

**UNIVERSIDAD NACIONAL
DE LA MATANZA**

**DEPARTAMENTO DE INGENIERIA E
INVESTIGACIONES TECNOLOGICAS**

INGENIERIA EN INFORMATICA

**ARQUITECTURA DE
COMPUTADORAS (1109)**

Profesor Titular: Ing. Fernando I. Szklanny

**PLANIFICACIÓN Y GUIA DE TRABAJOS
PRACTICOS**

**AÑO ACADEMICO 2015
SEGUNDO Cuatrimestre**

OBJETIVOS DE LA MATERIA	4
PROGRAMA ANALÍTICO. CONTENIDOS TEÓRICOS Y PRÁCTICOS	5
BIBLIOGRAFIA	7
METODOLOGÍA DE ENSEÑANZA	9
EXPERIENCIAS DE LABORATORIO, TALLER O TRABAJOS DE CAMPO.....	9
USO DE COMPUTADORAS	10
METODOLOGÍA DE EVALUACIÓN	10
HORARIOS DE DICTADO.....	10
CALENDARIO DE ACTIVIDADES – AÑO 2015 - SEGUNDO CUATRIMESTRE.....	11
ANEXO AL REGLAMENTO DE PROMOCION	14
DOCENTES DE LA CÁTEDRA	16
GRUPO DE AFINIDAD	17
TRABAJO PRÁCTICO 0- REVISIÓN Y NIVELACIÓN.....	21
A.- REPRESENTACIÓN DE INFORMACIÓN BINARIA.....	21
B.- CÓDIGOS - APLICACIONES DE CÓDIGOS.....	22
C.- ÁLGEBRA DE BOOLE	23
TRABAJO PRACTICO 1: CARACTERÍSTICAS DE SEÑALES DIGITALES	24
TRABAJO PRACTICO 2: TECNOLOGÍA DE CIRCUITOS INTEGRADOS	25
TRABAJO PRÁCTICO 3: ARQUITECTURA DE COMPUTADORAS - ANÁLISIS FUNCIONAL - UNIDAD CENTRAL DE PROCESO	27
TRABAJO PRACTICO 4: INSTRUCCIONES Y MODOS DE DIRECCIONAMIENTO	29
TRABAJO PRÁCTICO 5: UNIDAD CENTRAL DE PROCESO: PROGRAMACIÓN EN LENGUAJE SIMBÓLICO.....	31
TRABAJO PRÁCTICO 6: UNIDAD CENTRAL DE PROCESO: PROGRAMACIÓN EN LENGUAJE SIMBÓLICO - SEGUNDA PARTE: – PROGRAMACIÓN AVANZADA.....	35
TRABAJO PRACTICO 7: UNIDAD DE MEMORIA.....	38
TRABAJO PRACTICO 8: UNIDADES DE ENTRADA SALIDA , PERIFÉRICOS Y COMUNICACIONES	41

OBJETIVOS DE LA MATERIA

La asignatura Arquitectura de Computadoras, que forma parte del segundo año de la currícula aprobada para la Ingeniería en Informática que se dicta en la Universidad Nacional de La Matanza, aparece planteada dentro del plan de estudios establecido por dicha currícula como una materia básica en lo que hace al análisis y estudio del hardware de computadoras, elaborando dicho estudio desde los bloques funcionales que configuran su estructura y llegando hasta el nivel básico de los elementos que a su vez conforman dichos bloques.

En este sentido, el alumno cursante de la materia Arquitectura de Computadoras deberá poseer, al momento de su ingreso a la materia, los conocimientos básicos referidos a los elementos fundamentales en la configuración de computadoras como lo son la representación de información dentro de las mismas y el manejo de información binaria en sus distintas formas, así como los conceptos básicos del álgebra booleana y los circuitos lógicos.

Estos conocimientos, adquiridos en las materias correlativas previas, permitirán el avance hacia las aplicaciones directas de los mismos, lo que, a su vez, permitirá que el alumno tenga, una vez completado el cursado de la misma, un panorama general sobre las estructuras de hardware, a partir de un análisis completo de una configuración clásica tipo Von Neumann, y una introducción a las arquitecturas modernas de computadoras, tema que se desarrollará en forma completa, a posteriori, en las materias correlativas a la presente.

Asimismo, el alumno recibirá una formación básica referida a programación de computadoras en lenguaje de máquina, la que, más allá de los lenguajes de alto nivel desarrollados en otras asignaturas, le permitirá tener un panorama introductorio al software de base y los sistemas operativos. Este punto será también desarrollado en forma completa en las materias correlativas.

Dado que, además, los alumnos de segundo año estarán cursando simultáneamente Física II, en la que se desarrollan temas vinculados con electricidad y algunos fundamentos de electrónica, la formación recibida en dicha materia permitirá elaborar algunos conocimientos sobre tecnologías de integración de circuitos digitales y los tipos de técnicas utilizadas en computadoras. Este desarrollo formará parte de la segunda mitad de la materia, en la que, como se ha dicho, se procederá al análisis de un sistema de computación, a partir de una estructura convencional, y analizando sus diferentes bloques funcionales, tanto desde el punto de vista del hardware en sí, como, en lo posible, en su relación con los demás bloques y con el usuario.

PROGRAMA ANALÍTICO. CONTENIDOS TEÓRICOS Y PRÁCTICOS

UNIDAD 1: TECNOLOGIA DE LOS CIRCUITOS DIGITALES

1.1.- Introducción a la tecnología de circuitos integrados. Concepto de circuito semiconductor. Diodo. Transistor. Principios de funcionamiento. El transistor como elemento de conmutación. Transistores bipolares y unipolares. Utilización en circuitos integrados. Familias de integración. Características dinámicas. Tiempos de propagación. Retardos. Características eléctricas. Corrientes. Tensiones. Aplicaciones prácticas.

UNIDAD 2: ARQUITECTURA DE COMPUTADORAS

2.1.- Arquitectura de computadoras. Conceptos de arquitectura y estructura. Funciones básicas de un computador. Relación entre las funciones requeridas y la arquitectura del sistema. La arquitectura Von Neumann. Descripción básica de las unidades funcionales. Unidad central de proceso. Buses. Unidad de control. Unidad aritmético lógica. Memoria central. Interfaces de entrada-salida. Arquitecturas no Von Neumann. Diferencias y características. Análisis de rendimientos. Formas de mejorar el rendimiento de un sistema computador.

2.2.- Unidad central de procesos. Estructura interna. Registros de la U.C.P. Registros de uso general y registros dedicados. Interconexión con los restantes bloques del sistema. Funcionamiento de la Unidad aritmético lógica. Descripción de funcionamiento de la unidad de control.

2.3.- Concepto de instrucción. Código de operación. Operandos. Instrucciones de doble operando y operando único. Tipos de instrucción. Vinculación del juego de instrucciones de un procesador con la estructura del mismo.

2.4.- Interconexión entre partes. Bus. Concepto. Bus síncrono y asíncrono. Protocolos de funcionamiento. Configuración maestro esclavo. Handshake. Otros protocolos.

2.5.- Ejecución de una instrucción por parte de la C.P.U. Ciclos de instrucción. Registros intervinientes. Ciclo de búsqueda y ejecución. Ejecución de un programa.

2.6.- Direccionamiento de memoria. Modos de direccionamiento. Direccionamiento directo e indirecto. Direccionamiento absoluto y relativo. Instrucciones que no acceden a memoria. Direccionamiento implícito.

UNIDAD 3: INTRODUCCION A LOS LENGUAJES DE PROGRAMACION

3.1.- Lenguaje de máquina. Estructura de un programa en lenguaje de máquina.

3.2.- Lenguaje simbólico. Ensamblador. Conversión a lenguaje de máquina. Códigos mnemotécnicos. Descripción de funcionamiento de un programa ensamblador.

3.3.- Lenguaje de alto nivel. Ventajas y aplicaciones. Programa compilador. Programa interprete. Diferencias entre lenguajes interpretados y compilados. Ventajas y desventajas de cada uno.

UNIDAD 4: UNIDAD DE MEMORIA

4.1.- La función de almacenamiento y su implementación. Jerarquías en las estructuras de memoria de un computador. Tecnologías de memorias aplicables a las distintas jerarquías. Clasificación de memorias. Tipos de memoria. Criterios para la clasificación.

4.2.- Memoria principal. Capacidad y estructura. Vinculación con los otros bloques de la computadora. Características circuitales. Configuraciones típicas. Consideraciones tecnológicas.

4.3.- Memoria caché. Capacidad y estructura. Fundamentos de su utilización. Vinculación con los otros bloques de la computadora. Características circuitales. Configuraciones típicas. Administración de la memoria cache. Técnicas de asignación. Técnicas de reemplazo. Influencia sobre el rendimiento de la CPU. Tiempo medio de acceso a memoria.

4.4.- Elementos para almacenamiento secundario. Características. Principios de funcionamiento.

UNIDAD 5: COMUNICACION CON EL MUNDO EXTERIOR

5.1.- Elementos o sistemas periféricos. Concepto. Interfaces de entrada-salida.

