 y el LPC4337.**

#### **Unidad N° 7: Interrupciones. Acceso directo a memoria**

**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. Acceso directo a memoria. Robo de ciclo.**

## **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 aritmética 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. Circuitos sumadores, restadores, multiplexores y demultiplexores, 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. Especificación y simulación de circuitos digitales combinacionales utilizando el software abierto ghdl y gtkview.

#### Práctico N° 3: Programación assembly en el i386 (inicial)

Características generales de la arquitectura del i386. Organización de la memoria. Familiarización con el set de instrucciones del i386 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 i386 (modos de direccionamiento)

Realizar programas en los que se usen los distintos modos de direccionamiento: inmediato, registro, registro indirecto y indexado, relativos: a una base, a la próxima instrucción.

Implementación de las estructuras de control if-then, if-then-else, case, while y repeat. Implementación de arreglos multidimensionales. Utilización del ensamblador "as", el linker "ld" y el depurador "gdb" bajo linux para la programación de los ejercicios de aula.

#### Práctico N° 5: Programación assembly en el ARM (modos de direccionamiento)

Realizar programas en los que se usen los distintos modos de direccionamiento: inmediato, registro, indexado, relativos: a una base, a la próxima instrucción.

Utilización del ensamblador "arm-none-eabi-as", el linker "arm-none-eabi-ld" y el depurador "arm-none-eabi-gdb" bajo linux para la programación de los ejercicios de aula.

#### Práctico N° 6: Programación assembly en el i386 (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, en área asociada al llamador, en área asociada a la rutina, en línea, en la pila. Pasaje por valor, por dirección, por valor retorno y por retorno. Utilización del ensamblador "as", el linker "ld" y el depurador "gdb" bajo linux para la programación de los ejercicios de aula.

#### Práctico N° 7: 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 son expresados 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 (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] 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].
- [14] GERRIT A. BLAAUW-FREDERICK P. BROOKS, Jr. Computer Architecture. Concepts and Evolution. ED. ADDISON-WESLEY. [1997].

## X - Bibliografía 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].

## 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

### **XIII - Imprevistos**

Comunicarse con la cátedra.  
Arquitectura del Procesador I.  
Departamento de Informática.  
Of. 25. Bloque II. 1er. Piso.  
Facultad de Cs. Físico, Matemáticas y Naturales.  
Universidad Nacional de San Luis.  
Ejército de los Andes 950. CP 5700

### **XIV - Otros**

--