Ministerio de Cultura y Educación **(Programa del año 2025)**
Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
Departamento: Informatica
Area: Area II: Sistemas de Computacion

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
ARQUITECTURA DEL PROCESADOR I	LIC.CS.COMP.	RD-3-1/2023	2025	1° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
GROSSO, ALEJANDRO LEONARDO	Prof. Responsable	P.Asoc 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	1º Cuatrimestre

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
12/03/2025	17/06/2025	15	120

IV - Fundamentación

Con esta asignatura se inicia el estudio de las arquitecturas de procesadores para los alumnos de segundo año de la Lic. en Cs. de la Computación, tomando como punto de partida la diferenciación clara entre los conceptos: arquitectura, implementación y realización de un procesador. El alumno comienza con una arquitectura muy básica, experimenta con una posible implementación de tipo Von Neumann (La llamamos máquinas de primera generación con respecto a las arquitecturas y no a la tecnología utilazada para su realización) y la realización de algunas partes pequeñas de dicha implementación.

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.

Finalmente el alumno estudia procesadores comerciales que se encuentran en los sistemas de computación más utilizados y experimenta sobre su programación en bajo nivel implementando algunas estructuras de datos (arreglos n-dimensionales y listas vinculadas) y estructuras de control (secuencia, selección, iteración), subrutinas, pasaje de parámetros a subrutinas, entrada/salida, llamadas al sistema operativo e interrupciones.

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.

El alumno estudia y resuelve:
*Problemas de representación de los datos que son manipulados por el procesador.
*Cómo diseñar dispositivos físicos que realicen la memorización, transferencia 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 implementaciones más avanzadas que intentan mejorar el rendimiento de los procesadores secuenciales o que intentan mejorar las limitaciones estructurales de los procesadores de tipo Von Neumann que utilizan una memoria lineal (Debido a que tales procesadores están basados en la máquina de Turing) o limitaciones sobre el secuenciamiento de las instrucciones. 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 implementados los procesadores secuenciales y cómo es su ciclo de instrucción.
- Desarrollar una actitud crítica frente a la implementación de distintos procesadores.
- Obtener experiencia en programación de bajo nivel. Comprender cómo interactúan los procesadores con su medio externo.
- Aprender la relación entre la arquitectura de los procesadorers y los lenguajes de programación.
- Identificar, formular y resolver problemas de informática.
- Concebir, diseñar y desarrollar proyectos de informática.
- Utilizar técnicas y herramientas de aplicación en la informática.
- Generar desarrollos tecnológicos y/o innovaciones tecnológicas.
- Auspiciar la comunicación efectiva.
- Fundamentar la acción ética y responsable.
- Aprender fundamentos para evaluar y actuar en relación con el impacto social de su actividad en el contexto global y local.
- Aprender nociones para el aprendizaje continuo.

VI - Contenidos

Representación de los datos a nivel máquina. Circuitos digitales: Señales analógicas vs. digitales. Funciones Booleanas como formalización de circuitos digitales. Circuitos combinacionales para la implementación de operaciones lógicas y aritméticas. Circuitos secuenciales, realización de elementos de almacenamiento e interpretación de instrucciones de un procesador. Estructura y funcionamiento de un sistema de computadoras: Diferencia entre arquitectura, implementación y realización. Unidad de procesamiento central (CPU). Almacenamiento principal. Subsistema de entrada/ salida. Interconexión. “Arquitectura Von Neumann” y “Harvard”. Evolución de las arquitecturas de

procesadores. Lenguaje de ensamble y lenguaje de máquina. Programación de bajo nivel. Entrada/salida: Organización de la entrada/salida y sus consecuencias sobre la arquitectura del procesador. Protocolo de entrada y salida. Entrada/salida programada y solapada. Interrupciones: Multiprogramación, administración dinámica de memoria, protección y seguridad sobre la memoria compartida, cambios de control (interrupciones, llamadas al sistema y despacho de procesos). Entrada/salida con interrupciones.

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 coma fija y coma flotante: rango de representación y operaciones aritméticas. Representación de caracteres. Código BCD, ASCII y EBCDIC.

Unidad N° 2: Circuitos Digitales

Diferencia entre señales analógicas y 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. Simulación de la 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. Simlación de la especificación en VHDL de una máquina secuencial de Mealy.

Unidad N° 3: Organización Básica de una Memoria.

Organización de una memoria RAM (Random Access Memory). Construcción de una memoria de cuatro palabras a partir de una memoria de dos palabras. Unidades de almacenamiento. Convenciones de numeración de bits y de bytes. Definición de contenido y continente de la información. Cómo se codifican los nombres de los continentes y cómo el contenido de los mismos. Ventajas que se logran, en cuanto a la protección y la adminstración de la memoria, cuando se pueden diferenciar los nombres de continentes del propio contenido de los mismos.