5.2.- Manejo de entrada-salida. Vinculación de la CPU con sus interfaces de entrada-salida. Control por programa. Control por interrupciones. Control por acceso directo a memoria.

5.3.- Priorización de periféricos. Criterios de priorización. Implementación. Prioridad establecida por hardware o software. Técnicas de implementación de interrupciones. Tiempo real.

5.4.- Periféricos estándar. Vinculación entre la interfaz de entrada salida y el periférico controlado. Distintos tipos de interfaz. Interfaces de comunicaciones. Comunicación serie. UART. Comunicación serie sincrónica y asincrónica. Velocidad de transmisión. Baud. Ventajas y desventajas de las comunicaciones serie respecto de la comunicación paralelo. Protocolos y normas de comunicación. USB.

5.5.- Interfaces dedicadas: controladores.

UNIDAD 6: INTRODUCCION A LAS ARQUITECTURAS MODERNAS.

6.1.- El concepto de arquitectura. Revisión de arquitecturas no Von Neumann. Introducción a las arquitecturas modernas. Arquitecturas CISC Arquitecturas tipo RISC: Características principales. Principios de funcionamiento. Comparación con las arquitecturas CISC: La unidad de procesos y la mejora del rendimiento,. Pipelining. Objeto de su existencia. Paralelismo de instrucciones. Otras arquitecturas de computadoras.-

BIBLIOGRAFIA

1.- Bibliografía básica

Murdocca – Heuring
Principios de Arquitectura de Computadoras
Pearson – Prentice Hall (2002)

Andrew Tanenbaum
Organización de computadoras.
Ed. Prentice Hall

William Stallings
Organización y arquitectura de computadoras
Prentice Hall

Hennesy – Patterson
Arquitectura de computadoras, un enfoque cuantitativo
Mc. Graw Hill

Patterson – Hennesy
Estructura y diseño de computadoras
Reverté

M. Morris Mano
Arquitectura de computadoras.
Ed. Prentice Hall

M. Morris Mano
Ingeniería Computacional: Diseño del Hardware
Ed. Prentice Hall

2.- Bibliografía de consulta

Heuring – Jordan
Computer systems design and architecture
Addison – Wesley

Shiva
Computer Design and Architecture
Marcel Dekker

V. Hamacher.
Organización de computadoras.
Ed. Prentice Hall

3.- Para referencia acerca de los procesadores utilizados.

Manuales de Usuario Motorola 6809

Manuales de usuario Motorola 68HC11

Manuales de usuario Motorola 68HC908

(Se consultan y se obtienen en versión pdf en www.freescale.com)

METODOLOGÍA DE ENSEÑANZA

La materia se desarrollará a través de clases teóricas, clases prácticas de resolución de ejercicios, clases de laboratorio con utilización de sistemas simuladores y emuladores, y un trabajo monográfico final sobre un tema relacionado con la materia, cuya presentación y aprobación será de carácter obligatorio.

Los trabajos prácticos se desarrollarán en la forma de problemas de aplicación y prácticos de laboratorio (incluyendo simulación).

En lo que hace a los trabajos prácticos relacionados específicamente con la arquitectura de computadoras, se podrán realizar en la forma de trabajos de investigación y presentación de monografías sobre temas específicos definidos por la cátedra.

Asimismo, y con relación al tema de programación y arquitectura de CPU, se realizarán trabajos prácticos de simulación en computadora, para poder visualizar el funcionamiento de un programa desarrollado en lenguaje de máquina.

No se requerirá presentación obligatoria de los trabajos prácticos de problemas, excepto por aquellos trabajos que sean específicamente indicados por los docentes a cargo de cada curso. Aquel alumno que desee presentar los trabajos prácticos para su revisión podrá hacerlo antes de las fechas establecidas para cada evaluación.

EXPERIENCIAS DE LABORATORIO, TALLER O TRABAJOS DE CAMPO

A efectos de complementar el aprendizaje de los diversos temas a desarrollar, se propone la realización de trabajos prácticos de laboratorio o de simulación sobre computadora, en los cuales puedan aplicarse los conceptos vistos en clases teóricas y prácticas. Al respecto, se requiere:

- Para los temas vinculados con circuitos lógicos, combinatorios y secuenciales, la implementación de trabajos prácticos de laboratorio (a definir), en los que se pueda verificar, vía instrumental electrónico convencional (osciloscopio, instrumentos de medición simples, frecuencímetro, etc.), el funcionamiento de dichos circuitos, lo que requerirá el previo armado de los mismos sobre plaquetas de experimentación o, preferentemente, sobre equipos del tipo de evaluación didáctica disponibles en forma comercial. Alternativamente, se podrán realizar prácticas de simulación en computadora, para lo cual se requerirá de un programa simulador de circuitos lógicos tipo Pspice o similar.
- Para los temas relacionados con los sistemas de computación, se requerirá la posibilidad de disponer de una máquina funcionando y desarmada para analizar los distintos bloques. En la misma se procederá a la verificación de características funcionales (velocidad de reloj, velocidad de bus, etc.) y circuitales. Asimismo, y con el objeto de desarrollar las prácticas referidas a programación, se requerirá la provisión de un assembler o crossassembler para procesador a determinar, con simulador que permita la visualización de la secuencia de ejecución de instrucciones a lo largo de un programa.

USO DE COMPUTADORAS

De acuerdo con lo especificado en el apartado anterior se prevé el uso de computadoras como elementos de aprendizaje, tanto a través de su utilización habitual, mediante la utilización de programas aplicativos orientados a los contenidos de la materia, como también en la forma de objeto de estudio, con el objetivo de observar y analizar su estructura hardware y la vinculación entre sus bloques.

METODOLOGÍA DE EVALUACIÓN

Se utilizará un proceso de evaluación continua, mediante el requisito de desarrollo y evaluación grupal o individual de cada uno de los trabajos prácticos a desarrollar en el curso. Dado el tiempo disponible, en las clases dedicadas a los trabajos prácticos, los docentes procederán al planteo de los objetivos de cada trabajo y orientarán al alumnado en cuanto a la metodología de realización. En las fechas fijadas para el vencimiento de aprobación de cada uno de los trabajos prácticos se procederá a una evaluación individual, oral o escrita, obligatoria, según el tipo de trabajo práctico de que se trate.

La misma se completa con las evaluaciones parciales y finales establecidas por el régimen de promoción vigente en la Universidad, el que a todos los efectos, se considera conocido por el alumno.

HORARIOS DE DICTADO

En el primer cuatrimestre del año 2015, la materia se dictará en los siguientes horarios:

TURNO MAÑANA: Miércoles, 8 – 12 hs, y Viernes, 10 a 12 hs.

Docentes: Ing. Hugo R. Tantignone, Ing. Carlos A. Rodríguez, Sr. Jair Hnatiuk

TURNO NOCHE: Martes, 19 a 23 hs, y Jueves, 19 a 21 hs.

Docentes. Ing. Roberto Di Lorenzo, Lic. Carlos E. Maidana, Lic. Jorge Fiter.

CALENDARIO DE ACTIVIDADES – AÑO 2015 - SEGUNDO CUATRIMESTRE

CLASE	SEMANA DEL	TEMA A DESARROLLAR – CLASES TEORICAS
1	17.8	INTRODUCCION A LA MATERIA – PRESENTACION DE CONTENIDOS
2	17.8	INTRODUCCIÓN A LAS TECNOLOGÍAS SEMICONDUCTORAS
3	24.8	INTRODUCCIÓN A LOS CIRCUITOS INTEGRADOS Y FAMILIAS LÓGICAS
4	24.8	INTRODUCCIÓN A LOS CIRCUITOS INTEGRADOS Y FAMILIAS LÓGICAS
5	31.8	ARQUITECTURA Y ORGANIZACIÓN DE UNA COMPUTADORA – DESCRIPCIÓN FUNCIONAL
6	31.8	UNIDAD ARITMÉTICO LOGICA – UNIDAD DE CONTROL.
7	7.9	UNIDAD DE CONTROL
8	7.9	UNIDAD DE CONTROL – DESCRIPCION DE FUNCIONAMIENTO - EJECUCIÓN DE INSTRUCCIONES
9	14.9	MODOS DE DIRECCIONAMIENTO
10	14.9	PROGRAMACIÓN EN LENGUAJE DE MAQUINA.
11	21.9	PROGRAMACIÓN EN LENGUAJE DE MAQUINA.
12	21.9	PROGRAMACIÓN EN LENGUAJE DE MAQUINA.
13	28.9	EJEMPLOS Y ALGORITMOS DE PROGRAMACION
14	28.9	EJEMPLOS Y ALGORITMOS DE PROGRAMACION
15	5.10	CONSULTAS Y REPASO
16	5.10	PRIMERA EVALUACION PARCIAL
17	12.10	CONFIGURACION DE SISTEMAS DE MEMORIAS – MEMORIA PRINCIPAL
18	12.10	MEMORIA CACHE
19	19.10	ESTRUCTURA DE ALMACENAMIENTO - RESUMEN
20	19.10	BLOQUE DE ENTRADA - SALIDA – CONFIGURACIÓN – INTERFACES DE ENTRADA SALIDA
21	26.10	CONFIGURACIÓN DE INTERFACES. MODELOS Y EJEMPLOS PRACTICOS DE UTILIZACIÓN HABITUAL.
22	26.10	ADMINISTRACIÓN DE PROCESOS DE ENTRADA – SALIDA – POR PROGRAMA – POR INTERRUPCIONES
23	2.11	ADMINISTRACIÓN DE PROCESOS DE ENTRADA – SALIDA POR INTERRUPCIONES – POR ACCESO DIRECTO A MEMORIA
24	2.11	COMUNICACION INTERFAZ PERIFÉRICO. COMUNICACIÓN SERIE Y PARALELO
25	9.11	COMUNICACION INTERFAZ PERIFÉRICO. COMUNICACIÓN SERIE Y PARALELO
26	9.11	
27	16.11	CONSULTAS Y REPASO
28	16.11	SEGUNDA EVALUACIÓN PARCIAL
29	23.11	ENTREGA DE NOTAS
30	23.11	CONSULTAS Y REPASOS
31	30.11	EVALUACION RECUPERATORIA,
32	30.11	ENTREGA DE NOTAS, CIERRE DEL CUATRIMESTRE