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 en memoria. Fase de procesamiento de los datos. Fase de almacenamiento en memoria del 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 y 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. Modos 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 de ensamble y programa ensamblador.

El procesador ARM-cortexM4: tipos de datos, conjunto de instrucciones, modos de direccionamiento, formato de instrucción. Lenguaje de ensamble: conceptos generales, sintaxis y pseudo operaciones. Programa ensamblador de dos pasadas. Operaciones en tiempo de ensamble, de carga y de ejecución. Implementación en lenguaje de ensamble de las estructuras de control: if-then, if-then-else, case, while y repeat. Acceso en lenguaje de ensamble a arreglos multidimensionales.

Máquinas de tercera generación. Máquinas de registros generales y Load Store. modos de direccionamiento. Utilización de la

técnica de segmentación y de paginado con el propósito de facilitar la protección y administración de la memoria. 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. Progamación en el microcontrolador LPC4337 multinúcleo asimétrico, compuesto por un núcleo ARM cortexM4 y el otro un cortexM0, utilizado por la plataforma abierta EDU-CIAA del proyecto Computadora Industrial Argentina Abierta desarrollada por un grupo de universidades nacionales entre otros.

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. Organización de la entrada/salida: dedicada e inmersa en el espacio de direcciones de memoria (memory mapped I/O). Protocolos de entrada/salida. Entrada/Salida programada. E/S solapada.
Ejemplos de la entrada/salida 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) y en el microcontralodor 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, 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 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. Con esta actividad evaluamos la capacidad del alumno de dentificar, formular y resolver problemas de informática, el uso de conocimientos adquiridos en la disciplina matemática aplicados a la informática y de aprender nociones para el aprendizaje continuo.

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 demutiplexores, decodificadores y codificadores. Biestables, flip-flop S-R, flip-flop D. Experimentación con algunos circuitos provistos por la cátedra que han sido realizados utilizando circuitos integrados CMOS basados en compuertas booleanas. Por ejemplo, utilización del flip-flop R-S para la eliminación de los rebotes de un pulsador mecánico.
Cerrojos (Latches). Registros. Contadores. Desplazamiento. Decodificación de direcciones de memoria. Diseño de máquinas de estado finitas. Esta actividad nos permite evaluar la capacidad del alumno para utilizar técnicas y herramientas de aplicación en la informática aprender nociones para el aprendizaje continuo.
Simulación de circuitos digitales combinacionales y secuenciales, utilizando especificaciones en VHDL provistas por la cátedra, y el software abierto ghdl y gtkview. Esta actividad al igual que la del párrafo anterior nos permite evaluar la capacidad del alumno de utilizar técnicas y herramientas de aplicación en la informática, y además de comunicación efectiva.

Práctico N° 3: Programación en lenguaje de ensamble de una arquitectura de primera generación (inicial).
Limitaciones de la arquitectura que sólo posee direccionamiento absoluto. Extensión de la máquina de primera generación a una de segunda generación. Resolución de problemas de programación sobre esta nueva arquitectura. Utilización de constantes inmediatas. Utilización en forma directa e indirecta de los modos registro, absoluto, indexado, relativos (a una base fija y a la próxima instrucción) , base-indexado con o sin desplazamiento y con o sin escala y/o factor. Con esta

actividad evaluamos la capacidad del alumno de identificar, formular y resolver problemas de informática como también la de concebir, diseñar y desarrollar proyectos de informática, y de aprender nociones para el aprendizaje continuo.

Características generales de la arquitectura del ARM (como ejemplo de una arquitectura de tercera generación). Organización de la memoria. Familiarización con el conjunto de instrucciones del ARM y pseudo instrucciones. Programación en lenguaje de ensamble del ARM. Instrucciones aritméticas y lógicas. Instrucciones de control condicional. Traducción de diferentes tipos de instrucciones. Cálculo de los desplazamientos que conforman algunas instrucciones y que están expresados en complemento a dos. Esta actividad nos permite evaluar la capacidad del alumno de utilizar técnicas y herramientas de aplicación en la informática.

Práctico N° 4: Programación en lenguaje de ensamble del ARM (modos de direccionamiento).
Realizar programas en los que se usen los distintos modos de direccionamiento: inmediato, registro, registro con escala, registro indirecto y indexado, base-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 ambiente para la programación de la EDU-CIAA basado en eclipse, provisto por el proyecto CIAA, bajo linux para la programación de los ejercicios de aula. Analizar los listados generados por el programa ensamblador. Traducción de diferentes tipos de instrucciones. Calcular los desplazamientos que conforman algunas instrucciones y que son expresados en complemento a dos. Con esta actividad evaluamos la capacidad del alumno de identificar, formular y resolver problemas de informática, aprender nociones para el aprendizaje continuo y su capacidad para utilizar técnicas y herramientas de aplicación en la informática

Práctico N° 5: Programación en lenguaje de ensamble del ARM (subrutinas y pasaje de parámetros)
Instrucciones de llamada a subrutina y de retorno de subrutina. Manejo de la pila. Subrutinas anidadas. Implementación de las distintas formas de pasajes de parámetros: en registro, en área asociada al llamador, en área asociada a la subrutina, en línea, en la pila. Pasaje por valor, por dirección, por valor retorno y por retorno. Con esta actividad evaluamos la capacidad del alumno de dentificar, formular y resolver problemas de informática y de aprender nociones para el aprendizaje continuo.