REGLAMENTO DE PROMOCIÓN

1.- La aprobación de la materia Arquitectura de Computadoras (1109) en el curso lectivo correspondiente al año 2015 se basará en las normas que se detallan a continuación, en tanto no se contradigan con disposiciones expresas de la Universidad.

2.- El dictado de la materia se dividirá en clases teóricas y de realización de trabajos prácticos, según calendario adjunto.

3.- En las clases de contenido teórico se desarrollarán los temas teóricos establecidos en el programa analítico adjunto. En las clases prácticas los alumnos, orientados por los docentes a cargo de las mismas, resolverán problemas de aplicación de los temas vistos en clase. Las guías de trabajos prácticos estarán a disposición de los alumnos con anterioridad a la fecha de realización de cada uno de los trabajos.

4.- Las clases de realización de trabajos prácticos se dedicarán a la resolución de los trabajos prácticos enunciados en el apartado correspondiente de la presente planificación. En cada comisión se dividirá al total de los alumnos en grupos de no más de cinco integrantes. Dichos grupos tendrán la obligación de entregar, a requerimiento del docente, aquellos trabajos prácticos que se definan como de entrega y aprobación obligatoria.

5.- La condición de alumno regular en la materia se obtendrá a través de:

- ❖ La aprobación de los trabajos prácticos, según lo detallado en el presente..
- ❖ La aprobación de **dos exámenes parciales** en las fechas y condiciones establecidas por los reglamentos vigentes en la Universidad.

6.- El calendario académico incluye **una instancia de evaluación recuperatoria**, la que podrán ser utilizada por el alumno para rendir nuevamente un parcial no aprobado. La fecha correspondiente figurará en el calendario académico de la asignatura. El alumno podrá utilizar esta fecha para recuperar el parcial no aprobado.

La condición de alumno regular, de acuerdo con lo establecido por la Universidad, deberá obtenerse en las fechas dispuestas por la Universidad. Aquel alumno que no haya logrado dicha condición en la fecha mencionada, figurará aplazado y deberá recurrir la materia.

7.- La aprobación de los dos exámenes parciales, cada uno de ellos con calificación igual o superior a 7 (siete) puntos, producirá la aprobación de la materia, por promoción directa sin examen final, la que se considerará aprobada con una calificación final igual al promedio de los parciales. Se hace notar que, de acuerdo con lo establecido en la reglamentación de la Universidad, la aprobación por promoción directa requiere que el alumno, al momento de la fecha límite establecida en el párrafo anterior, tenga aprobadas las materias correlativas previas.

Por consiguiente, aquellos alumnos que al momento del cierre de las actas de cursado adeuden las correlativas previas no tendrán acceso al régimen de promoción sin examen final. No se harán excepciones a esta condición.

8.- La obtención en los exámenes parciales (o sus recuperatorios) de calificación superior o igual a 4 puntos pero inferior a 7 puntos dejará la materia en condición de **cursada**. Para la aprobación definitiva de la materia, se requerirá del alumno la aprobación de un examen final, el que se rendirá en los turnos establecidos por la Universidad. La calificación mínima requerida para la aprobación de este examen final será de cuatro (4) puntos. El número de oportunidades en que se puede rendir este examen final está establecido en las reglamentaciones de la Universidad.

9.- La condición de aplazado en la instancia recuperatoria dejará la materia en condición de aplazada, lo que implicará la obligación de recurrar la misma, o rendirla en condición de alumno libre.

RESOLUCION 026/2001 - RÉGIMEN DE CURSADO Y APROBACIÓN

Parte pertinente de la Resolución aprobada el 11.05.2001

ANEXO I - Régimen de cursado y aprobación de asignaturas - Asignación de calificaciones.

Título I – Asignaturas de grado

1.- Se requiere una asistencia a clases no inferior al setenta y cinco por ciento (75%). El incumplimiento de este requisito coloca al alumno en condición de **“ausente”**.

2.- Las asignaturas se aprueban por régimen de promoción por exámenes parciales y recuperatorios.

2.a.- Para las asignaturas de cursado semestral habrá dos (2) evaluaciones parciales y una (1) instancia recuperatoria salvo las detalladas en el Anexo II del presente régimen para las cuales, por razones de índole pedagógica, se aprecia necesario una instancia recuperatoria adicional, por lo que debe entenderse que, para tales asignaturas, habrá dos (2) evaluaciones parciales y dos (2) instancias recuperatorias. **En las asignaturas de cursado semestral, se entiende “ausente” el alumno que no obtenga calificación alguna en una (1) de las instancias de evaluación parcial (y su correspondiente recuperatorio)**

Para las asignaturas de cursado anual habrá al menos dos (2) evaluaciones parciales y las instancias recuperatorias serán tres (3), pudiendo aplicar hasta dos (2) instancias recuperatorias para uno (1) de los exámenes parciales. En estas asignaturas se entiende “ausente” el alumno que no obtenga calificación alguna en 2 (dos) de las instancias de evaluación parcial.

2.b.- Los exámenes parciales (y sus recuperatorios) se entenderán **“aprobados”** cuando la calificación asignada, en una escala de 0 a 10 puntos, resulte:

- Superior o igual a 6 (seis) puntos para las asignaturas correspondientes a las carreras residentes en el Departamento de Ciencias Económicas.
- Superior o igual a 7 (siete) puntos para las asignaturas correspondientes a las carreras residentes en los Departamentos de Ingeniería e Investigaciones Tecnológicas y de Humanidades y Ciencias Sociales.

2.c.- Las asignaturas se entenderán “aprobadas” cuando se aprueben todos los exámenes parciales (en primera instancia o por recuperatorio). La calificación asignada al examen recuperatorio (cualquiera sea el resultado) anula y reemplaza, a todos los efectos, a la obtenida en el examen parcial que se recupera. La calificación final se calculará como promedio de los exámenes parciales rendidos y aprobados. Para las asignaturas detalladas en el Anexo III del presente régimen, en las que el último examen parcial se establece como integrador del total de los aspectos a evaluar, la calificación final se calculará como promedio de los exámenes rendidos y no aplazados, siempre que el parcial integrador resulte aprobado.

2.d.- De esta manera la calificación final necesaria para que la asignatura resulte **“aprobada”** será:

- Superior o igual a 6 (seis) puntos para las asignaturas correspondientes a las carreras residentes en el Departamento de Ciencias Económicas.
- Superior o igual a 7 (siete) puntos para las asignaturas correspondientes a las carreras residentes en los Departamentos de Ingeniería e Investigaciones Tecnológicas y de Humanidades y Ciencias Sociales.

3.- Los exámenes parciales (y sus recuperatorios) calificados con tres (3) o menos puntos se entenderán "**aplazados**" y podrán ser recuperados.

3.a.- Las asignaturas semestrales en las que ocurran dos (2) aplazos en los exámenes (parciales y/o recuperatorios) se entenderán "aplazadas**" y deberán ser recusadas.**

3.b.- Las asignaturas anuales en las que ocurran tres (3) aplazos en los exámenes (parciales y/o recuperatorios) se entenderán "**aplazadas**" y deberán ser recusadas.

4.- Los exámenes parciales (y sus recuperatorios) calificados con:

- Cuatro (4) o cinco (5) puntos para las asignaturas correspondientes a las carreras residentes en el Departamento de Ciencias Económicas.
- Cuatro (4), cinco (5) o seis (6) puntos para las asignaturas correspondientes a las carreras residentes en los Departamentos de Ingeniería e Investigaciones Tecnológicas y de Humanidades y Ciencias Sociales,

Se entenderán "**desaprobados**" y podrán ser recuperados.

4.a.- Las asignaturas con calificación final, calculada como promedio de los exámenes parciales (o el recuperatorio correspondiente) rendidos y no aplazados de:

- Cuatro (4) o cinco (5) puntos para las asignaturas correspondientes a las carreras residentes en el Departamento de Ciencias Económicas.
- Cuatro (4), cinco (5) o seis (6) puntos para las asignaturas correspondientes a las carreras residentes en los Departamentos de Ingeniería e Investigaciones Tecnológicas y de Humanidades y Ciencias Sociales,

se entenderán "**cursadas**" y podrán ser aprobadas por examen final. La calificación necesaria para aprobar el examen final será de cuatro (4) o más puntos.

4.b.- La validez de la asignatura "cursada" será de:

- Tres (3) turnos consecutivos de examen final para las asignaturas correspondientes a las carreras residentes en el Departamento de Ciencias Económicas.
- Cinco (5) turnos consecutivos de examen final para las asignaturas correspondientes a las carreras residentes en los Departamentos de Ingeniería e Investigaciones Tecnológicas y de Humanidades y Ciencias Sociales.

Dichos turnos serán contados a partir del turno inmediato siguiente al periodo de cursado. A esos fines, la situación académica del alumno deberá quedar definida al último día hábil del mes de julio (para las asignaturas que se cursen en el primer cuatrimestre) y al último día hábil de la primer semana de diciembre (para las asignaturas anuales y las que se cursen en el segundo cuatrimestre. Extinguida la validez de "cursada" la asignatura deberá cursarse nuevamente.

5.- En aquellas asignaturas que reglamentariamente requieran la aprobación de Trabajos Prácticos, dicha aprobación se entiende como requisito para rendir los exámenes parciales y los recuperatorios correspondientes.

-

DOCENTES DE LA CÁTEDRA

Ing. Fernando I. Szklanny	Jefe de Cátedra fszklanny@unlam.edu.ar
Lic. Carlos E. Maidana	Profesor Asociado
Ing. Roberto Di Lorenzo	Profesor Adjunto
Lic. Jorge Fiter	Jefe de Trabajos Prácticos
Ing. Carlos Rodríguez	Jefe de Trabajos Prácticos
Ing. Hugo Tantignone	Jefe de Trabajos Prácticos
Ing. Edgardo Gho	Ayudante de Primera
Sr. Jaír Hnatiuk	Ayudante Alumno
Sr. Martín Ferreyra Birón	Ayudante Alumno

GRUPO DE AFINIDAD

El material necesario para el desarrollo de la materia se encuentra disponible en el sitio que se indica a continuación.

<http://groups.yahoo.com/group/acunlam/>

Dado que se trata de un grupo exclusivo para la cátedra, los alumnos deberán registrarse en dicho grupo, solicitando la autorización de acceso al coordinador del grupo. La autorización de acceso se obtiene mediante el envío de los datos personales: apellido y nombre, número de documento, curso en el que cursa.

**UNIVERSIDAD NACIONAL DE LA
MATANZA**

**DEPARTAMENTO DE INGENIERIA E
INVESTIGACIONES TECNOLOGICAS**

INGENIERIA EN INFORMATICA

**ARQUITECTURA DE COMPUTADORAS
(1109)**

**Profesor Titular:
Ing. Fernando I. Szklanny**

GUIA DE TRABAJOS PRACTICOS

AÑO ACADEMICO 2015

TRABAJO PRÁCTICO 0- Revisión y Nivelación

a.- Representación de información binaria.

0.01.- Definir qué se entiende por **señal analógica**, **señal digital**, **sistema analógico** y **sistema digital**.

0.02.- Definir los siguientes conceptos:

Sistema numérico.

Sistema numérico posicional.

Base de un sistema posicional.

Sistemas numéricos: Decimal, octal, binario, hexadecimal

Bit, byte y word.

Indicar cuál es la expresión general que permite obtener la cantidad de combinaciones o palabras posibles de n bits.

0.03.- Hallar el complemento a la base y el complemento a la base menos uno de los siguientes números aplicando la definición, según el formato de cantidad de bits que se indica.

00000110₂ (8 bits)

11100110₂ (8 bits)

2A78₁₆ (4 dígitos)

1202₃ (4

0.04.- Indicar cuál es el mayor número positivo y cuál es el menor negativo que se puede expresar con un byte, destinando el primero a la izquierda como bit de signo y almacenando los negativos en:

Signo y módulo

Signo y complemento a la base menos uno

Signo y complemento a la base

Nota :

Escribir los números binarios y los equivalentes en decimal correspondientes.

0.05.- Considerar los números de 8 bits A = 01011001 y B = 10110100. Realizar A + B en binario e interpretar los operandos y resultados en decimal suponiendo que se trata de:

- números sin signo (inherentemente positivos)

- números con signo y almacenando los negativos en cada una de las formas de representación de enteros (signo y magnitud, signo y complemento a la base, signo y complemento a la base menos uno).

0.06.- Representar el número decimal 33.75 en las normas IBM 370 e IEEE de simple precisión:

0.07.- Indicar qué número decimal representa el número binario 1001 1111 1001 1101 0000 0000 0000 0000, si se trata de un valor expresado en punto flotante norma IEEE.

0.08.- Repita el ejercicio anterior considerando que el número binario se encuentra expresado en la norma IBM 370 simple precisión.

b.- Códigos - Aplicaciones de Códigos

0.08.- Representar los siguientes números decimales en binario puro, BCD 8421, BCD Exceso 3, BCD 2421 y BCD 5421:

3975 0010 56173 63
Extraer conclusiones.

0.09.- Generar un código progresivo y uno progresivo cerrado que represente a los dígitos decimales. Ídem para un código reflejado.

0.10.- Dados los números decimales 342, 45 y 2513 , convertirlos a sistema binario, codificarlos en BCD 8421 y codificarlos en ASCII (8 bits).

0.11.- Efectuar las siguientes sumas, codificando previamente los números en BCD 8421, indicar cuáles son las correcciones a realizar y justificarlas. :

731 + 431 1162 + 895 1923 + 8931

0.12.- Definir y ejemplificar

- distancia entre dos palabras de un código
- distancia de un código

0.13.- Indicar cuál es la distancia del código BCD 8421, justificar y explicar por qué no resulta un buen código para la transmisión de información.

0.14.- Agregar el bit de paridad adecuado a las siguientes palabras, suponiendo que se trabaja con paridad par en los unos:

10100100 10111110 11110101

0.15.- En las aplicaciones de transmisión de la información, indicar cuál es la relación entre la distancia del código utilizado y la cantidad de bits detectados con error y corregidos en el receptor.

0.16.- Obtener un código de Hamming corrector de errores de un bit a partir del código BCD Exceso 3.

0.17.- Se quiere transmitir la palabra "TERA" en un código que permita corregir errores. Indicar :

la codificación previa al agregado de los bits de paridad, en ASCII.
la codificación a transmitir, utilizando el método de Hamming.

0.18.- Se recibe la siguiente secuencia, codificada en ASCII extendido con posibilidad de corrección de errores mediante el método de Hamming:

000110011000 100010001011 110110001100 100010010001

Verificar si hay errores en la recepción. En caso afirmativo, sabiendo que como máximo el canal de transmisión puede provocar error en un bit, efectuar la corrección. Determinar, si es posible, qué se transmitió.

c.- Álgebra de Boole

0.20.- Explicar la utilización del Álgebra de Boole en el tratamiento de sistemas digitales.

0.21.- A partir de los postulados de Huntington, simplificar las siguientes expresiones booleanas e implementarlas:

$$f = a.(b + \bar{b}) + a.a(b + b) + a.(b + 1).c.\bar{c}$$

$$f = a.b.\bar{c} + a.b.c + \bar{a}.b.c + a\bar{b}$$

0.22.- Escribir la expresión booleana correspondiente a las funciones dadas en las siguientes tablas de verdad en sus dos formas canónicas, simplificarlas e implementarlas con las compuertas más adecuadas.

abcd f1

0000	0
0001	1
0010	0
0011	1
0100	1
0101	0
0110	1
0111	0
1000	1
1001	1
1010	0
1011	1
1100	1
1101	0
1110	0
1111	1

abcd f2

0000	1
0001	0
0010	1
0011	1
0100	1
0101	0
0110	1
0111	1
1000	1
1001	1
1010	1
1011	1
1100	1
1101	0
1110	1
1111	1

TRABAJO PRACTICO 1: Características de Señales Digitales

1.01.- Digitalizar las siguientes funciones

- $f(x) = x^2 + 5x + 2$, en el intervalo $[0, 2]$, con 16 niveles. Asignar un código binario a cada nivel.
- $f(t) = \sin(2\pi t)$ en el intervalo $[0, 1]$, con una resolución de 6 bits. Indicar la cantidad de niveles.

Graficar ambas funciones en una escala adecuada.

1.02.- Dibujar un pulso no ideal, definir sus características e indicar las unidades de medición de los siguientes elementos:

- amplitud
- ancho del pulso (t_w)
- nivel alto (h , high)
- nivel bajo (l , low)
- flanco ascendente o positivo
- flanco descendente o negativo
- tiempo de subida (t_r , rise time)
- tiempo de caída (t_f fall time)
- ciclo de trabajo (duty cycle)

1.03.- Explicar los conceptos de

- función periódica
- frecuencia de una función periódica
- período de una función periódica
- amplitud de una función periódica

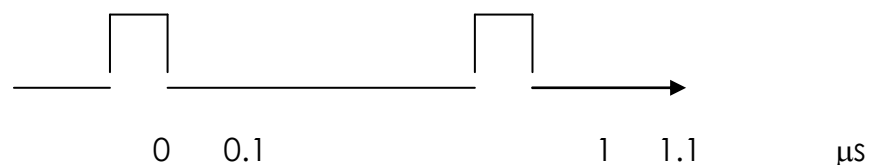
Ejemplificar con una función periódica continua y una función periódica discreta.