Práctico N° 6: Entrada/salida e interrupciones (ARM)
Protocolo entre los dispositivos para hacer una entrada/salida programada. Programas que realicen entrada/salida por ejemplo utilizando los GPIO (General Purpose Input/Output), y entada/salida en forma programada mediante los protocolos para la utilización del conversor D/A del microcontrolador LPC4337 de la EDU-CIAA.
Integración de las rutinas de E/S anteriormente desarrolladas en lenguaje de ensamble con un programa en lenguaje C para la reproducción de los bitonos “DTMF” (Dual-tone multi-frequency) de las 16 teclas de un teléfono digital. Idem al párrafo anterior pero utilizando interrpuciones para realizar la E/S. Cuestionario sobre conceptos generales de interrupciones.
Con esta actividad evaluamos la capacidad del alumno de dentificar, formular y resolver problemas de informática, el uso de conocimientos adquiridos en la disciplina matemática aplicados a la informática, de aprender nociones para el aprendizaje continuo. Utilizar técnicas y herramientas de aplicación en la informática. Generar desarrollos tecnológicos y/o innovaciones tecnológicas. Aprender fundamentos para evaluar y actuar tomando conciencia del aumento de la productividad del trabajo que se logra con el uso de la tecnología y el impacto social que produce en relación con su actividad sobre el empleo, los trabajadores y la competencia.

VIII - Regimen de Aprobación

Regularización
Para regularizar la materia el alumno debe cumplir con los siguientes requisitos:
Asistir al 70% de las clases teóricas, prácticas y de laboratorio.
Aprobar dos exámenes parciales. Cada examen parcial tiene dos recuperaciones. Este requisito ayuda a fundamentar la acción ética y responsable del alumno frente al proceso de evaluación.

Examen Final
Los alumnos regulares deben rendir un examen final (que puede ser oral o escrito) que consiste en preguntas sobre los temas desarrollados durante el dictado de la materia. Y forma parte de la evaluación de la comunicación efectiva de los temas desarrollados en la materia y fundamenta la acción ética y responsable del alumno frente al proceso de evaluación.

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 dicho trabajo práctico el alumno tiendrá derecho a rendir un examen oral con iguales características que el de los alumnos regulares.

IX - Bibliografía Básica

[1] [1] WILLIAM STALLINGS. Computer Organization and Architecture: Designing for Performance. ED. PEARSON PRENTICE HALL [2010].
[2] [2] JHON F. WAKERLEY. Microcomputer Architecture and Programming. Ed. JOHN WILEY AND SONS [1981].
[3] [3] NIKLAUS WIRTH. Digital Circuit Design. An Introductory Textbook. Springer [1995].
[4] [4] WILLIAM STALLINGS. Computer Organization and Architecture. ED. MACMILLAN [1990].
[5] [5] WILLIAM STALLINGS. Computer Organization and Architecture fourth edition. ED. PRENTICE HALL [1996].
[6] [6] JOHN L. HENNESSY & DAVID PATTERSON. Computer Architecture: A Quantitative Approach. 2nd Edition. ED. MORGAN AND KAUFMANN [1990].
[7] [7] JEAN-PIERRE MEINADIER. Estructura y Funcionamiento de los Computadores Digitales. Editorial AC, Madrid [1980].
[8] [8] DAVID MAXINEZ; JESSICA ALCALÁ. VHDL El arte de programar sistemas digitales. Editorial CECSA, Mexico [2002].
[9] [9] DANDAMUDI SIVARAMA P.. Introduction to Assembly Language Programming, for Pentium and RISC Processors. Springer. ISBN 0-387-20636-1. [2004].
[10] [10] 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.
*Dimensionar adecuadamente el rol de la ciencia y la técnica como método para la definición, implementación, realización de los procesadores y el uso de una jerarquía de abstracciones que tornan factible la realización de proyectos interdisciplinarios y extremadamente complejos, Sin dicha jerarquía sería imposible tener éxito en la construcción de los procesadores.

XII - Resumen del Programa

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. Organización interna de una CPU. Ciclo de instrucción. Conjunto de instrucciones de un procesador. Modos de direccionamiento. Clases de arquitecturas. Lenguaje de ensamble, programa ensamblador y programación en lenguaje de ensamble. Entrada/Salida. Interfaz con la CPU. Interfaz con los dispositivos. Interrupciones. 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