Establecer la relación matemática entre las magnitudes período y frecuencia.

Distinguir entre función matemática y señal física.

1.04.- Una señal digital periódica rectangular tiene un ancho de pulso de $25 \mu s$ y un período de $150 \mu s$. Calcular la frecuencia y el ciclo de trabajo.

1.05.- Dada la siguiente señal digital periódica con medidas expresadas en μs , determinar el período, la frecuencia y el ciclo de trabajo.



1.06.- Explicar qué es una señal de reloj (clock).

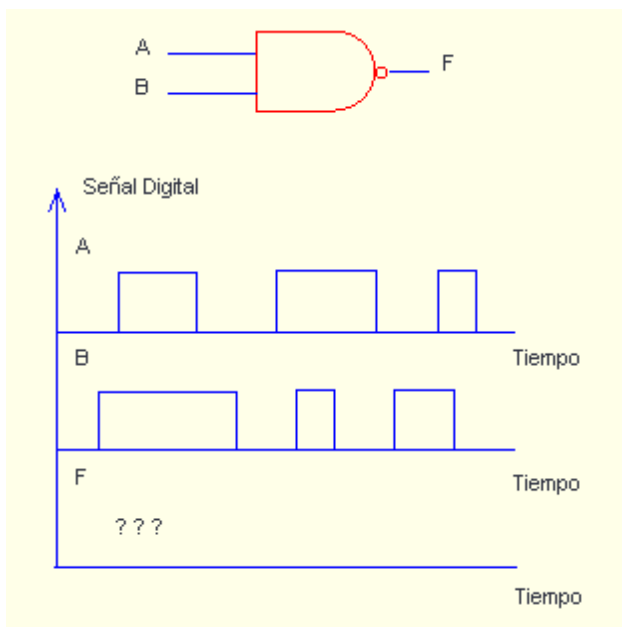
1.07.- Explicar cuál es el propósito de un diagrama de tiempos.

TRABAJO PRACTICO 2: Tecnología de Circuitos Integrados

2.01.- Para un circuito integrado digital, definir los siguientes parámetros:

- Tensión de alimentación
- Niveles lógicos
- Inmunidad al ruido
- Margen de ruido
- Disipación de potencia - Consumo
- Retardo de propagación o tiempo de propagación
- Factores de carga en la entrada y de salida (fan-in, fan-out)

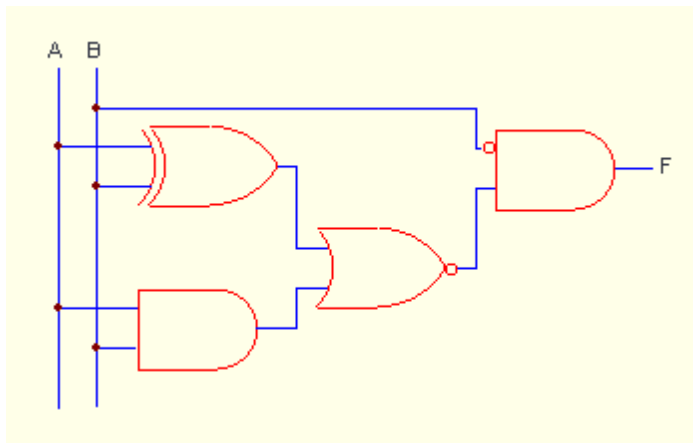
2.02.- Dibujar la forma de onda de salida para la siguiente compuerta, suponiendo que se trata de una tecnología de 5 ns de tiempo de retardo de propagación.



2.03.- Dadas dos compuertas inversoras, una de tecnología de 3 ns de tiempo de propagación y la otra, de tecnología de 10 ns, indicar cuál de las dos puede funcionar a mayor frecuencia y justificar.

2.04.- Para el siguiente circuito:

- Obtener la función de salida.
- Simplificar la función obtenida.
- Implementar la función simplificada.
- Hacer un diagrama de tiempos para los dos circuitos, para una entrada arbitraria, teniendo en cuenta un tiempo de retardo de propagación de 5 ns.
- Extraer conclusiones.



2.05.- Indicar las características de operación básicas de las familias TTL y CMOS. Tener en cuenta los parámetros definidos anteriormente. Hacer un cuadro comparativo de ventajas y desventajas.

2.06.- Hacer una clasificación de circuitos integrados de acuerdo a la cantidad de componentes integrados en el interior de un único chip.

2.07.- Por una determinada compuerta circulan $2\ \mu\text{A}$ cuando su salida esta en su nivel alto y $3.6\ \mu\text{A}$ cuando su salida está en su nivel bajo. Indicar cuál es la disipación media de potencia si la tensión de alimentación es $5\ \text{V}$ y la compuerta trabaja con un ciclo de trabajo del 50%.

2.08.- Determinar los márgenes de ruido de los niveles alto y bajo para una compuerta CMOS cuyos niveles lógicos son

$$\begin{aligned} V_{IHmin} &= 3,5\ \text{V} \\ V_{ILmax} &= 2,5\ \text{V} \\ V_{OHmin} &= 4,4\ \text{V} \\ V_{OLmax} &= 0,33\ \text{V} \end{aligned}$$

2.09.- Sabiendo que la misma compuerta, pero construida en tecnología TTL tiene una inmunidad al ruido de $0,4\ \text{V}$ para ambos estados, indicar qué tecnología debería utilizarse en un entorno de alto ruido.

2.10.- Calcular el fan-out en nivel alto y en nivel bajo para una compuerta, conociendo las corrientes máximas de entrada y de salida que maneja.

$$\begin{aligned} I_{OHmax} &= 400\ \mu\text{A} \\ I_{IHmax} &= 40\ \mu\text{A} \\ I_{OLmax} &= 16\ \text{mA} \\ I_{ILmax} &= 2,6\ \text{mA} \end{aligned}$$

TRABAJO PRÁCTICO 3: Arquitectura de Computadoras - Análisis Funcional - Unidad Central de proceso

a.- Arquitectura de un Sistema de Computación

3.01.- Indicar cuáles son los bloques fundamentales que componen un Sistema de Computación. Indicar las características funcionales y circuitales de cada uno de ellos. Justificar las razones por las que se plantea un sistema de computación dividido en diferentes bloques funcionales.

3.02.- Detallar las posibles formas de vinculación entre los bloques planteados en el punto anterior. Justificar la necesidad de dicha vinculación.

3.03.- Describir el sistema de interconexión física entre los bloques de un sistema de computación:

a.- Detallar los distintos tipos de buses, distinguiendo entre los sistemas de 3 y de 4 buses.

b.- Explicar las diferencias entre un bus síncrono y un bus asíncrono, justificando las ventajas y desventajas de cada uno.

c.- Explicar qué se entiende por configuración maestro-esclavo.

3.04.- Explicar las razones por las que se plantea que el bloque de almacenamiento tiene características jerárquicas.

3.05.- Explicar qué es una interfaz de entrada / salida, cuál es su función y ejemplificar .

3.06.- Analizando las características de computadoras comerciales actualmente existentes en el mercado, intente realizar una clasificación de sistemas de computación. Tomar en cuenta sus características en cuanto a tamaño, costo y performance.

3.07.- Tomando como referencia una microcomputadora de las que el mercado ofrece (folleto, publicidad, suplemento especializado), analizar sus componentes, clasificándolos de acuerdo a lo visto en los puntos anteriores, distinguiendo entre memorias, interfaces, periféricos y elementos de soporte.

b.- Arquitectura de la Unidad Central de Proceso

3.08.- Indicar cuáles son los bloques funcionales que constituyen la estructura interna de la CPU.

3.09.- Indicar cuáles son los registros y unidades que componen los siguientes bloques de la CPU: Unidad de Control, Unidad Aritmético Lógica, Entradas/Salidas. Hacer una clasificación de los registros internos de la CPU.

3.10.- Describir el formato de una instrucción. Comparar distintos tipos de formatos.

3.11.- Describir los ciclos que componen la ejecución de una instrucción. Explicar los conceptos de: ciclo de búsqueda, ciclo de ejecución, ciclo de máquina, ciclo de reloj y ciclo de instrucción.

3.12.- Una computadora funciona con una frecuencia de reloj de 100 MHz. Cada una de sus instrucciones se ejecuta, en tiempo fijo, igual a 5 ciclos de reloj. Determinar la eficiencia de funcionamiento de la máquina, medida en cantidad de instrucciones por segundo.

3.13.- Una computadora funciona con una frecuencia de reloj de 500 MHz en sus buses. La configuración de las instrucciones es de dos operandos en memoria. La memoria responde a la CPU en dos ciclos de reloj. Si el bus de direcciones y el bus de datos son ambos de 32 bits, determinar la eficiencia de la computadora, medida en cantidad de instrucciones por segundo, considerando que el tamaño del código de operación es de 16 bits.

c.- Aplicaciones comerciales

3.14.- Indicar las diferencias fundamentales entre un microprocesador y un microcontrolador en un chip.

3.15.- Analizar la estructura interna de los microprocesadores 6809, 68HC11 y 68HC908.

3.16.- Indicar la cantidad de bits del bus de datos y direcciones con que trabajan los procesadores mencionados en el punto anterior.

3.17.- Analizar la estructura y capacidad de memoria que administran los procesadores mencionados.

3.18.- Analizar el formato de instrucción de los procesadores mencionados.

3.19.- Analizar las características en cuanto a ciclo de máquina, ciclo de reloj, frecuencia de operación de los procesadores mencionados. Establecer un valor de eficiencia promedio de estos componentes según la cantidad de instrucciones que pueden ejecutar por unidad de tiempo.

TRABAJO PRACTICO 4: Instrucciones y Modos de Direccionamiento

4.01.- Explicar los conceptos de:

- Instrucciones con referencia a memoria.
- Instrucciones sin referencia a memoria.
- Instrucciones de doble operando.
- Instrucciones de operando único.

En el caso de instrucciones sin referencia a memoria, indicar cómo se obtienen los datos para la ejecución de la instrucción. Indicar si pueden existir instrucciones sin operandos.

4.02.- Describir y ejemplificar distintos tipos de

- instrucciones aritméticas.
- instrucciones lógicas.
- instrucciones de movimiento o transferencia de datos, con y sin referencia a memoria.
- instrucciones de salto o control.

4.03.- Explicar las diferencias entre saltos incondicionales y condicionales, sobre qué información actúan los saltos condicionales en cuanto a la evaluación de la condición y a la transferencia del control de programa.

4.04.- Relacionar las estructuras de control de un lenguaje de alto nivel con las instrucciones de salto analizadas anteriormente.

4.05.- Explicar el concepto de subrutina. Explicar las instrucciones de salto a subrutina y de retorno desde subrutina. Indicar qué registro interno de la CPU está involucrado en estas operaciones. Relacionar el concepto de subrutina con los subprogramas utilizados en lenguajes de alto nivel.

4.06.- Explicar el concepto de modo de direccionamiento.

4.07.- Definir los siguientes modos de direccionamiento:

- Absoluto
- Relativo
- Directo
- Indirecto
- Implícito o inherente

4.08.- Describir y relacionar con la clasificación anterior, los siguientes modos de direccionamiento:

- Paginado
- Inmediato
- Indexado
- Registro base y desplazamiento

4.09.- Explicar la forma de vinculación entre un programa escrito en lenguaje de alto nivel y el programa que realmente ejecuta la computadora.

4.10.- Analizar los modos de direccionamiento de los procesadores 68HC11 y 68HC908. Relacionar con la clasificación general. Destacar las diferencias.

4.11.- Analizar el conjunto de instrucciones de los procesadores 68HC11 y 68HC908. Relacionar las instrucciones de los procesadores mencionados con la clasificación general.

TRABAJO PRÁCTICO 5: Unidad Central de Proceso: Programación en Lenguaje Simbólico

5.01.- Indicar qué hace el procesador 68HC11 cuando ejecuta cada una de las siguientes instrucciones

LDAB #\$25	LDAB \$25	LDAB 25	LDAB \$2525	LDAB \$25,X
ADDA #\$EA	ADDA \$EA	ADDA \$EA,X	ADDA \$46EA	
CBA	BCC \$10	BGE \$F7	BSR \$65	

Especificar los registros de la CPU y las posiciones de memoria involucrados.
Especificar los bits del registro de estado que se modifican.
Escribir cada instrucción en lenguaje de máquina.

5.02.- indicar, para el procesador 68HC11, cuales de las siguientes instrucciones no son válidas y justificar el por qué:

LDAB #\$25	LDAB \$25	LDAB 25	LDAB #\$2525	LDD \$225,X
LDD 225,Y	LDA \$2525	LDD #2525		
ADDA #\$EA	ADDA \$EA	ADDA \$EEAA,X	ADDA \$46EA	
BCC \$100	BGE 100	BSR #\$65		

5.03.- Para una unidad de procesos 68HC11, indicar los contenidos de los acumuladores A y B, y de las direcciones de memoria 00F5, 00F7 y 00F8 después de la ejecución de las siguientes instrucciones:

```

INICIO      CLRA
            LDAB $F5
Otro ADDA    $F7
            DECB
            BNE otro
            STAA $F8
            BRA *
```

si los contenidos iniciales de las direcciones de memoria son respectivamente

00F5: 08
00F7: 12

Indicar qué hace el programa.

5.04.- Para una CPU tipo 68CH11, interpretar los contenidos de las siguientes posiciones de memoria, en donde hay un fragmento de programa. Indicar cuál es el contenido sucesivo de los registros del procesador. Indicar qué posiciones de memoria sufren alteraciones en sus contenidos. Los datos iniciales son
PC: 4000 SP: 0080 IX: 2000 A: 55 B: FF PSW: 07

3000 4F C6 00 E7 00 08 5A 4A 81 00 26 F7 39
4000 BD 30 00 20 FE

Para los planteos que se detallan a continuación, se pide:

Realizar los diagramas de flujo,

Plantear los programas en lenguaje simbólico (mnemónicos) y

Escribir el programa en lenguaje de máquina, para una CPU del tipo del 68HC11. Indicar las suposiciones que se realizan y justificar el algoritmo empleado.

5.05.- Realizar la operación $s = p + q - r$, siendo los datos las posiciones de memoria en donde se hallan los operandos y en donde se almacena el resultado.

5.06.- Realizar la sumatoria $s = n_1 + n_2 + n_3 + \dots + n_n$, siendo datos las direcciones de n_1 , n_n y s .

5.07.- Idem anterior, siendo datos las direcciones de n , n_1 y s . (n = cantidad de sumandos).

5.08.- Realizar la operación $s = p + q$, siendo p y q números de 16 bits. Los datos son las direcciones del MSB de p , de q y de s . (MSB : byte más significativo)

5.09.- Realizar la suma de dos números binarios de cualquier longitud. Los operandos pueden ser positivos o negativos. Los datos son las direcciones de los MSB de los operandos y del resultado, y la dirección en donde se halla la longitud de los operandos.

5.10.- Dados 3 números binarios sin signo, hallar el máximo.

5.11.- Dado un número con signo, devolver el módulo o valor absoluto del mismo. Suponer que los negativos se almacenan en complemento a la base.

5.12.- Dada una lista de números binarios en memoria, ordenarlos en forma creciente en los mismos lugares de memoria, siendo conocidas las direcciones del primer y del último dato en memoria.

5.13.- Repetir el ejercicio anterior considerando conocidas la dirección del primer dato y la cantidad total de datos.

5.14.- Dado un número binario de 8 bits, almacenado en una posición de memoria conocida, convertir el mismo en dos caracteres ASCII representativos

de los dos dígitos hexadecimales que lo forman. (las representaciones literales de los 6 últimos dígitos hexadecimales deberán ser letras mayúsculas). Los resultados se almacenan en direcciones de memoria también conocidas.

5.15.- Se desea convertir a decimal un número binario de ocho bits almacenado en la posición de memoria \$0000. El resultado debe quedar almacenado de la siguiente forma:

Posición \$0001 = CENTENAS.

Posición \$0002 = DECENAS

Posición \$0003 = UNIDADES.

5.16.- Se desea evaluar el dato almacenado en la posición de memoria \$0000. Para ello en la posición de memoria \$0001 debe aparecer almacenado:

">" si el valor es mayor que 128.

"=" si el valor es 128.

"<" si el valor es menor a 128.

5.17.- Dado un vector formado por elementos numéricos naturales, del que se conocen la dirección de su primer elemento y la dirección del último elemento, encontrar el valor máximo incluido en el mismo. Si el valor dado se encuentra una sola vez, indicar la posición del mismo dentro del vector. Si se encuentra más de una vez, indicar la posición en que se lo encontró por primera vez y la cantidad de veces que se repite dicho elemento.

5.18.- Dado un vector formado por elementos numéricos naturales, del que se conocen la dirección de su primer elemento y la dirección del último elemento, determinar si el mismo contiene un valor numérico que se conoce como dato. Si el valor dado se encuentra una sola vez, indicar la posición del mismo dentro del vector. Si se encuentra más de una vez, indicar la posición en que se lo encontró por primera vez y la cantidad de veces que se repite dicho elemento. Si no se lo encuentra, indicar dicha circunstancia de alguna forma visible.

5.19.- Dado un vector formado por caracteres ASCII, del que se conocen la dirección de su primer elemento y la cantidad de elementos, obtener otro vector en el que solamente se incluyan aquellos caracteres del primero que representen letras y/o números.

5.20.- Mediante el uso de un vector, programar un decodificador BCD a 7 segmentos, de acuerdo con lo visto en la práctica de circuitos combinatorios. Cuantos elementos requiere el vector en cuestión?

5.21.- Desarrollar un programa que funcione como "reloj de tiempo real". Con tal objetivo se requerirá que almacene en BCD las horas (en formato 24), minutos y segundos, valiéndose del cambio de estado de la posición de memoria \$0000 (la que, de alguna manera que no nos interesa, cambia su valor entre 0 y 1 cada 500 mseg). Tomando ese cambio como base de tiempo, realizar las rutinas para hacer funcionar el programa como un reloj. Cuando se

produce un cambio es necesario almacenar un \$00 para poder esperar el próximo cambio.

5.22.- Una computadora basada en el microcontrolador 68HC11 tiene conectados ocho pulsadores en un dispositivo mapeado en la posición de memoria \$0080. Al leer esa posición de memoria, si un pulsador está presionado, fuerza un cero lógico en el bit asociado y cuando no lo está, se lee un uno lógico. Cada pulsador está conectado a cada uno de los bits de esa posición de memoria. Se desea generar un programa que mediante las instrucciones BRSET y BRCLR permita discriminar qué pulsador está presionado, teniendo en cuenta que al detectarse más de un pulsador presionado escriba un código de error en la posición de memoria \$0000 ("E"). En caso contrario, deberá escribir en la posición mencionada el código ASCII correspondiente al número del pulsador seleccionado (entre 0 y 7).

5.23.-Una empresa desea construir un tablero de BINGO basado en un microcontrolador 68HC11. Como elemento de entrada se informa el número de la bolilla extraída, el que aparece expresado en BCD empaquetado en la posición de memoria \$0000.

A medida que van saliendo las bolillas deberá ponerse en uno un bit que se corresponda con el número de bolilla. Es decir que se dispone de tantos BITS de memoria como bolillas existen en el juego del BINGO. Cada bit puesto en 1 enciende una lámpara.

Al comenzar el juego todos los bits deben estar en cero y se pueden utilizar las posiciones de memoria necesarias a partir de la posición de memoria \$0002. En caso de detectarse dos veces la misma bolilla debe escribirse en la posición de memoria \$0001 la letra "E", caso contrario esa posición de memoria debe tener una "N".

Se recomienda utilizar instrucciones BSET y BCLR.

Se pide:

- determinar cuántas bolillas tiene un juego de BINGO.
- cuántas posiciones de memoria son necesarias.
- Escribir el programa que administre el funcionamiento del tablero.

TRABAJO PRÁCTICO 6: Unidad Central de Proceso: Programación en Lenguaje Simbólico - Segunda parte: – Programación avanzada.

Introducción al problema

Una variable es una posición de memoria que puede ser leída y escrita. Desde un lenguaje de alto nivel se le indica al compilador el tipo de dato al que pertenece una variable. El tipo de dato indica el tamaño a reservar, es decir, la cantidad de bits que debe ocupar la variable y cómo se decodificará el patrón de bits hallado en esa posición de memoria. La misma palabra binaria tiene significado diferente dependiendo del tipo de dato de la variable. Para la electrónica son solo señales representadas por ceros y unos, es el programador el que asigna un sentido especial a cada palabra de ceros y unos.

Objetivos

Relacionar el concepto de tipo de dato en alto nivel con bajo nivel. Evidenciar los efectos de intentar sobrepasar la capacidad en bits de una variable declarada de un tipo. Afirmar el concepto de overflow en la representación interna de números enteros y caracteres.

Enunciados

6.01.- Dado el programa en C++ que se adjunta, realizar su equivalente en lenguaje simbólico y lenguaje de máquina del HC11 (operaciones y asignaciones, no entradas y salidas). Observar las salidas de C++ y que los únicos mensajes del compilador se refieren al warning "Conversion may lose significant digits". Explicar y mostrar la razón de las salidas obtenidas.

```
/*  
*/  
  
#include <iostream.h>  
#include <conio.h>  
#include <math.h>  
  
//Ejemplos de overflow  
  
void main (void)  
{  
    short i ;  
    int ii ;  
    long l ;  
    float f ;  
    char c ;  
    cout << "Presionar cualquier tecla para avanzar" << endl ; getch() ;  
    i = 32767 ;
```

```

i++;
cout << endl << "32767 + 1 como short da " << i << endl ; getch() ;
ii = 32767 ;
ii++;
cout << endl << "32767 + 1 como int da " << ii << endl ; getch() ;
ii = pow(2,31) ;
ii++;
cout << endl << "2^31 como int da " << ii << endl ; getch() ;
l = pow(2,31) ;
l++;
cout << endl << "2^31 como long da " << l << endl ; getch() ;
i = -32768 ;
i-- ;
cout << endl << "-32768 - 1 como short da " << i << endl ; getch() ;
i = 45000 ;
cout << endl << "i = 45000 como short da " << i << endl ; getch() ;
    i = 32000 + 25000 ;
f = i ;
cout << endl << "32000 + 25000 como entero ==> " << i << endl ;
cout << endl << "32000 + 25000 como real ==> " << f << endl ; getch() ;
    f = 32000 + 25000 ;
i = f ;
cout << endl << "32000 + 25000 como entero ==> " << i << endl ;
cout << endl << "32000 + 25000 como real ==> " << f << endl ; getch() ;
c = 'A' ;
cout << endl << "la A como char da " << c << endl ; getch() ;
    c = 65 ;
cout << endl << "c = 65 como char da " << c << endl ;
c = 321 ;
cout << endl << "c = 321 como char da " << c << endl ;
c = -191 ;
cout << endl << "c = -191 como char da " << c << endl ;
c = 128 ;
cout << endl << "c = 128 como char da " << c << endl ;
c = -128 ;
cout << endl << "c = -128 como char da " << c << endl ;
cout << endl << "Tama" << char(164) << "o de short, int, long, float, char = "
    << sizeof(short) << ", "
    << sizeof(int) << ", "
    << sizeof(long) << ", "
    << sizeof(float) << ", "
    << sizeof(char) << " respectivamente " << endl ;
cout << endl << "Presionar cualquier tecla para terminar" << endl ;
getch() ;

```

Introducción al problema

Los algoritmos típicos para calcular productos y multiplicaciones por sumas sucesivas o restas sucesivas, pueden resultar de baja performance en velocidad cuando se trata de operandos binarios y lenguaje a nivel máquina. Se desarrollan algoritmos alternativos que permitan estadísticamente un mejor desempeño, basados en operaciones sencillas para la electrónica y en la menor cantidad de iteraciones posibles para llegar al resultado con suficiente precisión.

Objetivo

Investigar, analizar ventajas y desventajas e implementar algoritmos para multiplicar en lenguaje de máquina.

Enunciados

6.02.- Plantear los algoritmos para multiplicar números de 16 bits, sin signo y con signo, por medio de sumas sucesivas. Implementar en el lenguaje del 68HC11.

6.03.- Plantear los algoritmos para multiplicar números de 16 bits, sin signo y con signo, por medio de sumas y desplazamientos. Implementar en el lenguaje del 68HC11.

6.04.- Existen varios métodos que permiten realizar la operación de multiplicación a bajo nivel, cuando no se dispone de la instrucción de multiplicación, que tienden a utilizar operaciones sencillas y a buscar buena performance en velocidad. Entre ellos se destaca el algoritmo de Booth, cuyo funcionamiento se basa en que, cuando en el multiplicador existen secuencias de ceros o unos, no se requieren sumas sino solamente desplazamientos. Las sumas se realizan cuando se detecta una transición de 0 a 1 o de 1 a 0. Se pide implementar este algoritmo para números de 16 bits, sin signo y con signo, en el lenguaje del HC11.

6.05.- Implementar el algoritmo de multiplicación de números de 16 bits sin y con signo, utilizando la instrucción MUL del H11.

6.06.- Comparar las implementaciones resultantes en los puntos anteriores en cuanto a velocidad (ciclos de máquina), en cuanto a tamaño del programa, utilización de memoria y de registros internos de la CPU. Analizar las ventajas y las desventajas. Establecer los rangos de aplicación y analizar la calidad en legibilidad.

TRABAJO PRACTICO 7: Unidad de Memoria

a.- Conceptos generales.

7.01.- Explicar los conceptos de capacidad de memoria, velocidad (tiempo de acceso y tiempo de ciclo), palabra de la CPU y palabra de memoria.

7.02.- Hacer un cuadro sinóptico que muestre la clasificación de los distintos tipos de memoria según:

- forma de acceso
- persistencia de la información
- tipo de utilización en relación a la posibilidad de lectura y escritura
- tecnología
- principio de funcionamiento
- jerarquía

7.03.- Definir y ejemplificar los distintos tipos de memoria del punto anterior.

7.04.- Comparar las características de consumo, velocidad y capacidad para las distintas tecnologías y principios de funcionamiento. Identificar dentro de la clasificación realizada previamente.

7.05.- Comparar las características de velocidad, capacidad y costo para las distintas jerarquías de memorias.

7.06.- Indicar a qué tipo de memoria corresponden las estructuras conocidas como stack y buffer. Ejemplificar su utilización.

7.07.- Indicar las funciones de la memoria ROM y de la memoria R/W de una memoria central

b.- Memoria caché

7.08.- Indicar qué es, dónde y para qué se utiliza una memoria cache.

7.09.- Indicar cuáles son las formas típicas de organización de una memoria cache. Analizar la utilización de memorias asociativas y memorias de acceso aleatorio.

7.10.- Sea una memoria cache con un tiempo de acceso de 1 nseg por palabra y una memoria principal, de 10 nseg por palabra. La tasa de aciertos es de 0,90 y la memoria caché está constituida por líneas de 256 palabras cada una. Calcular el tiempo promedio de acceso del sistema. Justificar.

7.11.- Recalcular el valor anterior considerando que la tasa de aciertos es ahora de 0,975.

7.12.- Recalcular el valor anterior considerando ahora que el tamaño de cada línea de memoria caché es de 1024 palabras.

7.13.- Se tiene una unidad de memoria de 256M palabras de 16 bits, y una memoria cache de 1M palabras. La cache utiliza mapeo directo y está dividida en líneas de 256 palabras cada una. Indicar cuántos bits hay en los campos de rótulo e índice de la dirección, y cuántos bits hay en cada palabra de la cache. Determinar en cuantos bloques ha quedado dividida la memoria principal.

7.14.- Repetir el problema anterior considerando que la memoria cache utiliza mapeo asociativo. Analizar los cambios producidos en el sistema de memoria.

7.15.- Para el ejercicio 7.13, analizar qué ocurre si se debe reemplazar un bloque de memoria almacenado en memoria caché. Indicar las razones por las cuales debería ser reemplazado, y cual es el procedimiento más adecuado para efectuar dicho reemplazo.

7.16.- Repetir el análisis anterior para el caso en que la memoria caché esté asignada como mapeo asociativo. Analizar qué diferencias se producen en las técnicas de reemplazo.

7.17.- Analizar las ventajas de la configuración de memoria caché configurada con un esquema de mapeo asociativo por conjuntos de dos vías.

c.- Memoria principal

7.18.-Indicar la cantidad de líneas de dirección, de datos y de control, la cantidad de bits de los registros de direcciones y de datos, la cantidad de posiciones de memoria y entre que números decimales y hexadecimales se encuentran las direcciones, para las siguientes memorias:

a. RAM-RW de 64 Mbytes b. RAM-R/W de 10 Mbytes c. RAM-R/W de 256 Kbits

7.19.- Dibujar el circuito básico de una memoria ROM de 4 direcciones de memoria de 2 bits cada una.

7.20.- Dibujar el circuito básico de una memoria RAM R/W construida con Flip-Flops D de 4 direcciones de 2 bits cada una.

7.21.- Indicar cuál es la máxima capacidad de memoria administrable desde una CPU con un bus de direcciones de 16 bits. Ídem de 32 bits. Influye en algo el tamaño del bus de datos??

7.22.- Indicar cuál es la diferencia de capacidades que hay entre las memorias centrales de dos máquinas con un bus de direcciones de 16 bits, si una de ellas tiene un bus de datos de 8 bits y la otra un bus de datos de 16 bits.

7.23.-Indicar cuántos circuitos integrados se requieren para la memoria central de una PC de 256M x 16 bits, si se dispone de circuitos integrados de 4 Mbits. Ídem si fueran CI de 1024k x 4 bits. Ídem si fueran CI de 1M x 1 bit.

7.24.- Implementar una memoria de 16 Mbytes, utilizando circuitos integrados de 2 M direcciones de un bit cada una. Indicar cuántos circuitos integrados se requieren y cómo se configura el banco de memoria. Si se utiliza un procesador de 28 líneas de direccionamiento, indicar cuál es la decodificación requerida para ubicar el banco a partir de la dirección hexadecimal 1000000.

7.25.- Implementar una memoria ROM de 512 K por 16 bits, utilizando circuitos integrados de 128 Kbytes. Indicar cuántos circuitos integrados son necesarios y cómo se configura el banco. Indicar qué señales de control son necesarias para su manejo por parte de la CPU.

TRABAJO PRACTICO 8: Unidades de Entrada Salida , Periféricos y Comunicaciones

8.01.- Explicar el concepto de unidad o sistema periférico. Ejemplificar.

8.02.- Explicar el concepto de interfaz. Clasificar las interfaces de un sistema de computación en base al uso que se les brinde. Analizar la diferencia entre interfaces genéricas y dedicadas.

8.03.- Explicar el concepto de controlador (desde el punto de vista de la programación de un sistema de computación).

8.04.-Explicar y ejemplificar las formas en que se comunican la unidad central de una computadora con las interfaces de E/S, en cuanto al sistema de buses utilizado. Explicar los conceptos de:

- E/S mapeada en Memoria.
- E/S aislada.
- Procesadores de E/S (Canal de Datos).

8.05.- Dibujar el diagrama en bloques de una interfaz genérica. Indicar los registros internos necesarios y explicar su función. Indicar las líneas de conexión con la unidad central y la memoria principal y con un periférico genérico, y explicar la función particular de cada una de las líneas utilizadas.

8.06.- Explicar las características básicas de los distintos modos de transferencia de datos entre la unidad central de proceso y las interfaces de E/S:

- Controlada por programa.
- Controlada por interrupción.
- Acceso Directo a Memoria (DMA).

Indicar a qué tipo de periféricos se aplican y por qué.

8.07.- Analizar otros tipos de interrupciones que puede admitir un procesador de un sistema de computación, y que no se relacionan con el manejo de periféricos.

8.08.- Analizar, a nivel del hardware de la CPU de un sistema de computación, como se procesan las interrupciones externas.

8.09.- Explicar qué es una interrupción enmascarable y una no enmascarable.

8.10.- Explicar las formas de elección de la dirección de salto a una rutina de servicio de interrupción (vectorizadas y no vectorizadas).

8.11.- Explicar la ventaja obtenida con el control de una operación de E/S iniciada por interrupción, frente al método de control por programa.

- 8.12.- Explicar el concepto de prioridad de interrupción. Ejemplificar.
- 8.13.- Explicar las características de una interfaz con capacidad de DMA (Acceso Directo a Memoria).
- 8.14.- Dibujar un diagrama en bloques que muestre los registros internos necesarios en una interfaz con DMA. Indicar las líneas de conexión con la parte central y con un periférico genérico. Explicar la función particular de cada una.
- 8.15.- Explicar las formas de transferencia por DMA: Masiva o por Robo de Ciclos.
- 8.16.- Escribir una rutina de entrada de datos para un procesador que atiende cuatro interfaces mediante el método de transferencia controlada por programa, indicando las etapas de inicialización, espera y transferencia propiamente dicha de la CPU. Indicar cuál es la etapa que hace a este método ineficiente y por qué. Ejemplificar.
- 8.17.- Repetir la rutina anterior para el caso en que el manejo se realice por interrupciones, considerando que todos los dispositivos de interfaz interrumpen por la misma línea.
- 8.18.- Repetir la rutina anterior considerando ahora que cada uno de los dispositivos tiene asignada una línea individual de interrupciones. Qué consideración adicional hay que hacer en este caso?
- 8.19.- Una computadora utilizada en un sistema de monitoreo controla 32 sensores configurados en cuatro registros de 8 sensores cada uno. Cada sensor genera una señal de estado una vez cada 15 segundos. A su vez, la computadora recibe una señal de interrupción no enmascarable a una frecuencia de 1000 interrupciones por segundo. Se requiere escribir una rutina que, basada en dicha interrupción determine el momento en que debe leer los sensores, almacenando, en cada lectura, una imagen de dichos estados en posiciones determinadas de memoria. En caso de falla (estado del sensor en 1), deberá indicarse en una posición dada de la memoria cual fue el número de sensor (en el rango 0 a 31) que dio falla. Si fuese más de un sensor el que fallara en una vuelta, se indicará tal situación con el valor binario -1
- 8.20.- Explicar las características particulares de la comunicación paralelo y de la comunicación serie.
- 8.21.- Explicar las características particulares de la comunicación serie asincrónica y de la comunicación serie sincrónica.
- 8.22.- Indicar un formato de transmisión serie asincrónica.
- 8.23.- Definir la unidad BAUD. En qué casos se hace diferencia con la unidad de velocidad bit/segundo?

8.24.- Analizar el formato de transmisión sincrónica orientada a caracteres. Analizar un formato de transmisión sincrónica orientada a bits y explicar la ventaja con respecto al anterior.

8.25.- Explicar a qué se refiere la norma RS232 y dar ejemplos de utilización.

8.26.- Indicar qué es una UART y qué es una USRT. Describir sus características circuitales y funcionales.

8.27.- Indicar cuál es la velocidad de transmisión en bauds de un canal de comunicaciones que transmite en formato serie asincrónico, si la frecuencia del reloj de control de la UART correspondiente es de 28800 Hz. Suponiendo que la UART está configurada para transmitir caracteres de 7 bits + paridad y un bit de stop, indicar cuál es la velocidad medida en bits por segundo, en bits de información por segundo y en caracteres por segundo. Justificar las diferencias y relacionar con la unidad baud.

8.28.- Indicar en cuánto tiempo se puede transmitir un documento de 3 Mbytes, utilizando un canal serie asincrónico de 33600 bauds, si cada carácter se transmite en ASCII de 7 bits + paridad y dos bits de stop. Justificar.

8.29.- Analizar las características básicas de la norma de comunicación serie conocida como USB.